

Guía RESUELTA 2022

CENEVAL EGEL Plus ISOFT

PARA SU VENTA EXCLUSIVA EN LOS SIGUIENTES SITIOS WEB:



www.cenevalguias.com



www.pixoguias.com



www.guaceneval.mx



www.eduguia.mx



www.cenevalrespuestas.com



www.guiasceneval.net



www.ceneval.net



www.miguiaceneval.com



www.ceneval.org



www.cenevalmexico.com



www.capacitacionceneval.com



www.comprarlibro.com.mx

ENTORNO SOCIAL

A. Las organizaciones

I. Teoría de las organizaciones

1. Tipos y principios básicos de las organizaciones

Organización.

Arreglo sistemático de personas para llevar a cabo algún propósito específico.

Las organizaciones.

Constituyen la forma de institución dominante en nuestra sociedad: son la manifestación de una sociedad muy especializada e interdependiente, caracterizada por un creciente estándar de vida.

Es una unidad social dentro de la cual las personas establecen relaciones estables entre sí, orientadas a facilitar el logro de un conjunto de objetivos o metas.

• Organismos públicos y privados (R)

Empresas públicas y privadas.

- ✓ **Pública:** en este tipo de empresas el capital pertenece al Estado y generalmente, su finalidad es satisfacer necesidades de carácter social.
- ✓ **Privadas:** lo son cuando el capital es propiedad de inversionistas privados y la finalidad es eminentemente lucrativa.

• Empresas e industrias (R)

Empresas.

Grupo social en el que, a través de la administración del capital y el trabajo, se producen bienes y/o servicios tendientes a la satisfacción de las necesidades de la comunidad.

Clasificación:

- ✓ **Actividad o giro:** Clasificación de acuerdo con la actividad que desarrollean:
 1. Industriales: la actividad primordial es la producción de bienes mediante la transformación y/o la extracción de materias primas. Se clasifican en:
 - a) Extractivas: cuando se dedican a la explotación de recursos naturales renovables o no renovables.
 - b) Manufactureras: son empresas que transforman materias primas en productos terminados.
 2. Comerciales: son intermediarios entre productor y consumidor; su función primordial es la compra-venta de productos terminados. Se clasifican en:
 - a) Mayoristas: cuando efectúan ventas en gran escala o a otras empresas.
 - b) Minoristas o detallistas: las que venden productos al menudeo o en pequeñas cantidades, al consumidor.
 - c) Comisionistas: se dedican a vender mercancía que los productores les dan a consignación, percibiendo por esta función una ganancia o comisión.
 3. Servicios: aquellas que brindan un servicio a la comunidad y pueden tener o no fines lucrativos. Se clasifican en:
 - a) transporte
 - b) turismo
 - c) instituciones financieras
 - d) servicios públicos varios
 - e) servicios privados varios
 - f) educación
 - g) salubridad
 - h) fianzas, seguros
- ✓ **Origen del capital:** dependiendo del origen de las aportaciones de su capital y del carácter a quienes se dirijan

sus actividades, pueden clasificarse en:

1. públicas
2. privadas

A su vez, pueden ser nacionales, cuando los inversionistas son nacionales o nacionales y extranjeros, y transnacionales, cuando el capital es preponderantemente de origen extranjero y las utilidades se reinvierten en los países de origen.

- ✓ **Magnitud de la empresa:** de acuerdo con el tamaño de la empresa se establece que puede ser pequeña, mediana o grande. Criterio para clasificación:
 1. Financiero: el tamaño de la empresa se determina con base en el monto de su capital.
 2. Personal ocupado: una empresa pequeña es aquélla en la que laboran menos de 250 empleados, una mediana aquélla que tiene entre 250 y 1000 trabajadores, y una grande es aquella que se compone de más de 1000 empleados.
 3. Producción: clasifica a la empresa de acuerdo con el grado de maquinización que existe en el proceso de producción.
 4. Ventas: establece el tamaño de la empresa en relación con el mercado que la empresa abastece y con el monto de sus ventas.
 5. Criterio de nacional financiera: para esta institución, una empresa grande es la más importante dentro del grupo correspondiente a su mismo giro. La empresa chica es la de menor importancia dentro de su ramo, y la mediana es aquélla en la que existe una interpolación entre la grande y la pequeña.

Industria.

La actividad primordial es la producción de bienes mediante la transformación y/o la extracción de materias primas. Se clasifican en:

- ✓ Extractivas: cuando se dedican a la explotación de recursos naturales renovables o no renovables.
- ✓ Manufactureras: son empresas que transforman materias primas en productos terminados. Pueden ser de dos tipos:
 1. Empresas que producen bienes de consumo final: producen bienes que satisfacen directamente la necesidad del consumidor
 2. Empresas que producen bienes de producción: satisfacen preferentemente la demanda de las industrias de bienes de consumo final

• Estructura y comportamiento de las organizaciones (R)

Estructura de la organización.

Marco de referencia de una organización según lo expresan su grado de complejidad, formalización y centralización.

- ✓ **Complejidad**: grado de diferenciación en una organización.
- ✓ **Normalización**: grado en que una organización se basa en reglas y procedimientos para dirigir la conducta de los empleados.
- ✓ **Centralización**: tiene que ver con el sitio donde radica la autoridad. Concentración de la autoridad para la toma decisiones en la administración superior.

Principios de la estructura organizacional.

- ✓ **Del objetivo**: todas y cada una de las actividades establecidas en la organización deben relacionarse con los objetivos y propósitos de la empresa
- ✓ **especialización**: el trabajo se realiza más fácilmente si se subdivide en actividades claramente relacionadas y delimitadas
- ✓ **jerarquía**: es necesario establecer centros de autoridad de los que emane la comunicación necesaria para lograr los planes, en los cuales la autoridad y la responsabilidad fluyan en una línea clara e ininterrumpida, desde el más alto ejecutivo hasta el nivel el más bajo
- ✓ **Paridad de autoridad y responsabilidad**: a cada grado de responsabilidad conferido, debe corresponder el grado de autoridad necesario para cumplir dicha responsabilidad
- ✓ **Unidad de mando**: al determinar un centro autoridad y decisión para cada función, debe asignarse un solo jefe, y que los subordinados no deberán reportar a más de un superior
- ✓ **difusión**: las obligaciones de cada puesto que cubren responsabilidad y autoridad, debe publicarse y ponerse

- por escrito a disposición de todos aquellos miembros de la empresa que tengan relación con las mismas
- ✓ **Amplitud o tramo de control:** hay un límite en cuanto al número de subordinados que deben reportar a un ejecutivo, de tal manera que éste pueda realizar todas sus funciones eficientemente
 - ✓ **De la coordinación:** las unidades de una organización siempre deberán mantenerse en equilibrio
 - ✓ **continuidad:** una vez que se ha establecido la estructura organizacional requiere mantenerse, mejorarse y ajustarse a las condiciones del medio ambiente

Etapas de organización.

- ✓ **División del trabajo:** es la separación y delimitación de las actividades, con el fin de realizar una función con la mayor precisión, eficiencia y el mínimo esfuerzo, dando lugar a la especialización y perfeccionamiento en el trabajo
- ✓ **coordinación:** es la sincronización de los recursos y los esfuerzos de un grupo social, con el fin de lograr oportunidad, unidad, armonía y rapidez, en el desarrollo y la consecución de los objetivos.

Tipos de organización.

- ✓ **Organización lineal o militar:** la actividad decisional se concentra en una sola persona, quien toma todas las decisiones y tiene la responsabilidad básica del mando
- ✓ **Organización funcional o de Taylor:** consiste en dividir el trabajo y establecer la especialización de manera que cada hombre ejecute el menor número posible de funciones
- ✓ **Organización línea-funcional:** combinación del organización lineal y la funcional aprovechando las ventajas y evitando las desventajas inherentes a cada una
- ✓ **Organización staff:** surge como consecuencia de las grandes empresas y del avance de la tecnología, lo que origina la necesidad de contar con ayuda en el manejo de detalles, y de contar con especialistas capaces de proporcionar información experta y de asesoría a los departamentos de línea
- ✓ **Organización por comités:** consiste en asignar los diversos asuntos administrativos a un cuerpo de personas que se reúnen y se comprometen para discutir y decidir en común los problemas que se les encomiendan
- ✓ **Organización matricial:** consiste en combinar la departamentalización por proyectos con la de funciones

Niveles organizacionales.

- ✓ **Nivel institucional:** más elevado, compuesto por directivos o altos funcionarios. Nivel estratégico, porque se encarga de definir los principales objetivos y estrategias de la organización y determinar los asuntos relacionados con el largo plazo y la totalidad de la organización.
- ✓ **Nivel gerencial:** se encarga de transformar las decisiones tomadas por el nivel institucional en planes y programas para que el nivel técnico las ejecute. Se encarga de detallar problemas, buscar recursos necesarios y asignarlos a las diversas partes de la organización y distribuir y colocar los productos y servicios de la organización.
- ✓ **Nivel técnico:** nivel operacional, se ejecutan las tareas; se desarrollan los programas y se aplican las técnicas. Se encarga de la ejecución de las operaciones y tareas y está orientado a corto plazo y sigue los programas y rutinas establecidos en el nivel gerencial.

• Áreas administrativas funcionales (R)

Áreas de actividad (funciones básicas de la empresa).

Están en relación directa con las funciones básicas que realiza la empresa a fin de lograr sus objetivos. Dichas áreas son: producción, mercadotecnia, recursos humanos y finanzas.

- ✓ **Producción:** formula y desarrolla los métodos más adecuados para la elaboración de productos, al suministrar y coordinar: mano de obra, equipo, instalaciones, materiales y herramientas requeridas.
- ✓ **Mercadotecnia:** su finalidad es la de reunir los factores y hechos que influyen en el mercado, para crear lo que el consumidor quiere, desea y necesita, distribuyéndolo en forma tal, que esté a su disposición en el momento oportuno, en el lugar preciso y al precio más adecuado.
- ✓ **Finanzas:** se encarga de la obtención de fondos y del suministro del capital, se utiliza en el funcionamiento de la empresa, procurando disponer con los medios económicos necesarios para cada uno de los departamentos, con el objeto de que puedan funcionar debidamente.
- ✓ **Administración de recursos humanos:** su objeto es conseguir y conservar un grupo humano de trabajo cuyas características vayan de acuerdo con los de la empresa, a través de programas adecuados de reclutamiento, de selección, de capacitación y desarrollo.

• **Manuales de organización (R, E)**

Explican los detalles más importantes de la organización:

- ✓ finalidad de cada elemento en la organización
- ✓ declaración de funciones
- ✓ glosario de términos utilizados

Contienen:

- ✓ objetivos generales de la organización
- ✓ políticas generales
- ✓ glosario de términos administrativos
- ✓ nombres de áreas o departamentos y puestos
- ✓ procedimientos de la organización
- ✓ responsabilidades de los altos niveles
- ✓ funciones
- ✓ cartas de organización
- ✓ descripción de puestos
- ✓ descripción actividades
- ✓ introducción y objetivos del manual
- ✓ historia de la empresa

2. Proceso administrativo

Conjunto de fases o etapas sucesivas a través de las cuales se efectúa la administración, mismas que se interrelacionan y forman un proceso integral.

• **Planeación (R)**

La determinación de los objetivos y elección de los cursos de acción para lograrlos, con base en investigación y elaboración de un esquema detallado que habrá de realizarse en un futuro.

• **Organización (R)**

El abastecimiento de la estructura necesaria para la sistematización racional de los recursos, mediante la determinación de jerarquías, disposición, correlación y agrupación de actividades, con el fin de poder realizar y simplificar las funciones del grupo social.

• **Dirección (R, E)**

Ejecución de los planes de acuerdo con la estructura organizacional, mediante la guía de los esfuerzos del grupo social a través de la motivación, la comunicación y la supervisión.

• **Control (R)**

La evaluación y medición de la ejecución de los planes, con el fin de detectar y prever desviaciones, para establecer las medidas correctivas necesarias.

3. Recursos humanos

Son trascendentales para la existencia de cualquier grupo social; de ellos depende el manejo y funcionamiento de los demás recursos. Poseen características tales como: posibilidad de desarrollo, creatividad, ideas, imaginación, sentimientos, experiencia, habilidades, etc., mismas que los diferencian de los demás recursos.

• **El personal (R)**

Según la función que desempeñen y el nivel jerárquico en que se encuentran dentro de la organización, pueden ser:

- ✓ **obreros:** calificados y no calificados
- ✓ **oficinistas:** calificados y no calificados
- ✓ **supervisores:** se encargan de vigilar el cumplimiento de las actividades

- ✓ **técnicos:** efectúan nuevos diseños de productos, sistemas administrativos, métodos, controles, etc.
- ✓ **ejecutivos:** se encargan de poner en ejecución las disposiciones de los directivos
- ✓ **directores:** fijan los objetivos, estrategias, políticas, etc.

• Perfiles de puestos y funciones (R)

Puesto.

Grupo de tareas que se deben desarrollar para que una organización pueda alcanzar sus objetivos.

Análisis de puestos.

Proceso sistemático de determinar las habilidades, deberes y conocimientos necesarios para desempeñar puestos en una organización.

Método cuya finalidad estriba en determinar las actividades que se realizan en el mismo, los requisitos que debe satisfacer la persona que va a desempeñarlo con éxito, y las condiciones ambientales que privan en el sistema donde se encuentra enclavado.

Es una técnica en la que se clasifican pormenorizadamente las labores que se desempeñan en una unidad de trabajo específica e impersonal, así como las características, conocimientos y aptitudes, que debe poseer el personal que lo desempeña.

Propósito:

- ✓ determinar cuáles son las tareas físicas y mentales que desarrolla el trabajador
- ✓ cuando se debe terminar el trabajo
- ✓ donde se debe desarrollar el trabajo
- ✓ como desempeña su puesto el trabajador
- ✓ por qué hace este trabajo
- ✓ cualidades necesarias para desarrollar el puesto

Objetivos:

- ✓ encausar el reclutamiento de personal
- ✓ ayuda para selección objetiva del personal
- ✓ fijar programas de capacitación y desarrollo
- ✓ base para estudios de calificación de méritos
- ✓ elemento para estudios de evaluación de puestos
- ✓ para manuales organización
- ✓ orientar contratación
- ✓ fines contables y presupuestales
- ✓ estructurar sistemas de higiene y seguridad industrial
- ✓ sistema de incentivos
- ✓ determinar montos de fianzas y seguros
- ✓ planeación de recursos humanos
- ✓ efectos organizacionales
- ✓ supervisión
- ✓ auditorías administrativas
- ✓ técnica para una mejor administración de recursos humanos

El análisis de puestos proporciona un resumen de sus deberes y responsabilidades con otros puestos, los conocimientos y habilidades necesarios y las condiciones de trabajo en las que se realiza.

Métodos para el análisis de puestos.

- ✓ **Cuestionarios:** que los empleados identifiquen las tareas que desempeñan.
- ✓ **Observación:** observar al empleado en el desarrollo de sus tareas y registrar sus observaciones.
- ✓ **Entrevistas:** entrevistar al empleado y al supervisores sobre los deberes que cumple.
- ✓ **Registro del empleado:** bitácora de los empleados de sus actividades laborales diarias.

El análisis de puestos contiene:

- ✓ **Descripción del puesto:** determinación técnica de lo que el trabajador debe hacer, integrada por:
 1. el encabezado o identificación:
 2. título

- 3. ubicación
 - 4. instrumental
 - 5. jerarquía
- ✓ **Descripción genérica:** definición breve y precisa del puesto.
- ✓ **Descripción específica:** detalle de las actividades que se realizan en el puesto.
- ✓ **Especificación del puesto:** enunciación precisa de los requisitos que debe satisfacer el trabajador para desempeñar el puesto:
- 1. escolaridad y conocimientos
 - 2. requisitos físicos, legales, mentales y de personalidad
 - 3. esfuerzo
 - 4. responsabilidad
 - 5. condiciones de trabajo

• Liderazgo y motivación (RP)

Liderazgo.

Autoridad basada en fundamentos carismáticos, es decir, personales. La persona se encuentra investido de características personales que lo colocan arriba del promedio de los demás miembros del grupo. El líder establece metas, reconcilia posiciones internas y presiones ambientales.

Tipos de supervisores.

- ✓ **Autócrata consumado:** para dirigir el grupo se apoya en la autoridad de la organización formal. Supone que precisa forzar a la gente a hacer lo que no desean: trabajar. Siente el trabajo como un castigo, para él y los demás.

Posibles reacciones:

- 1. sumisión y resentimiento
- 2. aceptación mínima de responsabilidad
- 3. irritabilidad
- 4. la gran satisfacción es hacer tonto al supervisor

- ✓ **Autócrata benévolo:** supervisor paternalista, manipulador, su actitud es vender la idea. Trata de usar no la autoridad sino la relación amistosa con sus subordinados como instrumentos de influencia.

Posibles reacciones:

- 1. la mayoría siente simpatía, pero algunos detectan su verdadera actitud y les antipatiza
- 2. la iniciativa permanece estática en espera de la reacción de supervisor
- 3. sumisión y falta de desarrollo personal
- 4. nadie desarrolla ideas positivas

- ✓ **Indiferente:** individuo que no toma responsabilidad alguna, ni la suya ni la del empleado.

Posibles reacciones:

- 1. la moral del trabajo y la productividad al mínimo
- 2. descuido en el trabajo, rendimiento bajo
- 3. el empleado tiene poco interés en el trabajo o en mejorar
- 4. no hay espíritu de grupo o trabajo en equipo
- 5. nadie sabe qué hacer ni qué esperar

- ✓ **Demócrata:** dirige al grupo, no basándose en la autoridad formal, sino informal que se deriva de su persona. Toma lo suyo y da a cada quien lo que le corresponde: afronta su responsabilidad y respeta la de sus trabajadores. Proporcionan medios para que puedan mejorarse los subordinados.

Posibles reacciones:

- 1. alto índice de entusiasmo hacia trabajo
- 2. producción de excelente calidad y cantidad
- 3. excelente trabajo en equipo
- 4. sienten que hay éxito en el trabajo
- 5. menos problemas de rendimiento y motivación

Motivación.

Factores capaces de provocar, mantener y dirigir la conducta hacia un objetivo. Factores que originan causas.

El comportamiento puede explicarse mediante el ciclo motivacional, que concluye con la satisfacción o la frustración, o incluso con la compensación de las necesidades humanas. Estas pueden clasificarse jerárquicamente, colocando en la base las necesidades primarias (necesidades fisiológicas y de seguridad), y en la cima las necesidades secundarias (necesidades sociales, de autoestima y de autorealización). Aunque estas necesidades se presentan de manera simultánea, las primarias prevalecen frente a las secundarias. La motivación puede explicarse por medio de influencia de dos factores: los higiénicos o de no satisfacción y los motivacionales o de satisfacción. Pero la motivación también puede explicarse mediante un modelo situacional: la motivación para producir depende de la utilidad de los resultados intermedios respecto de los resultados finales.

• Capacitación (E)

Adquisición de conocimientos, principalmente de carácter técnico, científico y administrativo.

Diseñada para permitir que los aprendices adquieran conocimientos y habilidades necesarias para sus puestos actuales.

Su función es preparar y mejorar los conocimientos sobre los aspectos técnicos del trabajo. La capacitación se debe de ofrecer tanto a los empleados como a los ejecutivos y funcionarios en general; se puede realizar en aulas especiales, en el lugar de trabajo o en forma combinada. El método más efectivo para obtener un eficiente aprendizaje es a través de la participación en grupos, donde el elemento humano adquiere los conocimientos por medio de las aportaciones propias y de los demás.

La mayor parte de las actividades de capacitación de los empleados busca modificar una o más de estas habilidades.

Técnicas. La mayor parte de la capacitación se dirige a actualizar y mejorar las habilidades técnicas de un empleado. Estas incluyen las habilidades básicas, así como las competencias específicas del puesto.

Interpersonales. Esta incluye aprender a escuchar y no solo a oír, como comunicar las ideas propias con mayor claridad, y cómo reducir conflictos.

Solución de problemas. Cuando las habilidades para la solución de problemas de los empleados son deficientes, la administración quizás desee mejorar esta habilidad por medio de la capacitación. Esto incluirá la participación en actividades para afinar la lógica, el razonamiento y las habilidades para definir los problemas, evaluar sus causas, desarrollar alternativas, analizarlas y seleccionar soluciones.

Los administradores pueden valerse de cuatro procedimientos para determinar la capacitación que necesitan las personas de su organización o subunidad:

1. **Evaluación del desempeño.** El trabajo de cada empleado se mide comparándolo con las normas de desempeño o los objetivos establecidos para su trabajo.
2. **Analizar los requisitos del trabajo.** Se estudian las habilidades o los conocimientos que se especifican en la descripción del trabajo correspondiente y los empleados que no cuenten con las habilidades o los conocimientos necesarios pasan a ser candidatos para un programa de capacitación.
3. **Analizar la organización.** Se estudia la eficiencia de la organización y su éxito para alcanzar las metas, con el objeto de determinar las diferencias que existen.
4. **Las encuestas de los empleados.** Se pide a los gerentes y a los no gerentes que describan los problemas que están teniendo con su trabajo y las medidas que consideran necesarias para resolverlos.

Métodos de capacitación

- ✓ **Capacitación en el trabajo:** incluye la rotación de puestos y sustituciones. La rotación de puestos implica transferencia laterales que permiten a los empleados trabajar en diferentes puestos. Los empleados llegan a aprender una gran variedad de puestos, mientras obtienen una mayor percepción de la interdependencia que hay entre puestos, y una perspectiva más amplia de las actividades organizacionales. Empleados nuevos a menudo aprenden sus puestos al sustituir a un veterano fogueado, en ambos casos, el sustituto trabaja bajo la observación de un trabajador con experiencia, que actúa como modelo a quien el sustituto intenta imitar.
- ✓ **Capacitación fuera del trabajo:** los más populares son conferencias en salones de clase, películas y ejercicios de simulación. Las conferencias en salones de clase son muy convenientes para proporcionar información

específica. Se puede usar con efectividad para desarrollar habilidades técnicas y de solución de problemas. También se pueden exhibir películas y videos para mostrar explícitamente habilidades técnicas que no se presentan con facilidad mediante otros medios. Las habilidades interpersonales y de solución de problemas pueden aprenderse mejor por medio de ejercicios de simulación como análisis de casos, ejercicios sobre experiencias, representación de roles y sesiones de interacción en grupo. Así también es la capacitación vestibular, en la que los empleados aprenden sus puestos en el mismo equipo que ellos utilizarán; solo que el entrenamiento se lleva a cabo en un ambiente simulado de trabajo, no en el verdadero lugar de trabajo.

• **Relaciones interpersonales (E, RP)**

Relaciones humanas.

Intervención de dos o más personas.

Relaciones internas con los empleados.

Comprenden las actividades en la gerencia de recursos humanos que están asociadas con el movimiento de los empleados dentro del organización. Estas actividades incluyen promoción, transferencia, degradación, renuncia, despidos, suspensiones y jubilación. La disciplina y la acción disciplinaria también son aspectos cruciales de las relaciones internas con los empleados.

Un buen ambiente laboral, es un medio determinante para favorecer el óptimo cumplimiento de las metas y objetivos.

No es factible lograr que las actividades se realicen de una manera eficaz y eficiente, si dentro del grupo de trabajo se carece de disciplina, ética profesional, valores y sobre todo una buena dosis de respeto entre los individuos que lo conforman.

Las relaciones interpersonales en el trabajo constituyen un papel crítico en una empresa. Aunque la calidad de las relaciones interpersonales en sí no bastan para incrementar la productividad, sí pueden contribuir significativamente a ella, para bien o para mal.

Los problemas en las relaciones interpersonales ocurren como resultado del compromiso de los involucrados en sus propias perspectivas, ideas, opiniones y sentimientos que abusan o pasan por alto los de los otros.

La interacción entre dos personas puede ocurrir en varias etapas de la comunicación:

Comunicación abierta. La comunicación se procesa entre dos “yo” abiertos de las personas. Un análisis superficial de la interacción entre personas puede permanecer en la superficie de este tipo de relaciones interpersonales. Es la etapa más común de relaciones interpersonales.

Revelaciones subrepticias o inconscientes. En la segunda etapa ya se puede percibir señales o significados del “yo” ciego de una persona, cuya comunicación ella no percibe.

Confidencia o nivelación. Es el tercer grado de comunicación, cuando la persona deliberadamente revela algo que generalmente esconde. La persona hace la confidencia a alguien o nivela cuando comparte relaciones o sentimientos generados por acontecimientos inmediatos.

Contagio emocional. Cuando una persona puede influir los sentimientos de otra, sin que ninguno de los dos “yo” de los dos tenga conciencia del origen del sentimiento o de la comunicación.

4. Tópicos selectos de administración

• **Planeación estratégica (E, RP)**

Proceso por el cual la alta dirección determina propósitos y objetivos globales y la forma en que deben alcanzarse.

Es el proceso de determinar cuáles son los principales objetivos de una organización y los criterios que presidieran la adquisición, uso y disposición de recursos en cuanto a la consecución de los referidos objetivos. Estos en el proceso de planificación estratégica, engloban misiones o propósitos, determinados previamente, así como los objetivos específicos buscados por una empresa.

Es una estrategia para atraer, mejorar y conservar los Recursos Humanos en una empresa, la cual contiene la necesidad de :

- ✓ Evaluar los Recursos Humanos actuales
- ✓ Proyectar los requerimientos laborales futuros
- ✓ Asegurar la disponibilidad de los recursos laborales cuando se necesiten

• **Calidad y productividad (E, RP)**

Los mejores métodos para mejorar la calidad y la productividad global están directamente relacionados con los asuntos de recursos humanos. Motivación, cultura y educación son los mejores métodos para elevar la productividad.

El nuevo concepto de calidad tiene que ver con los requisitos de los consumidores, dado que un producto o servicio sólo tiene calidad en la medida que satisface las expectativas del cliente. Además, es una filosofía que debe convertirse en la forma de vida de todos los integrantes de la organización.

Buscar la calidad obliga a las organizaciones a mejorar el proceso de producción, cuidar el diseño exacto del producto final, reducir al mínimo los defectos, evitar los retrabajos, eliminar los desperdicios, uniformar los productos, lograr exactitud en el manejo de materiales; todo lo anterior trae disminución de costos; asimismo, cuidar todos estos detalles en la producción, evita las devoluciones, las quejas, los gastos en cubrir garantías, entre otros. Como vemos, esto es aumento de la productividad

• **Reingeniería (E, RP)**

La compañía repensa y rediseña su sistema de negocios para volverse más competitiva. Es el repensar fundamental y el rediseño radical de los procesos del negocio para alcanzar mejoras drásticas en medidas cruciales y contemporáneas de desempeño como costos, calidad, servicios y velocidad. Expone el acento en el rediseño radical del trabajo en las compañías que se organizan alrededor del proceso, en lugar de hacerlo por departamentos funcionales. Enfoca los aspectos globales del diseño de puestos, estructuras organizacionales y sistemas administrativos. Hace hincapié en que se debe organizar el trabajo alrededor de los resultados en contraste con las actividades o funciones.

Reingeniería es el concepto actual que se le da a los cambios dramáticos que sufre una organización al ser reestructurados sus procesos. La base de la reingeniería es el servicio al cliente, aplicar reingeniería a un proceso hará que éste se centre en obtener el mejor producto para satisfacer al cliente.

Los 13 conceptos fundamentales de la reingeniería:

1. Consiste en empezar de cero, en una hoja en blanco.
2. Consiste en cambios radicales, brutales, espectaculares.
3. Está enfocada a procesos.
4. Tiene una visión holística.
5. La división del trabajo ya no funciona.
6. Es enemiga de la especialización. Es multiespecialización (generalista).
7. Se apoya en el principio de la incertidumbre (Teoría del Caos).
8. Su herramienta principal es la destrucción creativa.
9. No hay un "modelo de reingeniería". No hay un plan preestablecido.
10. Lo más importante es un cambio de mentalidad o de enfoque. Metanoia.
11. En un primer momento debe realizarse de arriba hacia abajo.
12. En un segundo momento, la reingeniería requiere un impulso en sentido inverso, de abajo hacia arriba.
13. Si uno no está convencido es mejor no hacer reingeniería.

II. Tópicos de manejo financiero

1. Principios básicos de contabilidad

• **Contabilidad general (R)**

Es el conjunto de conocimientos y funciones que tienen por objeto la elaboración, comunicación, auditoria y análisis e interpretación de la información relevante acerca de la situación y evolución de la realidad económica y financiera de la empresa, expresada en su mayor parte en unidades monetarias, con el fin de que, tanto las terceras personas interesadas como los directivos de la empresa puedan adoptar decisiones.

• **Estados financieros básicos: balance general, estado de resultados (R)**

Estados Financieros

El principal medio para reportar información financiera de propósito general a las personas externas a la organización comercial es un conjunto de informes llamado “estados financieros”. Las personas que reciben estos informes son llamados usuarios de los estados financieros.

Balance General

Un “balance general o estado de situación Financiera” que muestra los datos específicos de la posición financiera de la empresa para indicar los recursos que posee, las obligaciones que debe y el monto del capital propio (inversión) en el negocio. Documento contable que presenta situación financiera de un negocio en una fecha determinada ya que muestra la situación financiera de un negocio, porque muestra clara y detalladamente el valor de cada una de las propiedades y obligaciones, así como el valor del capital.

El balance general presenta la situación financiera de un negocio en una fecha determinada, o sea, la del día en que se practica porque si se elaborará un nuevo balance al siguiente día no presentaría la misma situación financiera, debido a que los saldos de las cuentas serían distintos porque hay operaciones que se realizan solas, es decir, sin la intervención de ninguna persona, por ejemplo: la baja de valor que sufre el mobiliario por el transcurso del tiempo, la amortización que sufren los gastos de instalación, etc.

La situación financiera de la empresa le interesa conocerla a:

- ✓ Los propietarios, socios, accionistas, para saber si los recursos invertidos en ella han sido bien administrados y si han producido resultados satisfactorios.
- ✓ Al gobierno, para saber si el pago de los impuestos respectivos han sido calculado correctamente.
- ✓ A los acreedores (instituciones de crédito particulares, que otorgan préstamos o proveedores que conceden crédito), les interesa conocer la situación financiera de la empresa.

Para que la situación financiera indicada en el balance general sea lo mas exacta posible, se deben incluir absolutamente todo el activo y todo el pasivo valuados correctamente, es decir, con su verdadero valor, pues cualquier alteración, ya sea de más o de menos, se afectará directamente en el capital.

Para que la situación financiera indicada en el balance general sea lo mas exacta posible, se deben incluir absolutamente todo el activo y todo el pasivo valuados correctamente, es decir, con su verdadero valor, pues cualquier alteración, ya sea de más o de menos. Se afectará directamente en el capital.

El balance general debe de contener los siguientes datos:

ENCABEZADO

- ✓ Nombre de la empresa o nombre del propietario
- ✓ Indicación de que se trata de un balance general o estado de situación financiera.
- ✓ Fecha de formulación

CUERPO

- ✓ Nombre y valor detallado de cada una de las cuentas que forman el activo
- ✓ Nombre y valor detallado de cada una de las cuentas que forman el pasivo
- ✓ Importe del capital contable

FIRMAS

- ✓ Del contador que hizo y autorizó
- ✓ Del propietario o responsable legal del negocio

El Balance general se puede presentar en 2 formas:

- ✓ En forma de reporte

- ✓ En forma de cuenta

Estado de resultados

Un “**estado de resultados**” Documento contable que muestra detallada y ordenadamente la utilidad o perdida de ejercicio fiscal de una empresa es también un complemento del balance general que indica la rentabilidad del negocio con relación al año anterior (u otro periodo).

2. Contabilidad de costos

• Control de inventarios (R, E)

Control de Inventarios

Los inventarios constituyen las partidas del activo corriente que están listas para la venta. Mercancía que posee una empresa en el almacén valorada al costo de adquisición, para la venta o actividades productivas.

El control interno de los inventarios se inicia con el establecimiento de un departamento de compras, que deberá gestionar las compras de los inventarios siguiendo el proceso de compras.

Existen varios métodos para llevar el manejo y control de inventarios, los cuales son:

PEPS (Primero en Entrar Primero en Salir)

UEPS (Ultimo en Entrar Primero en Salir)

- ✓ **Método PEPS:** tipo de inventario perpetuo que detalla por medio de la Tarjeta de Control de inventario, las salidas y entradas de las mercancías. Establece que la primera mercancía que se compra es la primera en venderse o salir.
- ✓ **Método UEPS:** tipo de inventario perpetuo que estable que las últimas mercancías que se comprar son las que primero se venden o salen.

Entre las medidas de control interno tenemos:

1. Hacer conteos físicos periódicamente.
2. Confrontar los inventarios físicos con los registros contables.
3. Proteger los inventarios en un almacén techado y con puertas de manera que se eviten los robos.
4. Realizar entrega de mercancías únicamente con requisiciones autorizadas.
5. Proteger los inventarios con una póliza de seguro.
6. Hacer verificaciones al azar para comparar con los libro contables.

3. Presupuestos

• Teoría de presupuesto (R)

Los fundamentos teóricos y prácticos del presupuesto, como herramienta de planificación y control, tuvieron su origen en el sector gubernamental a finales del siglo XVIII, la palabra se deriva del francés antiguo bougette o bolsa. Intento perfeccionarse posteriormente en el sistema inglés con el término budget de conocimiento común y que recibe en nuestro idioma la denominación de presupuesto esto es uno de los instrumentos más valiosos en poder de la dirección financiera de cualquier organización para formular la política de liquidez, proceder a la inversión oportuna de fondos sobrantes, acudir a créditos cuando las disponibilidades monetarias no permitan la cobertura integral de los compromisos y trazar la política de dividendos.

• Clasificación de presupuestos: ingresos, egresos, áreas de responsabilidad, programas y actividades (R)

1. **Rígidos, estáticos, fijos o asignados:** Para un solo nivel de actividad
2. **Flexibles o variables:** se elaboran para diferentes niveles de actividad y pueden adaptarse.
3. **A Corto Plazo:** se planifican para cumplir un ciclo de operaciones de un año
4. **A largo Plazo:** Se ubican para los planes de desarrollo del estado y grandes empresas
5. **De operación o Económicos:** Presupuestan actividades para el periodo siguiente al cual se elabora y cuyo contenido a menudo se resume en un estado. De perdidas y ganancias.
6. **Financieros:** Incluyen el cálculo de partidas que inciden fundamentalmente en el balance.

7. **Presupuesto del sector Público:** Cuantifican los recursos que requiere la operación normal, la inversión y el servicio de la deuda pública de los organismos y las entidades oficiales.
8. **Presup. Del Sector Privado:** Los utilizan las empresas particulares como base de planificación de las actividades empresariales.

• Técnicas de presupuestación (R)

Son el conjunto de procedimientos utilizados para desarrollar su actividad, que pueden provenir de la experiencia o de las investigaciones realizadas en el desarrollo de su trabajo.

El Presupuesto es un conjunto de pronósticos referentes a un periodo predeterminado. Es un plan esencialmente numérico, para anticipar las operaciones. La certeza de sus resultados depende de la calidad de la información estadística de que se dispone al efectuar la estimación. Tiene el objetivo de obtener planificación unificada y sistematizada de las posibles acciones, en concordancia con los objetivos de la empresa.

Requisitos para integrar y ejercer un buen presupuesto:

Conocimiento de la empresa. La forma, contenido y profundidad del presupuesto siempre va ligado con los objetivos, necesidades y organización de la empresa. Su preparación se basa en el principio de que todas las transacciones de la entidad están íntimamente relacionadas entre sí

Presupuesto por el tipo de recurso que controla

- ✓ **De Ingresos.** Es el primer paso de la integración del programa presupuestal. En este instrumento se señalan los recursos económicos disponibles para la realización de los gastos que implica el programa de actividades correspondiente. Se compone de:
 - Presupuesto de ventas. Generalmente el presupuesto de ventas es el eje de los demás presupuestos. Debe cuantificarse en unidades, en especie, por cada tipo o línea de artículos; una vez afinado, se procede a su valuación, de acuerdo con los precios del mercado, regidos por la oferta y la demanda, o cuando no sea así, por el precio de venta determinado con los elementos anteriores, se tiene el monto de ventas en valores monetarios.
 - Presupuesto de otros ingresos.
 - Presupuesto de ingresos e inversiones
- ✓ **De Egresos.** También es indispensable para la integración del programa presupuestal. Se compone de:
 - Presupuesto de inventarios, producción, costo de producción y compras
 - Presupuesto de costo de distribución y administración
 - Presupuesto de impuestos
 - Presupuesto de aplicación de utilidades

Presupuesto por el tipo de empresa

- ✓ **Públicos.** Corresponden a gobiernos (municipales, estatales, Federación) y empresas descentralizadas. Se estima, en primer lugar, el monto de las necesidades sociales a satisfacer, y posteriormente se planea la forma de generar los ingresos necesarios para cubrirlas
- ✓ **Privados.** Pertenecen a la operación de las empresas privadas, primero se estiman los ingresos, y posteriormente se planea su distribución o aplicación

Presupuesto por su contenido

- ✓ **Principales.** Corresponden a la agregación y resumen de los presupuestos auxiliares y se presentan los aspectos relevantes de todos los presupuestos de una empresa.
- ✓ **Auxiliares.** Muestran en forma analítica las operaciones estimadas para cada uno de los departamentos de la organización

Presupuesto por su forma

- ✓ **Flexibles.** Consideran diferentes planes opcionales de aplicación, ante las posibles variaciones en las situaciones previstas.
- ✓ **Fijos.** No varían en todo el periodo. Se elaboran con un alto grado de exactitud, por lo que la entidad debe ceñirse inflexiblemente a ellos.

Presupuesto por su duración

- ✓ **De corto plazo.** Abarcan un año o menos
- ✓ **De largo plazo.** Contemplan más de un año

- ✓ **Planes gubernamentales.** Incluyen periodos completos de gobierno (cuatrienio en Estados Unidos, sexenio en México)

Presupuesto por su reflejo en los Estados financieros

- ✓ **De posición financiera.** Muestran la posición estática que tendría la entidad en el futuro, si se cumplieran las predicciones (Posición financiera presupuestada)
- ✓ **De resultados.** Señalan las posibles utilidades futuras
- ✓ **De costos.** Reflejan las posibles erogaciones futuras por concepto del Costo total

Presupuesto por las finalidades que pretende

- ✓ **De promoción.** Requiere una estimación previa de ingresos y egresos, para preparar un Proyecto financiero y de Expansión
- ✓ **De aplicación.** Se elaboran para solicitud de créditos. Pronóstico general sobre la distribución de los recursos con que contará la entidad en el futuro
- ✓ **De fusión.** Determinan anticipadamente las operaciones resultantes de una conjunción de entidades
- ✓ **Por áreas y niveles de responsabilidad.** Cuantifica la responsabilidad de los encargados de las áreas y niveles en que se divide la entidad
- ✓ **Por programas.** Expresan el gasto en relación con los objetivos que se persiguen. Determina el costo de las actividades concretas que cada dependencia debe ejecutar para desarrollar los programas a su cargo
- ✓ **Base cero.** Se formula sin tomar en cuenta las experiencias previas de operación
- ✓ **Presupuestos previos.** Constituyen el proyecto final, que se somete a la consideración de la dirección de la entidad. En el caso del sector público se denomina proyecto de presupuesto, el cual una vez aprobado por la Cámara de Diputados se constituye en el Presupuesto de Egresos de la Federación.
- ✓ **Presupuesto definitivo.** Son los presupuestos previos, más los cambios y correcciones indicados por la dirección general.
- ✓ **Presupuestos maestros o tipo.** Son "plantillas" presupuestales que permiten estandarizar la generación de presupuestos semejantes.

Presupuesto financiero

Comprende el Presupuesto de caja y todas las demás operaciones financieras donde no interviene la caja, como: el trueque y el intercambio.

Tiene por objetivo pronosticar y controlar todos los elementos que forman la posición financiera (Balance), y el efecto que producirán las estimaciones sobre las cuentas de caja y bancos, así como el "fondo de operación" (capital invertido en bienes: planta, maquinaria, equipo inventarios). Auxiliar en la toma de decisiones.

El presupuesto de caja se conforma por los saldos disponibles en caja y bancos, inversiones temporales de fácil realización y estimaciones de dinero a recibir (cobros, financiamiento).

Su realización debe ser cuidadosamente planeada y desarrollada, basada en los demás presupuestos auxiliares, ya que éste se realiza al último.

Objetivos:

a) Respecto a los ingresos

- ✓ Conocer la estimación correcta de los cobros, de acuerdo con las políticas de crédito establecidas
- ✓ Precisar el financiamiento exterior, de conformidad con los planes elaborados
- ✓ Evaluar el cumplimiento de las decisiones tomadas, en relación a los aumentos de capital y las emisiones de obligaciones
- ✓ Procurar la obtención de líneas de crédito para el descuento de documentos

b) Respecto a los pagos

- ✓ Procurar el cumplimiento oportuno de las obligaciones regulares, por pagos de sueldos y salarios, servicios de agua, luz, alquiler
- ✓ Promover el establecimiento de las provisiones necesarias para el cumplimiento de obligaciones derivadas del logro de otros presupuestos, como impuestos a la producción, IVA, ISR.
- ✓ Fijar políticas de pago por créditos de proveedores, en concordancia con la rotación de las cuentas por cobrar y el plazo medio de cobranza.

- ✓ Identificar las economías por pagos anticipados

c) Generales

- ✓ Mejorar el manejo de efectivo, manteniendo saldos a niveles lo más bajos posibles.
- ✓ Apoyar las decisiones sobre adquisición de bienes de operación (el capital invertido en operaciones de producción y ventas, generalmente permite mayor utilidad).
- ✓ Escoger el momento adecuado para elegir aumento de capital, emisión de acciones y créditos.
- ✓ Elaborar los pronósticos del efectivo por meses, semanas y días para inversiones a corto plazo.
- ✓ Realizar coordinadamente los cobros y pagos.

Presupuesto de operación (flujos de caja)

Es el estudio, análisis y pronóstico de la circulación del dinero, con referencia a sus fuentes y usos en una empresa, en un periodo futuro determinado, con el objeto de planear y controlar el dinero.

Objetivos:

- ✓ Mejorar el manejo de fondos
- ✓ Mantener saldos a la vista lo más bajos posibles, sin lastimar la estabilidad financiera
- ✓ Aplicar disponibilidades temporales en inversiones de corto, mediano y largo plazo
- ✓ Analizar posibles reducciones de efectivo, para invertirlo en bienes de operación
- ✓ Lograr economías por pagos anticipados
- ✓ Planejar la salida de dividendos por pagar
- ✓ Financiar el crecimiento
- ✓ Informar de disponibilidades de efectivo para obtención de préstamos
- ✓ Coordinar cobros y pagos
- ✓ Elaborar pronósticos a corto y largo plazo
- ✓ Servir de base para la elaboración del presupuesto de caja

Los elementos de un cash flow:

- ✓ **Fondos de capital.** Esta clasificación se hace para precisar la circulación continua del capital que va, del dinero a inventarios, de éstos a cuentas por cobrar, que finalmente se convierten en nuevo dinero, maquinaria y equipo.
- ✓ **Fondos de efectivo.** Se refiere al análisis del monto de los fondos que deben mantenerse líquidos para cubrir obligaciones de exigencia inmediata o bienes para la operación normal de la entidad
- ✓ **Fondos de operación.** Montos invertidos en bienes de disponibilidad inmediata, para el trabajo normal de la empresa. Incluye el capital para inventarios y para bienes a más de un año

Secuencia de efectivo. Estudio del movimiento de los saldos en cuentas bancarias

Obtención óptima de fondos.

- ✓ Estudio de la organización adecuada para la facturación
- ✓ Revisión e implantación de procedimientos en los departamentos de crédito y cobranzas
- ✓ Planeación fiscal (aprovechamiento máximo de plazos para retenciones y pagos)

Programación de la cobranza. Se analiza el porcentaje de cobranza de las ventas mensuales que se recuperará, contra el transcurso del tiempo

Pronósticos de efectivo. Es la resultante del proceso, para ayudar a la gerencia a solucionar problemas en el manejo de efectivo. Pueden ser de dos tipos:

1. De corto plazo: menores a un año
2. De largo plazo: mayores a un año, pero nunca mayores a tres años (plazo máximo para considerar vigentes las estimaciones del presupuesto).

Presupuesto por áreas y niveles de responsabilidad

Es una técnica de planeación, dirección y control, respecto a la predeterminación de cifras financieras, condiciones de operación y de resultados, encaminada a cuantificar la responsabilidad de los encargados de las áreas y niveles en que se divide la entidad. Expresa el costo pronosticado de las funciones del empleado o funcionario, así como

expresa los méritos o la desviación en la contribución para aumentar los ingresos, reducir los costos, aumentar la eficiencia y alcanzar los objetivos institucionales, departamentales y específicos.

- ✓ Motiva, al permitir que los empleados participen en la fijación de los objetivos del presupuesto y de su área e involucra al personal a participar activamente en la reducción de costos y el incremento de la productividad con la finalidad de lograr mayores ingresos para la organización menos anomalías funcionales y organizacionales y alcanzar los objetivos del propio presupuesto.
- ✓ Ejerce control positivo y permite la comunicación con y entre los empleados.
- ✓ Identifica los ingresos y las erogaciones con las personas que los realizan, dando lugar al reconocimiento personal.
- ✓ Se elaboran los presupuestos con ayuda de las personas que los ejecutan.
- ✓ Hace jueces de sus propias decisiones a quienes las toman
- ✓ Retroalimenta e informa acerca de la actuación de los encargados del área
- ✓ Consolida la estructura organizacional, delimitando autoridad y responsabilidad
- ✓ Los objetivos son susceptibles de cuantificación, en términos de: Moneda, tiempo, unidades de producción, márgenes de contribución, estadística e índices de crecimiento.

Presupuesto por programas

Es la estimación programada en forma matemática de las condiciones de operación de los resultados a obtener por un organismo, en un periodo determinado. Partiendo de la consolidación de una estructura de programas presupuesta los recursos y las acciones necesarias para su cumplimiento, estableciendo: Indicadores, unidades de medida, metas y costos, todos estos elementos deben permitir su evaluación.

Su finalidad general es alcanzar el logro de objetivos y metas con la utilización adecuada de recursos y tiempo. Es un sistema de objetivos múltiples, ya que permite una gran racionalidad en el uso de los recursos disponibles, dándoles cursos alternativos en función de metas precisas propicia una eficiente distribución de recursos relacionando las disponibilidades financieras con las necesidades prioritarias.

Objetivos:

- ✓ Estructurar todos los objetivos y acciones jerárquicamente por funciones y programas de operación en los campos del proceso administrativo.
- ✓ Mayor control interno para conocer y evaluar la eficiencia de cada una de las áreas orgánicas y programas.
- ✓ Racionalizar el gasto, de acuerdo con la eficiencia.

El método de elaborar, ejercer y evaluar en presupuesto público conforme a programas, se basa en tres consideraciones interrelacionadas:

- ✓ El diseño de programas para cada función, definiendo objetivos y metas
- ✓ La determinación de costos para cada programa, y
- ✓ El establecimiento de unidades de medida que permitan evaluar los resultados.

El presupuesto tradicional permite conocer cuánto se gasta y en qué, y también en qué período; el presupuesto por programas, además de lo anterior, permite conocer para qué se gasta al primero le interesa lo que se compra, al segundo lo que se hace.

El Presupuesto por Programas permite comparar el gasto con el grado de consecución del objetivo y por lo tanto, evaluar la relación costo-beneficio, que es en última instancia, el contenido de la eficiencia.

La técnica del presupuesto por programas enfatiza las responsabilidades para la ejecución presupuestaria en términos de calidad, cantidad y costo-beneficio de los resultados, antes que por el destino, que los encargados de la realización de los programas, den a los fondos confiados a su administración.

Presupuesto base cero

Es una metodología de planeación y presupuesto que trata de reevaluar cada año todos los programas y gastos de una entidad organizacional. Establece programas, fija metas y objetivos, toma decisiones relativas a las políticas básicas de la entidad. Se analizan a detalle las distintas actividades que se deben llevar a cabo para instrumentar los programas. Se hace un estudio comparativo de costo-beneficio entre las diferentes alternativas existentes, para cumplir con las actividades. Se selecciona la opción que permite obtener los resultados deseados.

Se ha utilizado con éxito en presupuestos parciales, sobre todo cuando la rutina y la inercia llevan a “arrastrar” partidas presupuestales de uno a otro ejercicio. Cuando se aplica el base cero, destacan claramente las partidas que no son necesarias, lo que permite eliminarlas de inmediato. Se elabora con base en “paquetes de decisión”, que son documentos que identifican y describen actividades específicas, de tal manera que la administración pueda evaluarla y jerarquizarla y decidir su aprobación o rechazo.

La técnica del PBC comprende cinco etapas:

1. Definir la unidad de decisión;
2. Establecer los objetivos de cada unidad de decisión;
3. Formular los paquetes de decisión;
4. Listar los paquetes de decisión estableciendo un orden de prioridades; y
5. Conducir un adecuado control de gestión de la ejecución presupuestaria.

Se entiende por unidad de decisión el nivel institucional responsable de la ejecución de una o un grupo de actividades o funciones, que puede por sí evaluar la marcha de aquellas, su costo y beneficio, y/o someter a consideración superior la necesidad de iniciar nuevos programas de acción.

El responsable de cada jurisdicción u organismo deberá determinar cuáles son los objetivos de la unidad de decisión.

La formulación de los paquetes de decisión es el elemento fundamental de la técnica del PBC, y consiste en identificar una actividad, evaluar los costos y beneficios de llevarla adelante en distintos niveles (el actual, uno mínimo y uno óptimo), evaluar formas alternativas de cumplir el mismo objetivo, y especificar por qué no son utilizados y considerar cuáles serían las consecuencias de no realizar esta actividad. Vale decir que lo que se persigue es presentar todos los elementos necesarios para tomar decisión con respecto a la conveniencia de incluir una determinada actividad en el presupuesto.

Redactados los paquetes de decisión, la siguiente etapa consiste en establecer un orden de prioridades entre ellos.

Una vez que las actividades han sido incluidas en el presupuesto, se debe llevar un control de gestión estricto sobre la ejecución de las mismas que se incluyeron en el presupuesto.

• Evaluación de presupuestos (R)

Estudio detallado de las fortalezas y debilidades que tiene la empresa para enfrentar las compañías rivales por tanto se impone la evaluación de parámetros como objetivos, estrategias, proyectos de inversión, calidad del producto, compañías promocionales, canales de distribución.

La evaluación es la fase que cierra el ciclo y al mismo tiempo lo reinicia, es sin duda una etapa tan importante como la programación misma, ya que una y otra se complementan y se retroalimentan.

La evaluación representa, en tiempo de crisis y carencia de recursos, la posibilidad y la oportunidad de alcanzar beneficios óptimos con recursos limitados. La EVALUACIÓN es un proceso técnico cuyas fases principales son las siguientes:

- ✓ Medición de resultados
- ✓ Comparación con lo programado
- ✓ Análisis de las variaciones
- ✓ Determinación de las causas
- ✓ Definición de medidas correctivas
- ✓ Aplicación de las medidas convenientes
- ✓ Re-programación
- ✓ Medidas Correctivas

4. Planeación financiera

• Análisis de estados financieros (R, E)

Distribución y separación de las partes de un todo hasta llegar a conocer sus elementos o bien es la descomposición de un todo en las partes que lo integran o tan bien se puede decir que es la técnica primaria aplicable para entender y comprender lo que dicen o tratan de decir los estados financieros.

Para analizar el contenido de los estados financieros observar los siguientes requisitos:

Clasificar: los conceptos y las cifras que integran el contenido de los estados financieros.

Reclasificar: los conceptos y las cifras del contenido de los estados financieros.

La clasificación y reclasificación debe ser de cifras relativas a conceptos Homogéneos del contenido de los estados financieros.

Tanto la clasificación como la reclasificaron de cifras relativas a conceptos homogéneos del contenido de los estados financieros debe hacerse desde un punto de vista definido.

El análisis puede ser Interno: Cuando lo realiza una persona que depende de la empresa y Externo Cuando el analista actúa con el carácter de independiente.

Los estados financieros básicos son:

- ✓ Balance general o Estado de situación financiera
- ✓ Estado de resultados
- ✓ Estado de variaciones en el capital contable
- ✓ Estado de cambios en la situación financiera
- ✓ Las notas a los estados financieros complementan los documentos con información relevante sobre todo o parte de ellos

La información contable que se genera en una empresa suele ser extensa, libros contables, auxiliares, pólizas contables. Sin embargo, no toda requiere darse a conocer a los interesados ya que la mayor parte se elabora como herramienta administrativa. Sin embargo, es responsabilidad de la administración e cualquier negocio reflejar en los estados financieros la información esencial, de tal forma que pueda formarse un juicio sobre el futuro de la empresa y tomar decisiones de carácter económico sobre la misma, tales como:

- ✓ El nivel de rentabilidad
- ✓ La posición financiera, que incluye solvencia y liquidez
- ✓ La capacidad financiera de crecimiento
- ✓ El flujo de fondos

El análisis de los estados financieros es un proceso mediante el cual se aplican diversos métodos a los estados financieros y la información complementaria de los mismos, para hacer una medición adecuada de los resultados obtenidos por la administración. Todo ello para poder emitir una opinión correcta acerca de las condiciones financieras de la empresa, la eficiencia de su administración, los hechos económicos que la afectaron y la detección de las deficiencias que deban ser corregidas mediante recomendaciones.

Para poder realizar dicho análisis se debe considerar que:

- ✓ Que los estados financieros estén de acuerdo con los Principios de Contabilidad Generalmente Aceptados (Son conceptos básicos que establecen la delimitación del ente económico, las bases de cuantificación de las operaciones y la presentación de la información financiera cuantitativa por medio de los estados financieros. Emitidos por el Instituto Mexicano de Contadores Públicos).
- ✓ Dictaminados sin salvedades (Si existen salvedades, analizar su efecto sobre la información financiera).
- ✓ Las notas a los estados financieros.
- ✓ El control interno.

El informe de interpretación es el resultado del proceso del análisis financiero, presentado por escrito. Algunas de sus características deben ser:

- ✓ Claro
- ✓ Completo
- ✓ Concreto, conciso
- ✓ Oportuno
- ✓ Evitar abstracciones y generalizaciones

La estructura que puede tener es la siguiente:

- ✓ Encabezado: fecha y a quien va dirigido
- ✓ Introducción
- ✓ Tesis
- ✓ Antítesis
- ✓ Recomendaciones

Tipo de información que se puede obtener del análisis de los estados financieros, a través de las razones financieras:

Liquidez

- ✓ La razón circulante indica la capacidad que se tiene de pagar las deudas, por cada peso de deuda de cuanto se dispone para pagarse.
- ✓ Razón rápida, se refiere a la capacidad de pago de las deudas con efectivo (activo monetario).
- ✓ La rotación de inventarios, es el número de veces al año que se tarda en entrar y salir el inventario, en días, es el número de días en los que se vende una unidad del inventario.
- ✓ Conversión de inventarios en efectivo, es el tiempo en que los inventarios se convierten en efectivo.
- ✓ La rotación de cuentas por pagar, es el tiempo que se tarda la compañía en pagar a los proveedores.
- ✓ El ciclo de financiamiento interno, es el tiempo que la compañía efectivamente financia a los clientes.

Fuerza Financiera

- ✓ El apalancamiento es la forma en que se tienen financiados los recursos, es la relación entre cada peso de recursos contra cada peso de deuda.
- ✓ La razón de capital a fuentes es la que dice como se encuentran financiados los activos, si por dinero propio o deuda con externos.
- ✓ Pasivo a capital, como se ha endeudado la empresa por cada peso aportado o ganado.
- ✓ La cobertura de intereses dice el número de veces en que la utilidad antes de impuestos paga el costo de la deuda.

Eficiencia

- ✓ Razón operativa, se refiere a que por cada peso de ventas, cuantos representan los gastos de operación.
- ✓ Margen de ventas, es la utilidad bruta entre las ventas, sirve para ver como se comporta la eficiencia de la empresa.
- ✓ Apalancamiento operativo dice como por cada peso de venta, la empresa genera utilidad.
- ✓ La rotación de activos dice como un peso de activos (o recursos) genera ventas.
- ✓ Eficiencia operativa, de cada peso de ventas, cuantos se ha pagado en gastos y costos de la operación de la empresa.

Rentabilidad

- ✓ Rendimiento sobre capital, es como cada peso invertido genera de utilidad.
- ✓ Rendimiento sobre activos, dice la relación de como los activos de la empresa generan utilidad.
- ✓ Rendimiento operativo sobre capital invertido, ROIC, dice como cada peso de activos fijos genera utilidades de operación.
- ✓ Rendimiento sobre ventas, dice como las ventas generan utilidad neta, cuantos representan la utilidad neta de las ventas.
- ✓ Valor económico Agregado, EVA, dice cuantos genera la empresa, considerando los costos de oportunidad, es decir, la ganancia de todo lo que se invierte en la compañía.

• Alternativas de financiamiento (R)

1. **Crédito Simple:** Se documenta con pagares que tienen un plazo de pago no mayor de 180 días
2. **Descuento de documentos:** Proviene principalmente de operaciones de compra-venta de mercancías, operario que se denomina descuento mercantil.
3. Prestamos directos sin garantías: Se otorga con base en la solvencia y moralidad del solicitante.
4. **Préstamo de habilitación o avio:** Se otorga a corto y mediano plazo siendo el máximo autorizado de cinco años.
5. **Prestamos Refaccionarios:** Crédito con garantía se otorga a mediano o largo plazo y se emplea para financiar activos fijos, inversiones de mediano o largo plazo.

6. **Crédito Comercial en cuenta corriente:** Se requiere primeramente establecer un contrato para que los bancos correspondientes paguen a un beneficiario por cuenta del acreedor.
7. **Créditos Hipotecarios:** Son Financiamientos a largo plazo de gran flexibilidad para la adquisición, construcción o mejoras de inmuebles destinados al objeto social de la empresa
8. **Préstamo Hipotecario Industrial:** Se otorga con garantía Hipotecaria en primer lugar de la unidad industrial, la que incluye todos los activos tales como el efectivo, cuenta. Por cobrar etc.
9. **Líneas Globales:** Nacional Financiera actúa como agente financiero del Gobierno Federal y como Banca de Fomento Industrial, tiene establecidas líneas de crédito en condiciones preferenciales con organismos y agencias gubernamentales.
10. **Arrendamiento Financiero:** Se establece por medio de un contrato a través del cual el arrendador se obliga a entregar el uso del bien al arrendatario a cambio del pago de una renta durante un cierto periodo

5. **Aspectos fiscales en las organizaciones**

• **Implicaciones fiscales: contribuciones, dividendos, declaraciones y obligaciones (R)**

Contribuciones

Es todo ingreso fiscal que recibe la Federación por parte de las Personas Físicas y Personas Morales, aportando de esta forma el Gasto Público. De acuerdo a lo previsto en el Artículo 2 del CFF se clasifican en impuestos, aportaciones de seguridad social, contribuciones de mejoras y derechos

Dividendos

Las ganancias distribuidas por personas morales residentes en México a favor de sus accionistas o socios, así como los rendimientos distribuidos por las sociedades cooperativas de producción a sus miembros. Artículos 10-A y 120 al 127 LISR.

Declaraciones

Es el documento oficial con el que un Contribuyente presenta información referente a sus operaciones efectuadas en un tiempo específico, a través de las formas autorizadas por la SHCP en términos del Artículo 31º del CFF. Se pueden clasificar de acuerdo a la periodicidad, por ejemplo Declaraciones Informativas de Clientes y Proveedores (anual), declaraciones anuales, trimestrales o mensuales, según el carácter de estas tales como las normales y/o complementarias, según el tipo de Contribuyente (Personas Físicas y Personas Morales) y según el impuesto a declarar (ISR, IMPAC, IVA, RETENCIONES)

Obligaciones

En términos fiscales se puede definir como los vínculos jurídicos existentes entre el Estado y los Contribuyentes, en los que se comprometen a contribuir al gasto público pagando sus impuestos. Artículo 31, fracción IV Constitución Política de los Estados Unidos Mexicanos.

• **Previsión social (R)**

Para los efectos de la Ley del Impuesto sobre la Renta, la previsión social es el conjunto de erogaciones efectuadas por los patrones a favor de sus trabajadores que tengan por objeto satisfacer contingencias o necesidades presente o futura, así como el otorgar beneficios a favor de dichos trabajadores, tendientes a su superación física, social, económica o cultural, que les permitan el mejoramiento en su calidad de vida y en la de su familia.

En esta materia, para que las erogaciones por concepto de gastos de previsión social, es requisito que las prestaciones correspondientes se otorguen en forma general en beneficio de todos los trabajadores.

Tratándose de trabajadores sindicalizados se considera que las prestaciones de previsión social se otorgan de manera general cuando las mismas se establecen de acuerdo a los contratos colectivos de trabajo o contratos ley.

Tratándose de trabajadores no sindicalizados, se considera que las prestaciones de previsión social son generales cuando se otorguen las mismas prestaciones a todos ellos y siempre que las erogaciones deducibles que se efectúen por este concepto, sean en promedio aritmético por cada trabajador no sindicalizado iguales o menores que las erogaciones deducibles por el mismo concepto efectuadas por cada trabajador sindicalizado.

Otros conceptos considerados como gastos de previsión social por la LISR son los Fondos de Ahorro y las Primas de Seguros otorgadas en beneficio del trabajador.

• **El área fiscal en la organización y su relación con el área informática (R)**

El área fiscal en la organización tiene como objetivo monitorear y cumplir de manera satisfactoria con todas aquellas obligaciones fiscales, encontrar los mecanismos más eficientes y las mejores alternativas. Por ejemplo, se requiere de la elaboración y/o revisión de declaraciones mensuales y anuales, de las declaraciones de carácter informativo y la presentación de avisos o solicitudes, todos ellos relacionados con los gravámenes con que se encuentre gravada la empresa.

Debido a lo anterior, la relación con el área informática de la empresa es muy importante, ya que una correcta relación entre estas áreas permite aprovechar las sinergias de ambas, mediante el desarrollo de las mejores y más eficientes herramientas computacionales para el logro de los objetivos del área fiscal.

El correcto y oportuno manejo de la información contable de la empresa, es el principal insumo del área fiscal, por lo tanto, el desarrollo de paquetería funcional, hace del área informática un importante complemento del área fiscal.

III. Tópicos de manejo económico

1. Conceptos básicos de macroeconomía

La macroeconomía es aquella parte de la economía que estudia el comportamiento agregado de los agentes económicos a nivel global y no individual. Algunas cuestiones que estudia son; el empleo, la inflación, el P.I.B., la política económica etc.

• **Banca: central, pública, privada, comercial, de desarrollo (R, E)**

Banca central.

Es una institución pública cuya finalidad primordial es proponer y aplicar la política de monetaria y crediticia de un país, con el objeto de coadyuvar al buen funcionamiento de la economía nacional. La banca central regula el sistema monetario y el volumen de crédito de un país. Las funciones principales de un banco central son:

- a) Mantener y regular la reserva monetaria del país.
- b) Emitir moneda en forma exclusiva.
- c) Fijar las tasa de interés que operan en el sistema monetario.
- d) Regular la circulación monetaria y el volumen de crédito.
- e) Actuar como banco de bancos y cámaras de compensaciones.
- f) Fungir como representante financiero del gobierno ante instituciones financieras internacionales.
- g) Realizar operaciones de mercado abierto.
- h) Administrar la deuda pública.
- i) Constituirse como prestamista del gobierno.

Banco comercial.

Es la institución privada de crédito organizado con el fin de obtener ganancias para sus dueños, mediante la aceptación de depósitos y ahorros del público y la concesión de préstamos con bajo interés, destinados al financiamiento del consumo y la inversión. Una característica importante del banco comercial es su capacidad de crear dinero prestando más dinero del que tiene y de acuerdo con la limitaciones impuestas por las autoridades monetarias del país. Los bancos comerciales forman parte del sistema bancario de un país y operan el mercado en mercado de dinero, ya que realizan operaciones a corto plazo.

Banco privado.

Es la institución que profesionalmente realiza el ejercicio de la banca y cuyo capital pertenece en su totalidad a los accionistas privados del banco; forma parte del sistema bancario del país y el estado solo regula sus actividades. A los bancos privados se les llama también bancos comerciales y están dedicados a la intermediación masiva y profesional en el uso de créditos y en las diversas actividades bancarias. La finalidad de los accionistas es obtener ganancias mediante la captación de depósitos y ahorros del público y la concesión de préstamos destinados al consumo y la inversión. Los bancos públicos tienen la capacidad de crear dinero hasta cierto límite fijado por las autoridades a través del canje legal o depósito bancario.

• **Situación económica de México (R)**

2. Economía de la empresa

• **Proyectos de inversión (E, RP)**

Las empresas maximizan el valor de los accionistas a través de la mejor combinación de las decisiones de inversión, financiamiento y dividendos. Las decisiones de inversión se refieren a la colocación de capital en proyectos de inversión viables con expectativas de retornos futuros.

Un proyecto de inversión se define como la unidad básica de planeación que se conforma por una serie de actividades entrelazadas, que se suceden y complementan en la toma de decisiones referente a la inversión de la empresa.

Los proyectos de inversión se clasifican en:

- a) De acuerdo a su naturaleza
- b) De acuerdo a su relación con otros proyectos de inversión

De acuerdo a su naturaleza se clasifican en:

1. **Crecimiento.** Son inversiones que buscan hacer crecer en una misma rama de negocios. Ejemplo.- Una fábrica de refrescos decide comprar una productora de jugos. Ambos productos son líquidos. Esta clase de proyectos de inversión también se le conoce como proyectos de expansión.
2. **Diversificación.** Son proyectos de inversión que buscan crecer en distintas ramas económicas. Ejemplo.- Una fábrica produce refrescos y jugos y se decide realizar un proyecto de inversión para incursionar en la elaboración de galletas.
3. **Mantenimiento.** Son proyectos de inversión destinados a sustituir el equipo existente. Ejemplo.- Se realiza un proyecto de inversión para sustituir el equipo actual que se utiliza para el llenado en los refrescos.
4. **De inversión regulatoria.** Son los de regulación gubernamental. Ejemplo. Los proyectos de inversión ecológica, tratamientos de agua, mudanzas, etc. Muchas veces estos proyectos no generan beneficios económicos pero se tienen que hacer.
5. **De inversión social.** Este tipo de proyectos están orientados a ofrecer un bienestar a la comunidad. Ejemplo.- Sistema de transporte colectivo.

De acuerdo a su relación con otros proyectos de inversión se clasifican en:

1. **Independientes.** No influye la realización de un proyecto de inversión en otro proyecto de inversión.
2. **Mutuamente excluyentes.** Se realiza la inversión en un proyecto u otro proyecto pero no al mismo tiempo.
3. **Complementarios.** Son los proyectos de inversión que complementan a procesos operativos.

La elaboración de proyectos de inversión se sustenta en cuatro estudios:

1. **Socioeconómico.** Este estudio comprende el comportamiento de las variables macroeconómicas, componentes sociales como sindicatos, nivel de ingresos, control de cambios, esquemas de fomento de exportaciones, etc.
2. **Mercado.** Identificar al posible consumidor, mercados futuros y todo lo relacionado a la mezcla de mercadotecnia como precio, plaza, producto, promoción y personas.
3. **Producción.** Identifica las características técnicas como capacidad utilizada, materia prima, tecnología, etc.
4. **Financiero.** Se realiza una proyección de estados de resultado, flujo de efectivo con el propósito de hacer una evaluación financiera del proyecto de inversión.

Existen dos técnicas de evaluación de proyecto:

- ✓ Las técnicas que no consideran el valor del dinero a través del tiempo:
 1. Período de recuperación (PayBack)
 2. Retorno sobre la inversión (ROI)
- ✓ Las técnicas que consideran el valor del dinero a través del tiempo.
 1. Valor presente neto (VPN)
 2. Tasa interna de rendimiento (TIR)
 3. Índice de lucratividad
 4. Período de recuperación a valor actual

La correcta valuación de proyectos de inversión genera los siguientes beneficios:

- ✓ Una mejor asignación de los recursos financieros de la empresa.
- ✓ Maximizar el Valor de los Accionistas.
- ✓ Establecer programas de expansión exitosos.
- ✓ Diversificar el riesgo.
- ✓ Fortalecer la relación accionista-empresa.
- ✓ Detectar oportunidades de negocio.
- ✓ Aumento en el precio de la acción. (En caso de que la empresa sea pública)

• Exportación e importación (R)

Los países comercian entre ellos por que pueden comprar bienes más baratos en el extranjero que en el interior. El comercio internacional permite que se incremente el consumo de bienes por dos razones: primero, la existencia de diferencias en cuanto a la disponibilidad de materias primas y otros factores de producción y a la tecnología significa que existen diferencias internacionales entre los costos de producción. El comercio permite que a los países especializarse en los bienes que pueden producir más baratos.

Esto se puede llevar a nivel empresarial, es decir, algunas empresas tienen mayor capacidad para producir ciertas cosas que otras, por lo que el comercio internacional les permite especializarse en lo que son mejores. Esto significa que las empresas aprovechan la ventaja comparativa derivada de la especialización en la producción de bienes que se pueden fabricar a un costo relativamente más bajo.

El tipo de restricción del comercio más frecuente es el arancel o impuesto sobre las importaciones, según el cual el importado de un bien tiene que pagar al Estado una proporción específica del precio mundial. Estas restricciones se han creado como métodos de protección para los productores nacionales. Existen otras políticas comerciales restrictivas como: contingentes, barreras no arancelarias y subvenciones.

• Empresas de bienes y servicios informáticos (R, E)

Actualmente, el mundo de los negocios está envuelto en un proceso de globalización que ha dado origen a un entorno altamente dinámico, cambiante y extremadamente competitivo.

Para afrontar con éxito estos nuevos retos, la empresa que quiera ser competitiva y encaminada a ofrecer el mejor servicio a sus clientes, debe maximizar el rendimiento de sus recursos, mediante un adecuado sistema de información que integre y controle la gestión de todos sus departamentos, de forma global y con la rapidez y eficacia que el mercado demanda.

Esto puede darse a través de aumentos en la productividad y calidad, reducción de los costos, y de gestionar de una forma eficaz y segura la información de todas las áreas de la empresa (financiera, producción, recursos humanos, compras, ventas, etc.).

La idea de la informatización de las empresas se debe encaminar a crear la comunicación necesaria y específica para el análisis, desarrollo, implantación y reingeniería necesaria del sistema de información de forma personalizada, así como crear los mecanismos adecuados para asegurar el mantenimiento posterior.

B. El área de informática

I. La función informática

1. Área de informática en las organizaciones

- Visión y misión (R)

Visión.

Como área de servicio y soporte administrativo buscar siempre el liderazgo. Para lograr este fin debemos promover y apoyar, en forma continua, mejoras en los procesos y políticas administrativas acordes al desarrollo tecnológico global.

Misión.

Ofrecer sistemas de información administrativa integral que permita en forma oportuna satisfacer necesidades de información, como apoyo en el desarrollo de las actividades propias de cada departamento.

- Función y objetivos (R)

Funciones.

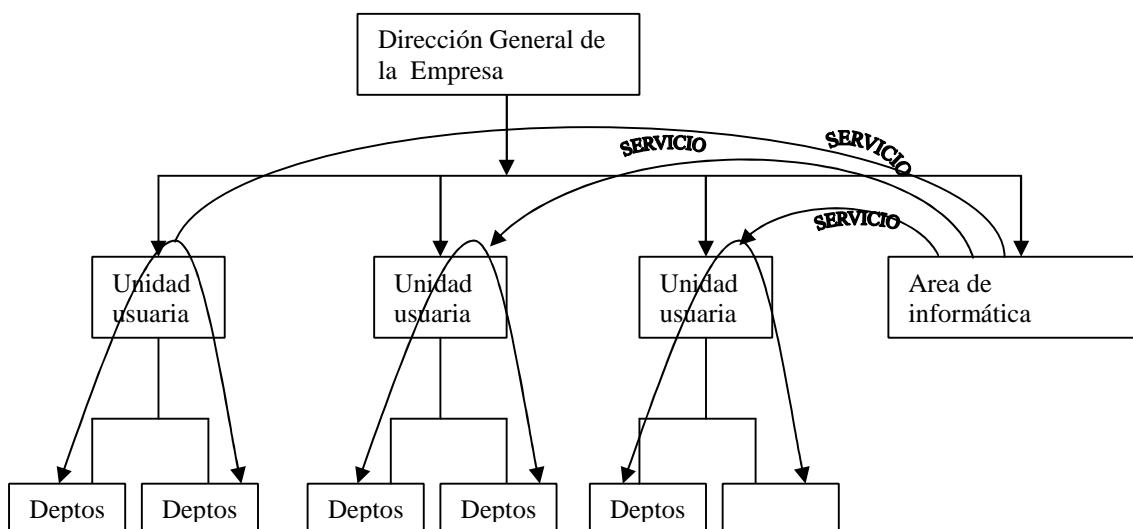
- ✓ Proporcionar los servicios de informática en apoyo a la organización.
- ✓ Analizar, desarrollar y proponer sistemas que permitan un mayor aprovechamiento del equipo informático.
- ✓ Operar y conservar el equipo informático.
- ✓ Proponer la simplificación de sistemas y procedimientos en materia de informática.
- ✓ Implantar y mantener el sistema de información integral, mediante métodos computarizados
- ✓ Apoyar el procesamiento electrónico de la información.
- ✓ Apoyar la función administrativa en materia de informática.
- ✓ Asesorar a las dependencias tanto en la adquisición de equipos de cómputo como en soporte técnico.

Objetivos.

Proporcionar con oportunidad y eficiencia el servicio de informática a las áreas que integran la organización, así como optimizar el aprovechamiento de los sistemas de información mecanizados, a efecto de ayudar en la toma de decisiones para el cumplimiento de las metas organizacionales.

- Ubicación en la organización (R)

Los recursos computacionales se encuentran administrados y organizados como parte de la alta gerencia de las organizaciones y es la herramienta principal en la toma de importantes decisiones para el desarrollo de la empresa.



• **Organización y funciones del área de informática (R)**

El área de informática cuenta con los siguientes departamentos:

- ✓ **Análisis y Diseño de sistemas.** Se encarga de definir los programas de cada sistema. Elabora un documento estructurado en donde se encuentra lo concerniente a cada programa y lo envía al departamento de programación, el cual esta conformado por 5 grupos de analistas especializados por tipo de sistema aplicativo, cada uno atiende sistemas locales, on line y micros.
- ✓ **Unidad de programación.** Se estudia la definición de cada programa y se aclaran dudas con el analista. Se procede a elaborar el programa y se regresa al analista una carpeta con los resultados de la ultima prueba y la lista fuente del programa. El cual esta conformado por: Programadores batch, on line, de micros de sistemas operativos de data base y mesa de control.
- ✓ **Unidad de operación.** Mantiene estrecha comunicación con los analistas de sistemas y con los usuarios para determinar los calendarios de producción. Esta conformado por Operadores especializados en periféricos, computadora y red on line, Operadores de sistemas, Control de la producción, Operadores del centro de captura e Ingeniería de instalaciones.
- ✓ **Sopporte técnico.** Se encarga de mantener en operación el sistema operativo y de asesorar técnicamente a los demás departamentos. El cual consta de Auditoria de sistemas y Asesoría a usuarios.

• **Estructuras informáticas centralizadas, descentralizadas, híbridas e independientes (R)**

- ✓ **Centralizadas.**- Todos los departamentos comparten la misma base de actividad e información.
- ✓ **Descentralizadas.**- Cada departamento mantiene sus datos de forma individual.
- ✓ **Híbridas.**- Combinación de estructuras Centralizadas-Descentralizadas.
- ✓ **Independientes.**- Cada departamento mantiene sus datos de forma independiente.

Área de informática descentralizada

Ventajas

- ✓ Racionalización del uso de los recursos de cómputo.
- ✓ Análisis detallado del costo-beneficio de las inversiones en los proyectos de informática.
- ✓ Satisfacción de las verdaderas necesidades de información de las unidades usuarias.
- ✓ Efectivo direccionamiento de la reingeniería de los sistemas de información.
- ✓ Posibilidades reales y costeables para establecer una cultura informática a nivel de la empresa.
- ✓ Capacidad informática para adaptarse rápidamente a las variantes de los negocios y objetivos de la empresa.
- ✓ Verdadero apoyo informático al plan de modernización de la empresa y su crecimiento futuro.
- ✓ Verdadero enfoque de la informática como un arma estratégica de competencia de la empresa.

Área de informática centralizada

Desventajas

- ✓ Despilfarro computacional
- ✓ No existen
- ✓ Muchos sistemas improvisados que no cubren plenamente las necesidades de información de los usuarios
- ✓ Confusión y descontrol
- ✓ Pocas posibilidades para el logro de la cultura informática de la empresa y demasiado desperdicio de esfuerzo y dinero.
- ✓ Imposibilidad absoluta de adaptación.
- ✓ Insuficiencia de apoyo para el plan de modernización de la empresa.
- ✓ Tendencia al rezago como una estrategia de competencia para la empresa.

• **Vínculos formales e informales del área de informática con los demás órganos (R)**

Se vincula de manera formal con las áreas donde se utilizan equipos de cómputo por necesidad y de manera informal con cualquier órgano que requiera la utilización de datos procesados para una toma de decisiones o como herramienta de apoyo en el cumplimiento de una actividad.

2. Empresas y proyectos informáticos

• **Tipos de empresas de servicios informáticos (R)**

Configuración e instalación de equipos

- ✓ Adecuación de la configuración de los equipos

- ✓ Instalación y personalización de software

Redes y comunicaciones

- ✓ Ingeniería de Proyecto
- ✓ Cableado Estructurado
- ✓ Asesoría, diseño, instalación y diagnóstico
- ✓ Instalación de sistema operativo y servicios de red. Configuración de dispositivos de conectividad LAN - WAN.
- ✓ Comunicaciones y Análisis a nivel protocolos y servicios de red. Conectividad local y remota

Consultoría.

- ✓ Aplicaciones de oficina y especializadas.
- ✓ Bases de Datos.
- ✓ Conectividad, Internet/Intranet y Seguridad en Internet.

Capacitación.

- ✓ Cursos a la medida y de especialización
- ✓ Atención para administradores y usuarios

Soporte Telefónico.

- ✓ Control de atención mediante levantamiento de Reportes de Servicio
- ✓ Asesoría telefónica.

• **Principios rectores para el desarrollo de tecnología nacional: adaptación, adecuación, transferencia, investigación, autonomía, independencia e inventos (R)**

• **Empresas pequeñas: la figura de los socios-técnicos; la figura de los dueños directores (R)**

Los Socios-técnicos.

Empresa en la cual los creadores tienen conocimientos de la actividad técnica y desarrollan las funciones operativas.

Los dueños-directores.

Los propietarios de la pequeña empresa dirigen y coordinan la dirección.

• **Recursos y servicios externos (outsourcing) (R)**

En los últimos años, en las empresas productoras de Bienes y Servicios se fue incrementando un cambio hacia la utilización de recursos y servicios externos para reemplazar la realización de actividades cuya ejecución ha sido realizada tradicionalmente con la estructura propia de la Organización.

Outsourcing: La palabra Outsourcing, hace referencia a la fuente externa de suministro de servicios; es decir, la subcontratación de operaciones de una compañía a contratistas externos. Con frecuencia se recurre al Outsourcing como mecanismo para reducir costos. Dicha subcontratación ofrece servicios modernos y especializados, sin que la empresa tenga que descapitalizarse por invertir en infraestructura.

Más allá de parecer un gasto adicional, la subcontratación representa la oportunidad de dejar en manos de compañías expertas la administración y el manejo eficiente y efectivo de los procesos que no están ligados directamente con la naturaleza del negocio y que, por el contrario, permite reducir costos y reorientar los recursos internos e influir de manera significativa en su nivel de competitividad.

Este servicio es una relación de largo plazo y va más allá del apoyo puntual, como sucede en el caso de la Consultoría Legal. En la medida que el volumen de transacciones de una empresa aumenta, aparece la oportunidad de procesarlas afuera para hacer más flexible la operación y es allí donde se empieza a dar el verdadero outsourcing.

El outsourcing podría definirse como un servicio exterior a la compañía y que actúa como una extensión de los negocios de la misma, pero que es responsable de su propia administración. También se podría definir como la

acción de recurrir a una agencia exterior para operar una función que anteriormente se realizaba dentro de una compañía.

Es importante mencionar otros de los beneficios proporcionados por esta herramienta, los cuales son:

- ✓ La proporción de mayor veracidad de respuestas
- ✓ Protección de inversiones perdidas
- ✓ Evitar quedar "atrapados" con el proveedor
- ✓ Reducción del riesgo del negocio

• Identificación de necesidades (E)

Si bien en los últimos años se ha incrementado en forma significativa el uso de las tecnologías de la información o informática en el país, esta situación no es generalizada y se observan diferencias notables en algunos sectores. Adicionalmente, existen deficiencias en cuanto a la infraestructura nacional, que han propiciado inefficiencias en los procesos de adopción y aprovechamiento de la informática.

En el sector público es notable el aumento de inversión en tecnologías de la información. El rezago que existía ha permitido, además, que las inversiones recientes se traduzcan en un parque instalado relativamente más moderno que el de otros países; sin embargo, la informática no se está explotando adecuadamente. Además, no puede dejar de considerarse que existen diferencias relevantes entre las dependencias y entidades de la Administración Pública Federal, y retraso en algunos gobiernos estatales y en prácticamente todos los municipales.

Como consecuencia de una oferta más abundante derivada de la apertura comercial, y de las mismas necesidades de inversión tecnológica para modernizarse también, el sector privado ha aumentado en forma considerable la demanda de bienes y servicios informáticos; en especial, ésta ha crecido debido a los requerimientos de los grandes grupos industriales, comerciales y financieros. Sin embargo, en las pequeña y mediana empresas la inversión en informática es prácticamente nula.

En cuanto al uso extensivo de la informática en la sociedad, el índice de computadoras por habitante y por hogar es bajo respecto a otros países, a pesar de que existe una percepción generalizada de los beneficios que brinda esta tecnología.

Por lo que respecta a la infraestructura informática, es claro que México requiere especialistas que manejen adecuadamente la tecnología. También se necesita impulsar actividades de investigación que permitan dotar al país de la capacidad de innovación y asimilación tecnológica. Además, es importante desarrollar los nichos de industria que satisfagan las necesidades del mercado y contar con redes de datos para transmisión y acceso a información.

La carencia de especialistas informáticos calificados es el aspecto de mayor preocupación, por ser la base fundamental para garantizar el sano aprovechamiento y desarrollo de la informática en el país. Aunque existe una gran cantidad de programas de estudio en los niveles técnico y de licenciatura, su calidad es muy heterogénea. En el nivel postrado, base para generar especialistas que sustenten la educación en los otros niveles, existen graves deficiencias en cuanto a la cantidad y calidad de los programas.

Las actividades de investigación y desarrollo tecnológico son también muy limitadas. Existen pocos grupos, dispersos geográfica y temáticamente, y la inversión pública y privada en esta materia es escasa.

En materia de redes de datos, existe un fuerte rezago en infraestructura que impide soportar la creciente demanda de servicios de transmisión y acceso a información, motivada por el gran potencial generado por las continuas innovaciones en el área de telecomunicaciones. Las recientes modificaciones normativas y la creciente participación de nuevos oferentes abren, sin embargo, la oportunidad de mejorar esta situación y de fundamentar la creación de servicios que faciliten la consulta de información que se conocen como de valor agregado.

En cuanto a los aspectos normativos, existen diversas disposiciones jurídicas que rigen a la informática. Las modificaciones realizadas recientemente, permiten contar con un marco más flexible y abierto. Sin embargo, la evolución tecnológica hace necesario actualizar las disposiciones jurídicas que inciden en el área.

En resumen, es clara la necesidad de realizar acciones en distintos ámbitos para lograr un desarrollo informático que permita asimilar y adecuar esta tecnología a las necesidades del país.

- **Definición de productos y servicios para el mercado (oportunidades, nichos, mercado horizontal y mercado vertical) (R, E)**

Mercado vertical.

El fabricación del hardware del dispositivo, el software de control del dispositivo, las aplicaciones, así como las ventas del dispositivo son realizadas generalmente por una misma compañía.

Mercado Horizontal.

La evolución de las tecnologías informáticas supuso una clara separación entre el hardware y las aplicaciones que se ejecutan sobre dicho hardware. La revolución del PC al cambiar la estructura de mercado vertical a horizontal supuso un incremento exponencial en la industria informática, fruto de la competitividad y de la interoperabilidad de los sistemas.

- **Desarrollo de proyectos informáticos (E, RP)**

Proyecto: es la búsqueda de una solución inteligente al planteamiento de un problema tendiente a resolver, entre muchas, una necesidad humana.

Proyectos Informáticos Nacionales

Los Proyectos Informáticos Nacionales son un instrumento para coordinar esfuerzos y recursos en la solución de una problemática nacional, en la cual las tecnologías de la información puedan tener una repercusión trascendente.

Este instrumento permitirá promover una vinculación efectiva entre el gobierno, las empresas y el sector académico alrededor de un proyecto en el cual se detecten oportunidades objetivas de innovación tecnológica.

En general, los Proyectos Informáticos Nacionales se identifican como tareas en que se requiere una participación interinstitucional y que tienen como elementos esenciales los siguientes:

- ✓ Atender una problemática importante del país, como contribuir a mejorar la calidad de vida de los mexicanos, la productividad en un sector o la competitividad del país.
- ✓ Que las tecnologías de la información constituyan un elemento fundamental y existan oportunidades de innovación científica o tecnológica que repercutan en el mejoramiento o la creación de procesos, productos o servicios competitivos internacionalmente.
- ✓ Que posibiliten ahorros o rendimientos importantes con la conjunción de esfuerzos, debido al monto de las expectativas de inversión o gasto en equipamiento, programas para computadora, integración de sistemas o mantenimiento.
- ✓ Que la participación coordinada de los sectores sea necesaria o benéfica.

Dentro de los Proyectos Informáticos Nacionales se realizarán distintos trabajos como:

- ✓ Estudios de factibilidad, términos de referencia, especificaciones técnicas, propuestas de difusión de resultados, evaluaciones de impacto, estudios de explotación, e inclusive proyectos piloto.
- ✓ Desarrollo de tecnología que permita innovaciones en la generación de soluciones o aportaciones originales, para llevar tecnología de punta a las prácticas cotidianas.
- ✓ Operación y mantenimiento de sistemas y productos resultantes.
- ✓ Los Proyectos Informáticos Nacionales atenderán a diversas líneas temáticas, de las cuales son ejemplo las que se enlistan a continuación:
 1. Aprovechamiento de las tecnologías de la información para la educación y la capacitación, considerando el diseño y desarrollo de aplicaciones como sistemas de enseñanza interactivos, aplicaciones multimedia en red, trabajo en grupo y servicios de documentación.
 2. Servicios telemáticos públicos, considerando diseño y desarrollo de aplicaciones en red, intercambio electrónico de datos, seguridad de la información y herramientas de explotación.
 3. Sistemas de información de instituciones públicas, considerando el diseño y desarrollo de bases de datos, interoperabilidad de aplicaciones, seguridad de la información en redes y desarrollo de herramientas de explotación.

4. Aprovechamiento de la informática en las empresas, considerando el diseño y desarrollo de aplicaciones modulares, sistemas de información en red, herramientas para prestación de servicios, intercambio electrónico de datos para la realización de transacciones financieras y comerciales, tecnologías de automatización industrial y de control digital de procesos.
5. Bibliotecas electrónicas de acceso público en red considerando el diseño y desarrollo de bases de datos, aplicaciones multimedia, tecnologías de almacenamiento y de acceso a información.

En particular, dentro de los Proyectos Informáticos Nacionales que han sido identificados en las diversas fases de elaboración del Programa de Desarrollo Informático, a través de la consulta con las diversas dependencias y sectores, están:

- ✓ Tecnologías de la información en el sector educativo: Incorporación de las tecnologías de la información en el sistema educativo nacional para mejorar los métodos de enseñanza y materiales de apoyo, proporcionar servicios de educación y capacitación a distancia, así como para capacitar profesorado.

Coordinador: SEP

Participantes: STPS, CONACyT, CONALEP, CONAFE, gobiernos estatales.

Sistema de compras gubernamentales: Red de compras gubernamentales para mejorar los procesos de adquisiciones del sector público, mediante un sistema en red para publicación de bases de licitación, inscripción de ofertas, publicación de resultados y seguimiento de adjudicaciones.

Coordinador: SECODAM

Participantes: SCT, SECOFI, SHCP, gobiernos estatales y municipales, CAIAP, CIAPEM, CONCANACO, CONCAMIN y COPARMEX.

Sistema de información de la Administración Pública: Sistemas de bases de datos de información de la Administración Pública que permitan desarrollar una infraestructura de información gubernamental de que pueda disponer la población a través de una red o en un medio magnético u óptico.

Coordinador: INEGI

Participantes: Todas las dependencias y entidades de la Administración Pública Federal, Banco de México, gobiernos estatales y municipales, CAIAP y CIAPEM.

Sistema Nacional de Información sobre Seguridad Pública: Para apoyar a las autoridades federales, estatales y municipales en la planeación, en la toma de decisiones en la materia y para propósitos de seguridad pública, se desarrollarán bases de datos sobre personal de seguridad pública, armamento y equipo, delitos y responsables, estadística general de seguridad pública, así como servicios de atención a la ciudadanía.

Coordinador: SEGOB

Participantes: PGR, SECODAM, SCT, SEDENA, SEMAR, PGJDF, DDF, gobiernos estatales y municipales, procuradurías de los estados y tribunales superiores de justicia.

Sistema para pequeñas y medianas empresas: Incorporación de las pequeñas y medianas empresas a una red informática nacional que les permita aportar información sobre insumos y mercados, y realizar transacciones con proveedores y clientes.

Coordinador: SECOFI

Participantes: SCT, CONACyT, NAFIN, cámaras de comercio e industriales, asociaciones informáticas.

Sistema para seguridad social: Apoyo a servicios públicos como seguridad social mediante el uso de tecnologías de la información para simplificar los trámites de los usuarios, así como la administración y el control hospitalario.

Coordinador: IMSS

Participantes: SS, cámaras empresariales.

Proyectos Informáticos Regionales

Los Proyectos Informáticos Regionales son un mecanismo instituido en este Programa, con la finalidad de facilitar esfuerzos de colaboración para llevar a cabo proyectos que responden a una problemática en el ámbito local, pero que tienen repercusiones nacionales.

Los Proyectos Informáticos Regionales permiten responder a necesidades propias de los gobiernos estatales y municipales que trascienden el ámbito estrictamente local, por corresponder a problemas comunes que requieren

una adecuada coordinación, entre iguales o diferentes órdenes de gobierno. Tal es el caso de los sistemas catastrales, los registros públicos de la propiedad, mercantiles y civiles, algunos sistemas de seguridad pública, control vehicular, manejo de recursos naturales e impacto ambiental, entre otros.

En este tipo de proyectos resulta fundamental establecer acuerdos que permitan la colaboración directa y complementaria en el proceso de diseño, construcción, explotación de sistemas o en la adopción de ciertas normas técnicas.

Los Proyectos Informáticos Regionales permitirán identificarlos sistemas requeridos, analizar sus posibles tratamientos técnicos, administrativos y jurídicos, establecer estrategias concertadas de trabajo y detectar oportunidades de desarrollo y empaquetamiento. En este caso, también se tendrán instancias de seguimiento.

• **Fuentes de financiamiento para proyectos informáticos (E, RP)**

El Programa de Desarrollo Informático será financiado mediante aportaciones que serán administradas con base en la normatividad vigente que corresponda.

Las aportaciones federales provendrán de los recursos que las dependencias y entidades participantes asignen para llevar a cabo, dentro de sus Programas Operativos Anuales, los proyectos y actividades que corresponden a las acciones definidas en el Programa. Por ello, deberán considerar dentro de sus previsiones presupuestales los recursos requeridos para llevarlos a cabo.

Asimismo, para la ejecución del Programa, se podrán considerar los fondos federales previsibles dentro de los convenios de desarrollo social entre la Federación y los gobiernos estatales, así como los fondos estatales y municipales previstos para actividades relacionadas con los proyectos definidos.

Para realizar acciones en las que participan diversas instituciones de la Administración Pública Federal, Estatal y Municipal, como los Proyectos Informáticos Nacionales y Regionales, se promoverá el establecimiento de mecanismos que permitan conjuntar recursos provenientes de los diferentes fondos mencionados.

Por lo que respecta a las aportaciones privadas, se promoverán mecanismos de concertación que permitan propiciar la inversión del sector industrial en los distintos proyectos y acciones del Programa, considerando sus intereses, así como los beneficios que puedan derivarse de su participación.

Este esquema permitirá establecer compromisos presupuestales para realizar proyectos tecnológicos más complejos y de mayor calidad técnica, propiciando un mejor aprovechamiento de las inversiones de los sectores público y privado, así como mayores rendimientos sociales.

3. Administración del área de informática

• **Planeación del área (RP)**

El manejo de sistemas de información es tan importante para los planes de las empresas, que en ocasiones se contratan gerentes con la única responsabilidad de administrar estos sistemas. Comprar la tecnología adecuada es la parte fácil, el reto es adecuar la tecnología a las necesidades de la organización. Alcanzar un alto grado de adecuación es un aspecto fundamental para el éxito de la compañía. Cualquier decisión para intervenir en alguna aplicación en particular significa mas que un compromiso de tiempo, esfuerzo y recursos financieros. La decisión también define el futuro de la empresa, dado que al concentrar los esfuerzos en una aplicación se limita la atención que puede darse a otros proyectos. Las aplicaciones seleccionadas deben ser aquellas que brinden los mayores beneficios para la compañía.

Planeación y administración de recursos.

El objetivo de los centros de cómputo debe ser procesar los sistemas de información de la empresa en forma tal que se obtengan los resultados esperados de manera confiable, oportuna y con el mínimo de recursos utilizados, para lo cual se relacionan 13 normas que coadyuvarán a este objetivo.

1. El jefe del área de informática es el directamente responsable de la administración y utilización de los recursos humanos, materiales, técnicos y tecnológicos de dicha área.

2. El jefe del área de informática debe prever todos los recursos que requieran para la correcta utilización de los servicios.
3. El jefe de cada área de trabajo deberá elaborar su presupuesto de operación, así como sus programas de trabajo.
4. Cada jefe de área será responsable de crear y mantener un ambiente de trabajo adecuado para que se cumpla totalmente con los objetivos.
5. Es responsabilidad del centro de cómputo impartir cursos y dar asesorías para actualizar a los usuarios en la utilización del recurso de cómputo.
6. El centro de cómputo debe realizar evaluaciones operativas y auditorías periódicas a la utilización de los servicios de informática.
7. El centro de cómputo debe implementar un sistema de información que permita conocer datos estadísticos y de control respecto a la utilización de los recursos materiales, técnicos y financieros de todas y cada una de las áreas involucradas en los servicios informáticos.
8. Es responsabilidad del centro de cómputo comunicar a los usuarios del equipo su disponibilidad para su utilización.
9. Los servicios de uso de estaciones de trabajo remotas o en red deben ser proporcionados de acuerdo a los horarios rigurosamente congruentes con los del equipo de cómputo.
10. Es obligación del usuario de las estaciones de trabajo el controlar la utilización del recurso utilizando las bitácoras de operación.
11. Si el usuario requiere utilizar las estaciones de trabajo en horarios no considerados deberá solicitar por escrito y por los conductos especificados.
12. Cuando los usuarios trabajan a nivel de pruebas, éstas deberán ser manejadas de tal forma que sobre utilicen innecesariamente los recursos del computador.
13. El área de operación deberá poner en marcha el proceso de eliminación de archivos fuera de estándar y capacitación para generar espacio.

Con el objeto de llevar un control racional de la utilización del equipo de cómputo, así como de sus periféricos y dispositivos magnéticos, es importante establecer procedimientos rutinarios de depuración de archivos para generar espacio y eliminar lo utilizado por única vez.

Los archivos de cómputo deben ser inventariados y controlados a través de los registros específicos, que contengan; nombre del archivo, fecha de creación, programa que lo genera, vigencia de su información, total de registros, etc.

La planeación es un proceso intelectual que requiere ser reforzada continuamente por un sistema de retroalimentación que evalúe el avance y sus resultados.

Planeación a largo plazo para los centros informáticos

- ✓ Deberán ser planeadas de modo que sus objetivos a largo plazo y corto plazo sean consistentes con las metas de la organización.

Comité de planeación –vigilancia para los sistemas de información.

- ✓ Supervisa el nivel de experiencia de los departamentos usuarios.
- ✓ Participa en la evaluación de los proyectos a desarrollar.
- ✓ Participa en la toma de decisiones en la adquisición de recursos.

Planeación de sistemas de información a largo plazo

- ✓ Debe considerar a los objetivos de la empresa, reconocer las metas y las políticas en la organización, los avances tecnológicos y los requerimientos gubernamentales.

Planeación a corto plazo para los sistemas de información

- ✓ Los planes a corto plazo deberán ser consistentes con los planes a largo plazo de la organización.
- ✓ Deben considerar a los programas de trabajo y presupuestos.

Revisión de la planeación para los sistemas de información

- ✓ Diseñar un sistema de información que permita el control de las funciones.
- ✓ Definir reportes gerenciales, avance del logro de las metas, gastos actuales contra presupuesto, productividad.

Alcance de la planeación contra desastres

- ✓ Debe abarcar tanto aplicaciones en proceso de desarrollo como las ya implantadas.
- ✓ Se debe considerar la documentación de los sistemas, la programación y la operación así como los equipos, instalaciones, datos, archivos y papeleería.

• Normas, estándares de operación y controles (E, RP)

La **norma** es una regla o medida establecida a la que se tienen que sujetar los involucrados para el cumplimiento del objetivo, prescribe los límites generales dentro de los cuales han de realizarse las actividades que regula.

Cinco normas básicas:

1. Es obligación de todas las áreas que integran la organización conocer los Manuales de Normas y Procedimientos para utilizar los servicios informáticos.
2. Es obligación del personal que integra la organización dar cumplimiento a las normas y procedimientos descritos en el Manual de Normas y Procedimientos. Habrá un solo original del manual bajo la custodia del centro de cómputo, el cuál se encargará de su difusión a todas las áreas de la organización que utilicen los servicios informáticos, así como de su actualización en caso de modificaciones.
3. Es obligación del centro de cómputo documentar los procedimientos y normatividad.
El éxito o fracaso de este manual depende en gran medida de la difusión que se haga del mismo, su desconocimiento provoca que no se cumplan y por consecuencia un mal servicio informático, permanentemente se deben programar acciones como:
 - a. Preparar presentaciones o juntas de coordinación con los usuarios.
 - b. Hacerlo del conocimiento del personal de nuevo ingreso.
 - c. Adición, supresión y modificación de normas y procedimientos.
4. Es obligación del centro de cómputo obtener la oficialización de dichas normas y procedimientos.
5. Se debe contar con una guía estandarizada que permita al usuario proponer adiciones, bajas y modificaciones a las normas y procedimientos para adecuarlas a la realidad

Todo el personal usuario a los servicios informáticos que considere que las normas y procedimientos no cumplen con sus expectativas deberá:

- a. Determinar que norma o procedimiento no cumple su cometido.
- b. Definir una nueva norma.
- c. Obtener el visto bueno de su jefe.
- d. Elaborar una propuesta de modificación

Los encargados de la normatividad del área de Informática deberán:

- a. Recibir la propuesta de actualización
- b. Analizar y evaluar la propuesta recibida.
- c. Considerar que la norma propuesta no afecte actividades y funciones de otras áreas usuarias.
- d. En caso de proceder, obtener la autorización para la integración al manual de Normas y Procedimientos.
- e. Actualizar los manuales que se encuentran en todas las áreas usuarias.
- f. Difundir a los involucrados de la nueva normatividad.
- g. En caso de que no proceda, describir las razones por las cuales no procede y devolver al usuario solicitante su propuesta.

Las **políticas** son guías de acción y del pensamiento, son criterios que orientan como se debe formular e interpretar la normatividad (se hace por única vez)

Los **estándares** son modelos o pautas a seguir para normalizar los procedimientos y poder crear un control sobre las funciones.

Estándares de trabajo que regulen:

- ✓ La adquisición de equipos.
- ✓ Elaboración de presupuestos y control de insumos
- ✓ El ciclo de vida de desarrollo de sistemas
- ✓ Modificación e implantación del software operativo.
- ✓ Operación de los sistemas de información
- ✓ Enlace con los usuarios.

Control de aplicaciones.

Se relacionan 11 normas que describen detalladamente la responsabilidad en la operación de los sistemas de información, explican con claridad cómo las diferentes áreas que conforman el centro de cómputo deben seguir los procedimientos para que el servicio que se solicite sea más eficiente, para mantener una comunicación permanente con el usuario.

Toda la comunicación entre las áreas involucradas debe ser por escrito y con toda la claridad posible con el objeto de que exista un entendimiento profundo del servicio que se solicita.

1. Es responsabilidad del área de producción aceptar sistemas a implantar que estén debidamente documentados de acuerdo a los estándares de la instalación.
2. Es responsabilidad del área de producción elaborar todos los reportes concernientes a la utilización de los recursos materiales.
3. Es responsabilidad del área de producción cumplir con los procedimientos y fechas de compromiso de la producción de los sistemas conforme a lo estipulado en la documentación de los mismos.
4. Es responsabilidad del área de operación evaluar técnica y administrativamente el sistema operativo correspondiente al equipo de cómputo utilizado.
5. Es obligación del centro de cómputo comunicar los alcances y restricciones que se observen en el sistema operativo y utilerías complementarias al mismo.
6. Es deber de todas las áreas del centro de cómputo de contar con una bitácora que muestre todas las incidencias (fallas, errores, mensajes...) que se presenten en la utilización del equipo y en la operación de sistemas.
7. Todo usuario que requiera cualquier servicio del área del centro de cómputo debe llevar la orden de trabajo correspondiente.
8. Es responsabilidad del centro de cómputo sujetarse a las instrucciones señaladas en la orden de trabajo.
9. Es obligación de las áreas del centro de cómputo mantener capacitados a los usuarios para promover el correcto uso de los procedimientos y sistemas de información.
10. Es obligación del personal de informática mantener actualizado en aspectos técnicos y de tecnología de punta para su aplicación en la operación de sistemas.
11. Todo servicio solicitado al área del centro de cómputo debe ser canalizado a través de la sección de control y enlace con los usuarios y de acuerdo a los calendarios de producción.

- **Análisis de requerimientos y definición de los servicios del área (E, RP)**

- **Volúmenes de información para la toma de decisiones (E, RP)**

- **Manuales informáticos (R, E)**

Dentro de la normatividad de los centros de cómputo, siempre debe de integrarse un **manual de normas de instalaciones físicas de equipos**, que nos permita garantizar la correcta instalación de la configuración:

El manual debe contener:

- ✓ Introducción
- ✓ Objetivo
- ✓ Descripción de los equipos
- ✓ Distribución e instalación física
- ✓ Tabla de especificaciones técnicas y documentación requerida.

Debe estar elaborado por el personal del área de soporte técnico y autorizado por el proveedor, la finalidad será de proporcionar la información técnica necesaria de los requerimientos de instalaciones físicas para los equipos a instalar así como recomendaciones.

Su objetivo es estandarizar los parámetros de las instalaciones físicas basadas en las especificaciones técnicas proporcionadas por el proveedor y para lograr instalaciones seguras y adecuadas.

También debe integrarse un **manual de normas de certificación y documentación de equipos**, que nos permita garantizar la correcta recepción del equipo adquirido, dicho manual deberá contener:

- ✓ Introducción
- ✓ Objetivo

- ✓ Políticas para el proveedor y para la instalación
- ✓ Certificación de instalaciones físicas
- ✓ Procedimientos para la entrega, instalación, aceptación del equipo y certificación de garantía.
- ✓ Formatos

Debe integrarse también un **manual de normas para las memorias de las instalaciones**, que nos permita garantizar el pleno control sobre las áreas que cuente con equipos de cómputo.

Es necesario establecer una serie de medidas que nos permitan contar con la documentación respectiva de las adecuaciones en las áreas de cómputo, la atención a fallas de hardware y software, así como actas administrativas o minutas de trabajo respecto a las condiciones de seguridad y de operación.

El manual deberá contener:

- ✓ Hoja de autorizaciones
- ✓ Introducción
- ✓ Objetivo
- ✓ Políticas para el departamento de procesamiento de datos
- ✓ Procedimientos para la integración y actualización de la memoria
- ✓ Anexos

El objetivo será contar con un documento normativo que nos permita conocer toda la información referente a la instalación, operación y mantenimiento del equipo de cómputo que se trate.

Como ejemplo de políticas se deben incluir:

- ✓ Se integrará una memoria de instalaciones por cada una de las áreas que cuenten con equipos de cómputo.
- ✓ Las modificaciones o cambios realizados en las instalaciones, se documentarán e integrarán en la memoria de instalaciones.

• Sistemas de información gerenciales (E, RP)

Qué es un SIG (en inglés MIS):

- ✓ Sistema de Información Gerencial
- ✓ Sistema de Procesamiento de Información
- ✓ Sistema de Información y de Decisiones
- ✓ Sistema de Información para la Organización
- ✓ Sistema de Información

Sistema de procesamiento de información basado en la computadora que apoya las funciones de operación, administración y toma de decisiones de una organización.

Definición:

Sistema integrado usuario-máquina para proveer información que apoye las operaciones, la administración y las funciones de toma de decisiones en una empresa. Utiliza equipo de cómputo (HW) y programas (SW), procedimientos manuales, modelos para el análisis, la planeación, el control y la toma de decisiones, además de una base de datos.

Disciplinas académicas relacionadas con los SIGs

- ✓ Contabilidad gerencial.
- ✓ Investigación de operaciones.
- ✓ Administración y organización.
- ✓ Ciencia de la computación.

Estructura del SIG

✓ Elementos operacionales

Equipo (Hardware)

Entrada/Salida

Almacenamiento

Procesamiento

Comunicaciones

Programas/Aplicaciones (Software)

De aplicación
De sistema

Base de datos

Conjunto de datos relacionados por mecanismos de referencia.

Procedimientos

Manuales de usuario
Manuales de errores
Manuales técnicos

Personal de operaciones

Operadores
Analistas
Programadores
Capturistas

✓ **Funciones de procesamiento dentro del SIG**

Proceso de transacciones:

Ejecución/realización de actividades básicas de la organización. Requiere registros para:

- ✓ Dirigir la realización de la Transacción.
- ✓ Reportar, confirmar o explicarla
- ✓ Comunicarla a quien la necesita

Mantener archivos maestros

Actualización de los catálogos/listas con información "permanente"

Producir informes

Generar reportes sobre la información/datos existentes en el sistema, pueden ser:

- ✓ preestablecidos
- ✓ a la medida "ad hoc"

Procesar preguntas

Responder a preguntas sobre la BD usando formatos

- ✓ preestablecidos
- ✓ "ad hoc"

Aplicaciones con interacción

Soporte a toma de decisiones, aplicaciones que permiten evaluar el "que pasa si? "

4. Administración de bienes informáticos

• **Hardware: uso, selección, licitación, adquisiciones, renta, licencia de uso y actualizaciones (R, E, RP)**

Los sistemas computacionales a menudo se adquieren sin estar acostumbrados a ello debido a la recomendación de alguien. También se examinan los factores financieros implicados en la adquisición y alquiler a largo plazo de sistemas y la contratación de servicio y mantenimiento. Puesto que las computadoras varían en un rango desde las microcomputadoras hasta los grandes sistemas de red, el número de opciones del cual elegir un sistema, obviamente, es muy grande. Aun dentro de las líneas de un solo fabricante, hay muchos modelos y configuraciones de los cuales se puede seleccionar. El punto de partida en un proceso de decisión acerca de un equipo son los requerimientos de tamaño y capacidad. Un sistema particular de computo puede ser apropiado para una carga de trabajo e inadecuado para otro. La capacidad de los sistemas es frecuentemente el factor determinante.

Entre las características relevantes a considerar están las siguientes:

- ✓ Tamaño de memoria interna
- ✓ Velocidad del ciclo del sistema para procesamiento
- ✓ Número de canales para entrada, salida y comunicación

- ✓ características de los componentes de despliegue y comunicación
- ✓ Tipos y números de unidades de almacenamientos auxiliares que se le pueden agregar
- ✓ Apoyo el sistema y software de utilerías que se proporciona o se encuentra disponible

Todos los sistemas tienen sus limitaciones, dependiendo de para que se han diseñado. Las limitaciones pueden ser o no un factor en una decisión de elección particular. Las necesidades de software a menudo dictan las necesidades de hardware, tales como los tamaños de memoria interna, puertos de comunicación, capacidad de disco y la posibilidad de usar cinta magnética.

Metodos de adquisición.

- ✓ **Renta:** La renta de computadoras es adecuada para usar a corto plazo un sistema, generalmente de uno a 12 meses. Cada 30 días se hace un pago por el uso del equipo. Tanto el usuario como el proveedor tienen la opción de cancelar la renta mediante un aviso anticipado, usualmente de 30 a 60 días antes de la fecha de terminación, en comparación con los otros métodos de adquisición, la renta es el más caro. Los pagos mensuales son más altos y la organización no recibe ningún beneficio fiscal o de propiedad, de no ser por la deducción de la renta mensual como gasto de la empresa.
- ✓ **Alquiler:** Un alquiler a largo plazo es un compromiso de uso de un sistema por un tiempo específico, generalmente de tres a siete años. Se determinan con anticipación los pagos y no cambian durante todo el periodo del alquiler. Según los términos del alquiler, los pagos son mensuales, trimestrales, semestrales o anuales e incluyen el costo del servicio y mantenimiento del equipo. En comparación con la renta, el alquiler es menos caro. Debido a que hay un compromiso mayor, el proveedor generalmente proporciona un mejor servicio y el usuario puede contar con la disponibilidad del sistema para su uso. El alquiler a largo plazo protege contra la obsolescencia técnica, que siempre es de interés al comprar equipo de cómputo.
- ✓ **Compra:** La adquisición de computadoras mediante la compra directa es el método más común, y se incrementa su popularidad al aumentar los costos de alquiler. Al transcurrir el tiempo, la opción de compra con frecuencia cuesta menos, especialmente a la luz de las ventajas en cuanto a impuestos que a veces se pueden obtener. Entre sus desventajas se encuentran: el riesgo de obsolescencia, compromiso permanente. Responsabilidad total en todos los problemas, requerimientos de un mayor desembolso rápido en comparación con las otras opciones.

Mantenimiento y soporte.

Un factor adicional en las decisiones sobre hardware se refiere al mantenimiento y soporte del sistema después de su instalación. Las consideraciones principales son la fuente del mantenimiento, términos y tiempos de respuesta.

Fuente del mantenimiento: Una vez que el sistema ha sido entregado e instalado, existe un periodo de garantía durante el cual el proveedor es responsable del mantenimiento. Usualmente de 90 días, aunque los términos específicos están sujetos a la negociación del contrato. Después de ese tiempo, el comprador tiene la opción de obtener el mantenimiento de varias fuentes. La fuente más común de mantenimiento para el nuevo equipo es la empresa con la cual se adquirió.

Términos: Al formular un contrato de mantenimiento, los términos del contrato son tan importantes como el costo. El contrato se puede redactar para que cubra tanto la mano de obra como las refacciones (todas las refacciones, independientemente del número necesario o su costo), la mano de obra y cierto margen para las refacciones, o bien solamente la mano de obra, añadiéndole cargos por las refacciones.

Servicio y tiempo de respuesta: El servicio de mantenimiento es útil, solo si está disponible cuando sea necesario. Dos conceptos conciernen al mantenimiento: El tiempo de respuesta y el horario de atención.

El usuario tiene el derecho de esperar un tiempo razonable de respuesta después de hacer una llamada de emergencia. A menudo las organizaciones especifican en el contrato que la respuesta a una llamada telefónica debe hacerse dentro de las dos primeras horas. Otras aseguran la respuesta el mismo día y otras más aceptan la respuesta no más allá de la mañana siguiente. Siempre que se haga un contrato de mantenimiento, debe acordarse un programa de mantenimiento preventivo.

• Software: uso, selección, licitación, adquisiciones, renta, licencia de uso y actualizaciones (R, E, RP)

La determinación de cual software comercial es el correcto para una tarea particular y el acuerdo de los términos contractuales son responsabilidad de la organización usuaria. Una de las tareas más difíciles en la elección del

software, una vez que se conocen los requerimientos del sistema, es el determinar si un cierto paquete de software cumple con los requerimientos. Después de la selección inicial, es necesaria escudriñar un poco mas para determinar lo deseable de un software particular comparado con otros candidatos. Cuando los analistas evalúan el posible software a adoptar, lo hacen comparando las características del software con los requerimientos de la aplicación desarrollados previamente. Entre las consideraciones representativas de requerimientos están las siguientes:

- ✓ ¿Qué transacciones y que datos de cada transacción se deben manejar?
- ✓ ¿Qué reportes, documentos y otras salidas debe producir el sistema?
- ✓ ¿Qué archivos y bases de datos maneja el sistema? ¿Qué archivos de transacciones son necesarios para mantenerlos?
- ✓ ¿Cuál es el volumen de los datos por almacenar?
- ✓ ¿Existen características únicas en esta aplicación que requieran de especial atención cuando se elija el software?
- ✓ ¿Qué características de hardware y comunicación requiere el software?
- ✓ ¿Cuáles son las limitaciones del software?

Con este conjunto de preguntas junto con la orden de limitación de costos, el analista es capaz de rechazar rápidamente aquellos paquetes que no cumplen los requisitos. Es necesario analizar más a los candidatos restantes para su adopción con base en su flexibilidad, capacidad y soporte del vendedor:

Flexibilidad.

La flexibilidad de un sistema de software debe incluir la capacidad de cumplir con los requerimientos cambiantes y las diferentes necesidades del usuario. El software flexible es en general más valioso que un programa que es totalmente inflexible. Sin embargo no es deseable la flexibilidad excesiva, ya que eso requiere que el analista o el usuario definan muchos detalles en el sistema que podrían incluirse en el diseño como una característica estándar.

Las áreas donde se desea flexibilidad son el almacenamiento, los reportes y sus opciones, la definición de parámetros y la captura de datos. Además, la flexibilidad del software varía de acuerdo con los tipos de hardware con los que trabajará, por ejemplo, un programa de envío por correo interactivo, diseñado de forma que el operador capture los datos por medio de un teclado, debería permitir el uso de una o dos líneas para una dirección. Las direcciones de las empresas generalmente requieren de más de una línea para el nombre de la compañía, la calle y el departamento o lugar adonde llegue el correo, además del nombre de la persona y la ciudad, estado y código postal. Por otro lado un sistema bien diseñado no necesitará que el operador proporcione dos líneas para la dirección cuando solo se necesite una. La segunda línea se puede dejar en blanco.

Previsiones de auditoría y confiabilidad.

A menudo, los usuarios tienen una tendencia a confiar en los sistemas más de lo que deberían, al extremo de que con frecuencia creen en los resultados producidos por un sistema de información basado en una computadora sin el escepticismo suficiente. Por lo tanto, la necesidad de asegurarse de incluir los controles adecuados en el sistema es un paso esencial en la selección de software. Los auditores deben tener la capacidad de validar los reportes y salidas y probar la autenticidad y precisión de los datos e información.

Entre los procedimientos de auditoria y control están los siguientes:

- ✓ Rastrear una transacción por cada paso del proceso y tener la capacidad de examinar los valores de datos intermedios producidos durante el procesamiento.
- ✓ Imprimir registros y transacciones seleccionados del sistema que cumplan ciertos criterios (tales como una cuenta altamente activa o una cuenta con un saldo alto) para validar la precisión y autenticidad tanto de las transacciones como de los resultados.
- ✓ Mantener un balance constante en el sistema cuando éste implique cuestiones financieras y reportar si el sistema está balanceado.
- ✓ Producir un diario detallado de todas las transacciones y el efecto de éstas en los saldos de las cuentas o en los registros del archivo maestro.
- ✓ Proporcionar los controles suficientes en la entrada, tales como controles y cuenta de lotes y transacciones.

La confiabilidad de un sistema quiere decir que los datos son confiables, que son precisos y creíbles. También incluye el elemento de seguridad, el que evalúa el analista determinando el método y adecuación de protección del

sistema contra el uso no autorizado. El hecho de que el sistema tenga contraseñas no es una protección suficiente del acceso.

Capacidad.

La capacidad del sistema se refiere al número de archivos que puede guardar y el número de archivos que puede conservar. Para mostrar su capacidad total, es posible que sea necesario tomar en cuenta el hardware específico en el que se va a usar el software; la capacidad también depende del lenguaje en que se escriba el software. Pero el software puede ser un sistema de base de datos, un sistema para generar software automáticamente o con la ayuda de la computadora, o un sistema de manejo de archivos.

La capacidad también se determina por medio de lo siguiente:

- ✓ El tamaño máximo de cada registro, medido en bytes.
- ✓ El tamaño máximo del archivo, medido en bytes.
- ✓ El tamaño máximo del archivo, medido en campos por registro.
- ✓ El número de archivos que pueden estar activos a la vez.
- ✓ El número de archivos que se pueden registrar en un directorio de archivo.

Soporte y mantenimiento del vendedor.

Al comprar el software comercial por medio de un proveedor, siempre es importante evaluar los servicios que se proporcionan. El software también necesita mantenimiento, y el analista debe determinar quién lo llevará a cabo y a qué costo antes de firmar el contrato de compra-venta. Además, se deben detallar los términos del mantenimiento:

- ✓ Con qué frecuencia se llevará a cabo el mantenimiento del software? ¿Se proporcionarán las nuevas versiones de manera regular? ¿Habrá un cargo por las actualizaciones?
- ✓ Si hay una cuota mensual por mantenimiento, ¿cuáles serán los servicios que cubrirá?, ¿cuáles servicios se excluirán?
- ✓ El proveedor del software, ¿proporcionará la programación especializada para adecuar aspectos específicos del software a las necesidades demostradas del usuario? ¿A qué costo? ¿Con qué prioridad, es decir, qué tan pronto después de realizar el contrato comenzará y terminará la programación?
- ✓ ¿Cuáles son los acuerdos para controlar los aumentos en las cuotas de mantenimiento?
- ✓ ¿Cómo se resolverán los desacuerdos acerca de las necesidades de mantenimiento de software?
- ✓ ¿Durante qué horas estarán disponibles los servicios de soporte del software? ¿Cuáles son las previsiones para recibir soporte de emergencia en caso contrario después de las horas hábiles?

Aunque algunos agentes de ventas de software evitan contestar estas preguntas, los proveedores confiables no; es más, pueden proporcionar la información antes de que surja la pregunta. En cualquier caso, el analista es responsable de hacer surgir las preguntas y obtener las respuestas.

El soporte del proveedor también incluye la capacitación que se proporcionan cuando se adquiere un sistema. Puede ser gratuita o mediante una cuota, dependiendo del proveedor, y en las instalaciones del usuario o en otro lugar. Si el sistema es de uso inmediato y el proveedor proporciona capacitación e instalación “en casa”, el contrato debe establecer claramente el número de horas de capacitación que se proporcionarán y el número de personas incluidas. Generalmente, también se especifica el número de días de capacitación en las instalaciones.

• Tipos de contratos (R, E)

La negociación del contrato es un proceso legal y debe involucrar a los expertos legales y financieros de una organización. Puesto que las empresas de hardware y software negocian contratos regularmente, a menudo son mejores negociadores que los analistas de sistemas o los gerentes del procesamiento de datos. Se recomienda que el abogado de la organización haga un borrador de contrato o modifique el sugerido por el vendedor.

Se pueden delinear dos tipos de contratos de software:

- ✓ Bosquejar los términos de alquiler de un programa de software. Cuando las organizaciones adquieren software, a menudo solo reciben el derecho de usarlo mediante una cuota, ya sea una vez o por cierto tiempo. Una licencia permite a la organización el uso indefinido del software. Pero el comprador no lo posee y no puede vender el paquete, regalarlo o distribuir copias.

- ✓ Bosquejar los términos de un proyecto de programación, en donde la organización contrata a un proveedor independiente para que produzca software. Los términos pueden establecer que el trabajo se hará mediante una cuota fija o por un monto fijo y una duración específica en horas o en días de calendario.

El contrato debe fijar claramente la propiedad del software de forma que no haya disputa posteriormente y si esta incluido un servicio o alquiler, debe establecer cuando se termina el contrato.

Los contratos son: “el acuerdo de dos o más voluntades para crear, transferir, modificar o extinguir derechos u obligaciones”

Los contratos informáticos: “son catálogos, en forma general, bajo las condiciones del proveedor y usuarios”.

Algunas características particulares más importantes que resaltan este tipo de contratos son:

- 1) **Proveedores:** Son aquellos encargados de dar o hacer y fundamentalmente constituidos por los constructores, distribuidores, vendedores de equipo, así como los proveedores de servicios informáticos.

Dentro de sus principales obligaciones y derechos se encuentran:

- ✓ Salvaguardar los intereses de su cliente, así como proporcionar consejo e información.
- ✓ Cumplir con los términos de entrega o prestaciones del servicio
- ✓ Garantizar sus productos y servicios.
- ✓ Realizar convenientemente el estudio de viabilidad en caso de ser solicitado.
- ✓ Actuar con honradez y buena fe frente a los intereses del usuario
- ✓ Recibir el pago por la prestación realizada.

- 2) **Usuarios:** Son aquellos quienes reciben la prestación, dar o hacer por parte de los proveedores, o bien satisfacer necesidades a través de los bienes informáticos, y están constituidos por el sector público y privado en sus diferentes niveles.

Dentro de sus derechos y obligaciones estan:

- ✓ Informarse adecuadamente respecto a las implicaciones generadas por la firma de este tipo de contratos.
- ✓ Determinar de manera precisa sus necesidades susceptibles de automatización, así como sus objetivos.
- ✓ Capacitar apropiadamente a su personal respecto al bien o servicio informático a recibir.
- ✓ Aceptar y recibir la prestación requerida, siempre que este dentro de los términos pactados.
- ✓ Respetar los lineamientos expuestos por el proveedor respecto al modo de empleo del material o los programas de cómputo.
- ✓ Pagar el precio convenido según las modalidades fijadas entre las partes.

Tipos de contratos.

- ✓ Aquellos referidos a los **bienes** (equipos, periféricos, dispositivos, etc.) y
- ✓ Aquellos referidos a los **servicios** (asistencia, formación, mantenimiento, programas, etc.)

Dichas prestaciones son pactadas bajo las condiciones de las categorías jurídicas contractuales más conocidas: compra-venta, arrendamiento, prestación de servicios; o de aquellas no tan conocidas, como es el caso: del arrendamiento con opción a compra, también conocido por el anglicismo leasing.

Algunos de los principales tipos de contrato de acuerdo con su naturaleza son:

- ✓ Contratos de material o de sistema
- ✓ Compatibilización de equipos y programas.
- ✓ Servicios y aprovisionamiento de refacciones.
- ✓ Contratos de programa-producto
- ✓ Adquisición de programas
- ✓ Licencia de uso de programas
- ✓ Desarrollo de programas
- ✓ Análisis y tratamiento de datos
- ✓ Contratos de mantenimiento
- ✓ Contratos de formación o capacitación.

Algunos de los inconvenientes son que en algunas ocasiones se tienen que pactar dichos contratos con diversos proveedores, por lo que el fenómeno de dependencia se desconecta, pudiendo provocar variantes inadecuadas dentro de la buena marcha de las diferentes actividades informáticas.

Contenido del contrato (cláusulas): “Estas darán estructura particular a aquella fuente de derecho y obligaciones derivada de un carácter general del convenio pactado de manera bilateral”.

Algunos de los elementos que resaltan son:

- ✓ **Objeto:** Es decir las modalidades de los derechos y obligaciones respecto a los bienes o servicios informáticos.
- ✓ **Duración y rescisión:** El término de vigencia del contrato, el cuál podrá verse interrumpido en caso de no cumplir con las cláusulas por alguna de las partes.
- ✓ **Precio:** Debe ser justo, verdadero y en dinero.
- ✓ **Facturación y pago:** Las cuales tendrán lugar de acuerdo con las consignas particulares establecidas por mutuo acuerdo.
- ✓ **Control, supervisión y acceso:** Los cuales recaerán en la responsabilidad de los usuarios, a fin de que las actividades informáticas se den en las condiciones más favorables, sin injerencias internas o externas inadecuadas.
- ✓ **Asistencia y formación:** Evitando de esta manera probables actitudes negligentes o impropias por falta de un conocimiento técnico pertinente por parte del usuario.

Propiedades de los programas:

- ✓ Protección material de la información: Este es el caso de aquella información propiedad del usuario que esté bajo resguardo del proveedor, a fin de desarrollar un programa.
- ✓ Secreto y confidencialidad: Esto es en cuanto a las informaciones que se provean por ambas partes con motivo de la celebración del contrato,
- ✓ Responsabilidades y garantías: Entendiendo por responsabilidad, a aquellas obligaciones inherentes a la firma de un contrato, en este caso informático.
- ✓ Disposiciones generales: Tales como no credibilidad de las organizaciones, incumplimiento del contrato en caso de nulidad, etc.
- ✓ Diversos: Son aquellas que se refieren a un concepto en especial y que las partes convienen en insertarlas para una mejor relación contractual.

Diferentes etapas contractuales: Aquellas de carácter previo a la firma del contrato así como otras tantas como son las de recepción, verificación y conformidad respecto a la prestación recibida.

Una adecuada elección, la cual dará lugar posteriormente a la firma del contrato en la que se formalizará o perfeccionará propiamente la relación contractual en la que los contratantes aceptan tácticamente las condiciones del contrato y externan su voluntad de obligarse al cumplimiento del mismo mediante el establecimiento de su firma, dando inicio a las etapas subsecuentes, como son, en el caso de equipos informáticos, la entrega o instalación del mismo (generalmente pactado en plazos de treinta a sesenta días posteriores a la firma del contrato).

Las licitaciones o contratos administrativos: Son aquellos contratos en que una de las partes en una Administración Pública y tiene un objeto directamente relacionado con la actividad administrativa y en consecuencia están sujetos a un régimen jurídico del Derecho Privado.

Una licitación se emplea cuando alguna dependencia de gobierno desea adquirir ciertos bienes, tales como equipos de cómputo y accesorios. Para ello se lleva a cabo un concurso en el cual la empresa pública publica una convocatoria dirigida a todas aquellas personas físicas y morales que estando constituidas conforme a las leyes mexicanas, se interesen en participar en dicha licitación. Entre las leyes que podemos mencionar están la ley de Adquisiciones y Obras Públicas, el Art. 134 de la Constitución Política de los Estados Unidos Mexicanos y otros artículos dependiendo del lugar en donde se realice la licitación, es decir, se toma en cuenta la constitución política de cada Estado.

De manera general el procedimiento de licitación se compone de las siguientes fases:

1. Entrega de Bases.
2. Junta de Aclaraciones
3. Recepción de propuestas
4. Apertura de propuesta técnica

5. Elaboración del dictamen técnico
6. Notificación del dictamen
7. Apertura de ofertas económicas
8. Notificación del fallo
9. Suscripción del contrato.

• **Mantenimiento de equipos (E)**

Un factor adicional en las decisiones sobre hardware se refiere al mantenimiento y soporte de soporte del sistema después de su instalación. Las consideraciones principales son la fuente del mantenimiento, términos y tiempos de respuesta. Una vez que el sistema ha sido entregado e instalado, existe un periodo de garantía durante el cual el proveedor es responsable del mantenimiento, usualmente es de 90 días. La fuente más común de mantenimiento para el nuevo equipo es la empresa con la cual se adquirió. Si se adquiere un sistema de red o una minicomputadora por medio de la fuerza de ventas del fabricante, generalmente existe también un grupo de apoyo en mantenimiento que da servicio por un precio fijo. También se puede obtener servicio por medio de compañías especializadas. Las compañías de mantenimiento por parte de terceros, con frecuencia dan servicio a comunidades pequeñas, donde a los fabricantes no es costeable mantener oficinas.

• **Virus (E, RP)**

Un virus informático, por definición, es un pequeño programa (o código) capaz de autoreproducirse. Sin embargo algunos virus también incluyen algo que los toma como peligrosos: rutinas dañinas, también conocidas como bombas.

• **Seguridad (E, RP)**

El objetivo de los centros de cómputo debe ser procesar los sistemas de información de la empresa en forma tal que se obtengan los resultados esperados de manera confiable, oportuna y con el mínimo de recursos utilizados.

Existen tres formas básicas de tratamiento de riesgos:

1. Atacarlos
2. Transferirlos o
3. Aceptarlos

Para el establecimiento de un programa integral de seguridad en los centros de cómputo, es indispensable considerar cinco áreas de oportunidad que son:

1. Seguridad física (instalaciones)
2. Seguridad lógica (Datos, información, software)
3. Seguridad contra contingencias (Casos de desastre)
4. Seguridad administrativa (Procedimientos, políticas)

En términos generales, podemos definir a la seguridad como la serie de pasos o medidas a realizar para mantener a salvo de alguna agresión, desastre o cualquier situación que implique riesgo, de los que hay que tomar conciencia.

La seguridad en el área de informática es el conjunto de medidas, procedimientos y dispositivos adoptados para la protección física del personal, equipo soporte de información y la privacidad e integridad de la misma.

Entonces la seguridad es la condición de estar seguro y eso significa estar libre, exento de riesgos, daños o de males.

El objetivo final de la seguridad y protección de la información es garantizar la integridad, disponibilidad y confidencialidad de la información, no solo se trata de instalar o implementar protecciones, sino identificar los puntos de riesgo en áreas vulnerables en las que se deberá implementar una serie de controles cronológicos de seguridad efectiva, tomando como base las causas y no los efectos que se presentan.

Diferentes tipos de seguridad en informática:

- ✓ **Seguridad del sitio:** Se refiere a la protección física contra incendios, inundaciones y cortes de energía, cerraduras en todos los puntos de acceso al área, verificación de identidad de quién pretenda entrar a las instalaciones, manejo y almacenamiento seguro de los dispositivos magnéticos, etc.
- ✓ **Seguridad de equipos:** Seguros para el encendido de equipos, claves de acceso a los sistemas, módem y equipos auxiliares con protecciones a prueba de ataques, ubicación en lugares lejos de radiaciones caloríficas, campos magnéticos, etc.
- ✓ **Seguridad del personal:** Los procedimientos de emergencia y entrenamiento para seguridad del personal deben ser considerados y actualizados, se deberá seleccionar al personal idóneo para coordinar las actividades en caso de siniestros, programar simulacros, identificar rutas para salidas de emergencia, señalamientos restrictivos y de información.
- ✓ **Seguridad de los sistemas:** Se refiere a la seguridad que debe tener los sistemas, estar debidamente validados a prueba de errores cometidos por parte de operadores y usuarios, el acceso a las bibliotecas debe ser restringido a personal autorizado, en un medio ambiente de acceso en línea, deben existir llaves de acceso, al software y al hardware

Los archivos de cómputo deben ser protegidos en contra de accidentes, destrucción y utilización por personal no autorizado, se debe prever que todos los documentos fuente y las formas se mantengan en forma privada, confidencial, vigentes y disponibles para respaldo.

- **Plan de contingencia (E, RP)**

II. Recursos humanos

1. Administración del personal informático

- **Manejo de personal: reclutamiento, selección, supervisión, motivación, estímulos, retención y egreso (R)**

La administración de personal se refiere a las políticas y a las prácticas que se requieren llevar a cabo los aspectos relativos a las personas o al personal que competen a un puesto de administración, que incluyen reclutar, seleccionar, capacitar, compensar y evaluar. Entre ellos:

- ✓ Realizar análisis de los puestos (determinar la naturaleza del trabajo de cada empleado).
- ✓ Planificar las necesidades laborales y reclutar a candidatos para esos puestos.
- ✓ Seleccionar a los candidatos para los puestos
- ✓ Orientar y capacitar a los nuevos empleados
- ✓ Administrar los sueldos y salarios (determinar como se compensara a los empleados)
- ✓ Brindar incentivos y prestaciones
- ✓ Evaluar el desempeño
- ✓ Capacitar y desarrollar

Selección de personal:

- ✓ El reclutamiento y promoción del personal basado en criterios objetivos, considerar escolaridad, experiencia, conocimientos, habilidad, riesgos.
- ✓ Es necesario que las decisiones estén fundamentadas sobre técnicas lógicamente estructuradas y siguiendo un procedimiento científico

1. **Vacantes.-** El proceso se inicia cuando se presenta una vacante, antes de proceder a cubrirla deberá estudiarse la posibilidad de redistribución del trabajo con objeto de que las tareas respectivas sean realizadas entre el personal existente y solo en caso de no ser posible, se solicitará que esta se cubra.
2. **Requisición:** El reemplazo y el puesto de nueva creación se notifican mediante una solicitud al área de selección de personal encargada de estas funciones, haciendo notar los motivos, fecha en que deberá cubrirse el puesto, tiempo por el cual se va a contratar, departamento, categoría, turno, horario sueldo, etc.
3. **Análisis y valuación de puestos:** Una vez recibida la requisición de persona, se recurrirá al análisis y valuación de puestos con objeto de determinar requerimientos que debe satisfacer el candidato para ocupar el puesto eficientemente así como su salario.
4. **Inventario de recursos humanos:** Consiste en la localización de las personas en el inventario de recursos humanos, que presten su servicio en la organización y reúnan los requisitos establecidos, esto permitirá proporcionar elementos que conozcan la organización, y de los cuales se sabe su actuación durante el tiempo

que tienen de prestar servicios, esto disminuirá el período de entrenamiento y lo más importante, contribuirá a mantener alta la moral del personal que ya trabaja en la organización, al permitir que cada vacante signifique la oportunidad de uno o varios ascensos.

5. **Fuentes de reclutamiento:** De no existir en el inventario el personal adecuado, se acudirá a la cartera de candidatos que se encuentran en espera de una oportunidad y al no localizarlo tampoco, se recurirá a fuentes de reclutamiento, entendiendo por tales, los medios de que se vale una organización para atraer personal idóneo. Existen fuentes internas: amistades, parientes, familiares de los trabajadores y fuentes externas: agencias de empleo, medios publicitarios, bolsas de trabajo.
6. **Solicitud de empleo:** Localizados los candidatos, el ambiente en que sean recibidos así como la manera en que sean tratados, contribuirá en alto grado a mejorar la impresión que se formen de la organización, se debe determinar el área donde serán recibidos los candidatos, se debe requisitar una solicitud de empleo que abarcará básicamente: datos personales, datos familiares, experiencia ocupacional, escolaridad, puesto y sueldo deseado, disponibilidad para iniciar labores, planes a corto y largo plazo, etc.
7. **Entrevista inicial o preliminar:** Esta entrevista pretende detectar grosso modo y en el mínimo tiempo posible, los aspectos más ostensibles del candidato y su relación con los requerimientos del puesto; por ejemplo, apariencia física, facilidad de expresión, habilidad para relacionarse, etc.
8. **Prueba de trabajo:** La realización de esta prueba la lleva a cabo habitualmente el futuro jefe inmediato, a fin de comprobar que el candidato tiene los conocimientos para los cuales se ha reclutado.
9. **Examen médico de admisión:** Es de gran importancia para las empresas ya que el estado de salud puede llegar a influir en elementos tales como la cantidad y calidad de la producción, el índice de ausentismo, la puntualidad.
10. **Examen psicosométrico:** Esta evaluación consiste en aplicar un test que determine la personalidad del aspirante con el fin de determinar si reúne las características que el perfil del puesto define.
11. **Estudio socioeconómico:** Se debe realizar una investigación que incluya el conocer las actividades sociofamiliar, para detectar posibles situaciones conflictivas, actitud, responsabilidad y eficacia en otras actividades desarrolladas, comprobar la honestidad y veracidad de la información proporcionada por el candidato.
12. **Decisión final:** Con la información obtenida en cada una de las diversas fases del proceso de selección se evalúan comparativamente los requerimientos del puesto con las características de los candidatos, se presenta al jefe inmediato para su consideración y decisión final.

Procedimientos para el aseguramiento del personal:

Revisar su contratación de acuerdo a estándares de la empresa, planear programas de desarrollo, motivación e incentivos

Procedimientos para la terminación de la relación laboral:

Los procedimientos deben proteger los recursos de cómputo y los archivos de datos de la empresa.

Capacitación y desarrollo de personal

- ✓ La capacitación es una actividad planeada y basada en las necesidades reales de una organización y orientada hacia un cambio de los conocimientos, habilidades y actitudes del colaborador
- ✓ Entrenamiento continuo para mantener los conocimientos técnicos, destreza y habilidades del personal.
- ✓ Se debe elaborar manuales de inducción que contengan los objetivos de la organización, del departamento y los lineamientos de seguridad y control.

Evaluación del desempeño de los empleados

El desempeño de los empleados debe ser evaluado contra los estándares establecidos

• Delitos informáticos en el área (R, E, RP)

Son actitudes ilícitas en que se tienen a las computadoras como instrumento o fin o las conductas típicas, antijurídicas y culpables en que se tienen a las computadoras como instrumento o fin. Algunas características de estas acciones son las siguientes:

- ✓ Son conductas criminógenas de cuello blanco, en tanto que solo determinad numero de personas con ciertos conocimientos pueden llegar a cometerlas.
- ✓ Son acciones ocupacionales, en cuanto que muchas veces se realizan cuando el sujeto se haya trabajando.

- ✓ Son acciones de oportunidad, en cuanto que se aprovecha una ocasión creada o altamente intensificada en el mundo de funciones y organizaciones del sistema tecnológico y económico
- ✓ Provocan serias perdidas económicas, ya que casi siempre producen “beneficios” de mas de 5 cifras a aquellos que los realizan
- ✓ Presentan grandes dificultades para su comprobación, esto por su mismo carácter técnico

Dependiendo de cual sea el caso, los delitos informáticos son: “actitudes ilícitas en que se tienen a las computadoras como un instrumento o fin”

Características que revisten este tipo de acciones:

- ✓ Son acciones ocupacionales, en cuanto que muchas veces se realizan cuando el sujeto se halla trabajando.
- ✓ Son acciones de oportunidad, en cuanto que se aprovecha una ocasión creada o altamente intensificada en el mundo de funciones y organizaciones del sistema tecnológico y económico.
- ✓ Provocan serias pérdidas económicas, pues casi siempre producen “beneficios” de más de cinco cifras a aquellos que los realizan.
- ✓ Ofrecen facilidades de tiempo y espacio, ya que en milésimas de segundo y sin una necesaria presencia física pueden llegar a cometerse.
- ✓ Son muchos los casos y pocas las denuncias, y todo ello, debido a la misma falta de contemplación por parte del derecho.
- ✓ Son sumamente sofisticados y relativamente frecuentes en el ámbito militar
- ✓ Presentan dificultades para su comprobación, esto, por su mismo carácter técnico.
- ✓ En su mayoría son imprudentes y no necesariamente intencionales
- ✓ Ofrecen facilidades para su comisión a los menores de edad.
- ✓ Tienden a proliferar cada vez más, por lo que requieren una urgente regulación.
- ✓ Por el momento siguen siendo ilícitos manifiestamente impunes ante la ley.

Clasificación de estos ilícitos de acuerdo a dos criterios:

1. **Instrumento o medio:** Se valen de las computadoras como método, medio o símbolo en la comisión del ilícito, por ejemplo:
 - ✓ Falsificación de documento vía computarizada (tarjetas de crédito, cheques, etc).
 - ✓ Variación de los activos y pasivos en la situación contable de las empresas.
 - ✓ Planeación o simulación de delitos convencionales (robo, homicidio, fraudes, etc.)
 - ✓ “Robo” de tiempo de computadora.
 - ✓ Lectura, sustracción o copiado de información confidencial.
 - ✓ Modificación de datos, tanto en la entrada como en la salida.
 - ✓ Simulación de servicios no rendidos.
 - ✓ Aprovechamiento indebido o violación de un código para penetrar a un sistema, introduciendo instrucciones inapropiadas (Esto es lo que se conoce como “caballo de Troya”)
 - ✓ Variación en cuanto al destino de pequeñas cantidades de dinero hacia una cuenta bancaria, apócrifa, método conocido como la “técnica del salami”
 - ✓ Uso no autorizado de programas de cómputo.
 - ✓ Introducción de instrucciones que provocan “interrupciones” en la lógica interna de los programas, a fin de obtener beneficios”
 - ✓ Alteración en el funcionamiento de los sistemas.
 - ✓ Obtención de información residual impresa en papel o cinta magnética luego de la ejecución de trabajo.
 - ✓ Acceso a áreas informatizadas en forma no autorizada.
 - ✓ Intervención en las líneas de comunicación de datos o teleproceso.
2. **Fin u objetivo:** Los que van dirigidos en contra de la computadora, accesorios o programas como entidad física. Algunos ejemplos son:
 - ✓ Programación de instrucciones que producen un bloqueo total al sistema.
 - ✓ Destrucción de programas por cualquier método.
 - ✓ Daño a la memoria.
 - ✓ Atentado físico contra la máquina o sus accesorios (discos, cintas, terminales, etc.)
 - ✓ Sabotaje político – terrorismo en que se destruye o surge un apoderamiento de los centros neurálgicos computarizados.
 - ✓ Secuestro de soportes magnéticos en los que figure información valiosa con fines de chantaje, pago de rescate, etc.

Para poder combatir este tipo de ilícitos es necesario un control y tener en consideración medidas preventivas, a través de diversas formas de carácter administrativo, normativo y técnico:

- ✓ Elaboración de un examen psicosométrico previo al ingreso al área de sistemas en las empresas.
- ✓ Introducción de cláusulas especiales en los contratos de trabajo con el personal informático que por el tipo de labores a realizar así lo requiera.
- ✓ Establecimiento de un código ético de carácter interno en las empresas.
- ✓ Adoptar estrictas medidas en el acceso y control de las áreas informáticas de trabajo.
- ✓ Capacitación adecuada del personal informático a efecto de evitar actitudes negligentes.
- ✓ Identificación y, en su caso, segregación del personal informático descontento.
- ✓ Rotación en el uso de claves de acceso al sistema (passwords).

En cuanto concierne al control correctivo, esté podrá darse en la medida en que se introduzcan un conjunto de disposiciones jurídicas específicas en los códigos penales sustantivos, ya que en caso de considerar este tipo de ilícitos como figuras análogas ya existentes, corre el riesgo de alterar flagrantemente al principio de legalidad de las penas.

• **Capacitación (importancia, necesidades y políticas de la actualización permanente) (R)**

Se refiere a los métodos que se usan para proporcionar a los empleados nuevos y actuales las habilidades que requieren para desempeñar su trabajo. Por consiguiente, capacitar podría significar enseñar a un operador de maquina a operar su nueva maquina. Los 5 pasos del proceso de capacitación y desarrollo son:

1. **Análisis de las necesidades.**- Identificar las habilidades específicas para el desempeño del trabajo que se necesitan para mejorar el desempeño y la productividad. Usar investigaciones para desarrollar objetivos mensurables de los conocimientos y el desempeño.
2. **Diseño de la instrucción.**- Recabar ejemplos de objetivos, métodos, medios, descripción y secuencia de contenido para la enseñanza. Asegurarse de que los materiales, se complementan, están redactados con claridad y sirven para una capacitación unificada, adaptados en forma directa a los objetivos del aprendizaje que se hayan definido.
3. **Validación.**- Introducir y validar la capacitación ante un público representativo. Basar las revisiones finales en resultados piloto para asegurar la eficacia del programa.
4. **Aplicación.**- Reforzar el éxito mediante un taller de capacitación al instructor, que se concentre en presentar conocimientos y habilidades adicionales al contenido de la capacitación.
5. **Evaluación y Seguimiento.**- Evaluar el éxito del programa de acuerdo con: Reacción, Aprendizaje, Comportamiento y Resultados.

III. Auditoría en el área de informática

1. Auditoría informática

La podemos definir como el conjunto de Procedimientos y Técnicas para Evaluar y Controlar un Sistema Informático con el fin de constatar si sus actividades son Correctas y de acuerdo a las Normativas informáticas y generales prefijadas en la Organización.

La Auditoria Informática verifica la utilización eficiente de los recursos informáticos disponibles o por disponer, en función de las estrategias de la Empresa y de los objetivos previstos. La Auditoria Informática para diferenciarla de la convencional, no aborda otro tipo de fraude que aquellos que se pueden realizar en forma directa desde, o con las computadoras

Auditoría: Es la revisión y supervisión sistemática de una actividad o grupos de actividades. El objetivo fundamental de la auditoría es el de llevar a cabo una revisión y consideración de la organización con el fin de precisar: pérdidas y deficiencias, y mejor y uso de los recursos físicos y humanos.

Tipos de auditoria.

Las auditorías se pueden clasificar en: auditorías de estados financieros, administrativas, operacionales, sociales y técnicas.

Normas, procedimientos y tecnicas de auditoria.

- ✓ **Normas de auditoría:** Son los requisitos mínimos de calidad relativos a la personalidad del auditor, al trabajo que desempeña y a la información que rinde como resultado de dicho trabajo.
- ✓ **Procedimientos de auditoría:** Son un conjunto de técnicas de investigación aplicables a un conjunto de hechos o circunstancias a la entidad sujeta a examen, mediante los cuales el auditor obtiene las bases para fundamentar su opinión.
- ✓ **Técnicas de auditoría:** Son los métodos prácticos de investigación y prueba, que el auditor utiliza para lograr obtener la información y comprobación necesaria para poder emitir su opinión profesional.
- ✓ **Control interno:** El control interno comprende la evaluación de la suficiencia de todos los métodos y procedimientos que en forma coordinada se adoptan en una entidad para salvaguardar sus activos, verificar la exactitud y confiabilidad de su información financiera, promover la eficiencia operacional y provocar adherencia a las políticas prescritas por la administración.
- ✓ **Riesgos:** Se puede definir como la probabilidad de que un suceso ocurra y provoque perdidas en la empresa. Este puede cuantificarse como el efecto de una causa (expresado en términos monetarios) multiplicado por la frecuencia probable de que el riesgo ocurra: ($R = EC \cdot Ff$) Mediante ésta representación matemática se efectuará el cálculo de la ocurrencia del mismo, donde R = Riesgo, EC= Efecto de una Causa, F= Frecuencia
- ✓ **Informe:** Un informe es una exposición verbal o por escrito presentada por una persona a otra, particularmente respecto a alguna cuestión investigada.

Un informe de auditoría es un documento mediante el cual, el auditor da a conocer el resultado obtenido en sus investigaciones; así como sus impresiones u opiniones. El informe debe ser claro, objetivo, breve y oportuno. Se debe presentar inmediatamente después de haber finalizado la auditoría. En caso de no presentarse el informe en un plazo breve y razonable, puede este caer en lo obsoleto, o bien resultar hasta inútil, motivado por la dinámica del desarrollo y operación de los sistemas.

- **Riesgos de ambientes de PC aisladas (E)**

- **Auditoría interna y externa (R)**

Auditoría interna.- Es la realizada por auditores que pertenecen a la misma empresa cuyos procedimientos están siendo revisados, este examen se conoce como auditoría interna y a quienes la realizan se les da el nombre de auditores internos.

Las responsabilidades de los auditores internos varían de una empresa a otra, pues su función es realizar evaluaciones operativas cuyos hallazgos y recomendaciones se reportan a la gerencia sin que trascienda de la institución.

Cabe acotar, que solamente las grandes empresas pueden poseer auditores internos, el resto de las organizaciones, acude a *Auditores Externos de Sistemas*, pero en general lo hacen cuando la situación es grave o está totalmente fuera de control. El elemento para analizar es la independencia del auditor interno y de la confianza que genere, limitando su radio de acción a describir hechos y situaciones débiles de la informática y de su entorno inmediato, sin opiniones carentes de prueba, estos son los pilares de su existencia, sino, habrá fracasado como Auditor Interno, aunque no la Auditoria Informática, cuya función final y de utilidad debe ser firme y sólida.

Un buen auditor es aquel que terminada su función en esta tarea, puede desarrollarse en otras dentro del mismo departamento auditado (Sistemas), conviviendo con sus anteriores compañeros de trabajo.

Objetivo: El objetivo principal y prioritario es: el mantenimiento de la Operatividad de las Computadoras. La Operatividad es una función de mínimos consistente en que la empresa y todo el hardware funcionen, aunque sea minimamente. No está previsto detener la informática para descubrir fallas y empezar de nuevo. Tal objetivo debe conseguirse tanto a nivel general como en algún área específica. La Operatividad de los Sistemas ha de constituir entonces la principal función del auditor informático.

Para contribuir a que la organización tenga éxito en el rápido cambio orientado a la tecnología, el auditor interno debe hacer lo siguiente:

- ✓ Centrarse en la efectividad del sistema para lo cual debe ser capaz de articular los costos y beneficios en el manejo de los controles. Los controles no deben estar aislados ni imponerse arbitrariamente en las funciones d los usuarios y los Sistemas de Información. Los controles deben estar incluidos en el diseño del sistema.

- ✓ Desarrollar habilidades técnicas, conocimientos relevantes y tener una continua educación sobre la tecnología de información, de tal manera que el auditor sea capaz de responder a las expectativas de la organización en el manejo e implementación de los controles.
- ✓ Adquirir conocimientos sobre los recursos de computación actual. El auditor debe reconocer que los recursos de computación están cambiando y que los sistemas de información pueden residir fuera de las supercomputadoras (mainframe). Los sistemas distribuidos o físicamente dispersos requieren nuevas técnicas de control y enfoques de auditoría.
- ✓ Llegar a ser un consultor o servicio de orientación que pueda auxiliar a los administradores en la identificación de los riesgos, implementación y evaluación de controles en el procesamiento electrónico de datos.

El objetivo de la auditoría interna es auxiliar a los miembros de la empresa en el desempeño efectivo de sus responsabilidades. Para este fin, el auditor interno presenta un informe en el cual proporciona un análisis, estimación, recomendaciones, consejos e información concerniente a las actividades revisadas.

Los miembros a los que auxiliar el auditor interno en una empresa incluye a los administradores y al grupo de Directores. El auditor tiene la responsabilidad de proporcionarles información sobre lo adecuado y efectivo que es el sistema de control interno así como de la calidad de la ejecución del mismo.

Estructura organizacional de auditoría interna:

- ✓ El área de auditoría regularmente se encuentra a nivel staff en las organizaciones, depende directamente de la alta gerencia.
- ✓ Este grupo realizará supervisiones independientes y reporta sus hallazgos y recomendaciones a la alta gerencia

Responsabilidad de la función de auditoría interna:

- ✓ Las funciones deben estar claramente definidas y diferenciadas con respecto a las funciones de control de calidad.
- ✓ Debe establecerse por escrito el alcance, el programa de trabajo y las recomendaciones de las revisiones que se realicen.
- ✓ Debe establecerse un procedimiento de seguimiento para la atención a las recomendaciones que se presenten.

Función de auditoría interna en los sistemas:

- ✓ Participación activa en el proceso de desarrollo de sistemas, para garantizar la incorporación de medidas adecuadas de seguridad y puntos de verificación.
- ✓ Revisión de los sistemas ya implantados tanto en las áreas usuarias como en el centro de cómputo en cuanto a controles de la aplicación y oportunidad de procesamiento.
- ✓ Revisión de las políticas y procedimientos de seguridad para con los datos, archivos y bibliotecas en cuanto a almacenamiento, respaldos y recuperaciones.

Función de auditoría interna en las instalaciones:

- ✓ Evaluación de las instalaciones eléctricas y de acondicionamiento.
- ✓ Revisión de los mantenimientos preventivos y correctivos de los equipos de cómputo y auxiliares.
- ✓ Revisión de las políticas y procedimientos de seguridad para con la gente, los equipos y las instalaciones.
- ✓ Revisar las pruebas y simulacros a los procedimientos de seguridad.

Función de auditoría interna en la administración.

- ✓ Evaluar la estructura orgánica y la descripción de los puestos con respecto a la realidad.
- ✓ Revisar los programas de trabajo y la aplicación del presupuesto de operación asignado.
- ✓ Evaluar los procedimientos establecidos en cuanto al enlace con los usuarios para los sistemas en producción, asesoría y atención a fallas de software y hardware, solicitud de mantenimiento a sistemas implantados, desarrollo de nuevas aplicaciones.
- ✓ Evaluar las estrategias de administración de los recursos de personal, técnicos y materiales.

Auditoría externa.- Las responsabilidades de los auditores externos se refieren de manera fundamental a estatutos legales, sus responsabilidades están claramente definidas por la ley. Su función principal es revisar las funciones de un centro de cómputo y expresar su opinión acerca de la actuación que tienen cada una de las áreas con respecto a la normatividad gubernamental.

Es la realizada por auditores independientes a los cuales se les da el nombre de Auditores Externos. Los administradores están autorizados a permitir que dichos auditores revisen el trabajo realizado por los auditores internos y ver si cumplen con los estándares permitidos por la profesión, debido a que en algún momento necesitan presentar informes sobre el estado de la organización a usuarios externos.

Los usuarios externos de la información necesitan tener la seguridad de que los informes se preparan sin prejuicios y cumpliendo con los principios de la profesión generalmente aceptados. Ellos tendrán poca satisfacción y poca seguridad sobre los informes que les dé un auditor interno debido a que es un empleado de la misma empresa y no confiarán en la objetividad de los informes.

Aunque los auditores internos deben ser independientes de los demás empleados dentro de la organización cuyo trabajo revisan, también son empleados de la organización. Motivo por el cual los usuarios externos exigen que la información sea revisada por el auditor independiente.

La Administración confía en el Auditor independiente para que le notifique las debilidades que descubre durante su revisión en el diseño o funcionamiento del sistema. El auditor independiente puede aconsejar a la Administración sobre el diseño inicial del sistema de control interno, identificar debilidades y riesgos importantes en el sistema y ofrecer recomendaciones para revisiones del sistema. Pera la responsabilidad de un buen Control Interno Corresponde a la Administración.

C. Normatividad jurídica

I. Marcos legales

1. Consideraciones legales

- Constitución Política de los Estados Unidos Mexicanos (R)
- Ley Federal del Trabajo (R)
- Ley de Adquisiciones y Obra Pública (R)
- Ley de Comunicaciones y Transportes (R)
- Ley Federal de Derechos de Autor (R)

Artículo 2o.- Para los efectos de este Reglamento se entenderá por:

- I. Ley: La Ley Federal del Derecho de Autor;
- II. Secretaría: La Secretaría de Educación Pública;
- III. Secretario: El Secretario de Educación Pública;
- IV. Instituto: El Instituto Nacional del Derecho de Autor;
- V. Director General: El Director General del Instituto Nacional del Derecho de Autor;
- VI. Registro: El Registro Público del Derecho de Autor;
- VII. Diario Oficial: El Diario Oficial de la Federación;
- VIII. Código Penal: El Código Penal para el Distrito Federal en Materia de Fuero Común y para toda la República en Materia de Fuero Federal;
- IX. Código Civil: El Código Civil para el Distrito Federal en Materia Común y para toda la República en Materia Federal;
- X. Reserva: La Reserva de Derechos al Uso Exclusivo;
- XI. Sociedades: Las Sociedades de Gestión Colectiva.

Artículo 36.- Se entiende por espectro electromagnético lo establecido en la fracción II del artículo 3o. de la Ley Federal de Telecomunicaciones.

Artículo 37.- Son medios electrónicos aquellos que hagan posible el acceso remoto al público de obras literarias y artísticas por medio del espectro radioeléctrico o las redes de telecomunicación.

Artículo 101. Se entiende por programa de computación la expresión original en cualquier forma, lenguaje o código de un conjunto de instrucciones que, con una secuencia, estructura y organización determinada, tiene como propósito que una computadora o dispositivo realice una tarea o función específica.

Artículo 102. Los programas de computación se protegen en los mismos términos que las obras literarias. Dicha protección se extiende tanto a los programas operativos como a los programas aplicativos, ya sea en forma de código fuente o de código objeto. Se exceptúan aquellos programas de cómputo que tengan por objeto causar efectos nocivos a otros programas o equipos.

Artículo 103. Salvo pacto contrario, los derechos patrimoniales sobre un programa de computación y su documentación, cuando hayan sido creados por uno o varios empleados en el ejercicio de sus funciones o siguiendo las instrucciones del empleador, corresponden a éste.

Artículo 104. Como excepción a lo previsto en el artículo 27 fracción IV, el titular de los derechos de autor sobre un programa de computación o sobre una base de datos conservará, aún después de la venta de ejemplares de los mismos, el derecho de autorizar o prohibir el arrendamiento de dichos ejemplares. Éste precepto no se aplicará cuando el ejemplar del programa de computación no constituya en sí mismo un objeto de la licencia de uso.

Artículo 105. El usuario legítimo de un programa de computación podrá realizar el número de copias que le autorice la licencia concedida por el titular de los derechos de autor, o una sola copia de dicho programa siempre y cuando:

- ✓ Sea indispensable para la utilización del programa, o
- ✓ La copia de respaldo deberá ser destruida cuando cese el derecho del usuario para utilizar el programa de computación.

Artículo 106. El derecho patrimonial sobre un programa de computación comprende la facultad de autorizar o prohibir:

- ✓ La reproducción permanente o provisional del programa en todo o en parte, por cualquier medio o forma;
- ✓ La traducción, la adaptación, el arreglo o cualquier otra modificación de un programa y la reproducción del programa resultante.
- ✓ Cualquier forma de distribución del programa o de una copia del mismo, concluido el alquiler.

Artículo 107. Las bases de datos o de otros materiales legibles por medio de máquinas o en otra forma, que por razones de selección y disposición de su contenido constituyan creaciones intelectuales, quedarán protegidas como compilaciones. Dicha protección no se extenderá a los datos y materiales en sí mismos.

Artículo 108. Las bases de datos que no sean originales quedan, sin embargo, protegidas en su uso exclusivo por quien las haya elaborado, durante un lapso de cinco años.

Artículo 109. El acceso a información de carácter privado relativa a las personas contenida en las bases de datos a que se refiere el artículo anterior, así como, la publicación, reproducción, divulgación, transmisión pública y transmisión de dicha información, requerirá la autorización previa de las personas de que se trate.

Artículo 110. El titular del derecho patrimonial sobre una base de datos tendrá el derecho exclusivo, respecto de la forma de expresión de la estructura de dicha base, de autorizar o prohibir:

- ✓ Su reproducción permanente o temporal, total o parcial, por cualquier medio y de cualquier forma.
- ✓ La distribución del original o copias de la base de datos.
- ✓ La comunicación al público, y
- ✓ La reproducción, distribución o comunicación pública de los resultados de las operaciones mencionadas en la fracción II del presente artículo.

Artículo 111. Los efectuados electrónicamente que contengan elementos visuales, sonoros, tridimensionales o animados quedan protegidos por esta Ley en los elementos primigenios que contengan.

Artículo 112. Queda prohibida la importación, fabricación, distribución y utilización de aparatos o la prestación de servicios destinados a eliminar la protección técnica de los programas de cómputo, de las transmisiones a través del espectro electromagnético y de redes de telecomunicaciones y de los programas de los elementos electrónicos señalados en el artículo anterior.

Artículo 113. Las obras e interpretaciones o ejecuciones transmitidas por medios electrónicos a través del espectro electromagnético y de redes de telecomunicaciones y el resultado que se obtenga de ésta transmisión estarán protegidas por esta Ley.

Artículo 114. La transmisión de obras protegidas por ésta Ley mediante cable, ondas radioeléctricas, satélite u otras similares, deberán adecuarse, en lo conducente, a la legislación mexicana y respetar en todo caso y en todo tiempo las disposiciones sobre la materia.

REGISTRO DEL SOFTWARE

1. Formato respectivo consignando todos los datos exigidos.
2. Manual del usuario.
3. La Parte Esencial o Característica del Código Fuente.
4. Indistintamente la Memoria Descriptiva o un Ejemplar del Soporte donde esté contenida la Obra (los ejecutables).
5. Copia de los documentos que acrediten la existencia de la Persona Jurídica.
6. Los Poderes que fueren necesarios.
7. En el caso de Documentos elaborados en el extranjero, deberán estar traducidos al español.

2. Política y legislación informática

Política informática.

Conjunto de medidas y acciones que a través de instrumentos disponibles influyen en el desarrollo y modelo de la aplicación de la tecnología informática, según condiciones de tiempo y espacio bien concretas de un complejo socioeconómico dado.

Legislación informática.

Conjunto de reglas de carácter preventivo y correctivo derivadas del uso de la informática.

Aspectos:

- ✓ regulación de los bienes informacionales
- ✓ protección de datos personales
- ✓ propiedad intelectual e informática
- ✓ delitos informáticos
- ✓ contratos informáticos
- ✓ comercio electrónico
- ✓ aspectos laborales de la informática
- ✓ valor probatorio de los soportes modernos de información

• Normas regulatorias y políticas de la actividad informática: nacionales e internacionales (R)

- ✓ Disposiciones tipo para la protección del soporte lógico (Organización Mundial para la Propiedad Intelectual OMPI).
- ✓ Proyecto de tratado para la protección de los programas de cómputo (OMPI).
- ✓ Ley del software (Brasil).
- ✓ Reglamento de ley brasileña del software (Brasil).
- ✓ Acuerdo no. 114, por el que se dispone que los programas de computación podrán inscribirse en el registro público del derecho de autor (México).
- ✓ Ley de Servicios de la Sociedad de la Informática y de Comercio Electrónico.
- ✓ Ley Federal de Transparencia y Acceso al Información Pública Gubernamental (capítulo IV)
- ✓ Ley Federal del Derecho de Autor (Protección de los programas de computación).
- ✓ Decreto por el que se reforman y adicionan diversas disposiciones del Código Civil para el Distrito Federal en materia común y para toda la República en Materia Federal, del Código Federal de procedimientos civiles, del Código de Comercio y de la Ley Federal de Protección al Consumidor (regulación del comercio electrónico).
- ✓ Política de solución de controversias en materia de nombres de dominio para .mx
- ✓ Decreto por el que se reforman y adicionan diversas disposiciones del código de comercio materia de firma electrónica.
- ✓ Legislación Mexicana sobre Delitos Informáticos y en materia autoral.

• Legislación específica (R)

Derecho informático.

Rama de la ciencia jurídica que considera a la informática como instrumento (informática jurídica) y objeto de estudio del derecho (derecho de la informática).

Clasificación:

- ✓ **Informática jurídica:** Técnica interdisciplinaria que estudia e investiga la información general aplicable a la recuperación de análisis y tratamientos jurídicos.

Clasificación:

1. documentaria
2. control y gestión
3. mercadotecnia

- ✓ **Derecho de informática:** conjunto de leyes, normas y principios aplicables a hechos y actos derivados de informática.

Clasificación:

1. política informática
2. legislación informática

• Delitos informáticos (R)

Delito por computador: cualquier violación a la ley realizada por medio del conocimiento o uso de la tecnología de computación.

Delito o crimen informático es toda interrupción, uso indebido, modificación, o fabricación de datos ajenos que se encuentren en sistemas de computación, sin autorización expresa o implícita de su dueño y/o de quien ostente la propiedad intelectual, con el objeto de obtener un provecho económico o no.

Clases de delitos Informáticos:

✓ **Ataques Pasivos**

1. Divulgación del contenido de mensajes ajenos
2. Análisis del tráfico de información de terceros

✓ **Ataques Activos**

1. Utilización de passwords ajenos
2. Modificación o alteración de mensajes y/o archivos
3. Obstaculización de accesos legítimos

Características:

- ✓ conductas delictivas de cuello blanco
- ✓ acciones ocupacionales
- ✓ enormes pérdidas económicas
- ✓ facilidades de tiempo y espacio
- ✓ no se denuncian
- ✓ frecuentes en el ámbito militar
- ✓ grandes dificultades para su comprobación
- ✓ dolosos
- ✓ facilidades de cometer por menores de edad

Clasificación:

✓ **Como instrumento o medio:**

1. falsificación de documentos vía computarizada
2. variación de los activos y pasivos en la situación contable de las empresas
3. planeación o simulación de delitos convencionales
4. robo de tiempo de computadora
5. lectura, sustracción o copiado de información confidencial
6. modificación de datos en la entrada o salida
7. aprovechamiento indebido o violación de un código para penetrar a un sistema con el fin de introducir instrucciones inapropiadas
8. variación en cuanto al destino de pequeñas cantidades de dinero hacia una cuenta bancaria apócrifa
9. uso no autorizado de programas de cómputo
10. insertar instrucciones que provocan interrupciones en la lógica interna de los programas, a fin de obtener beneficios
11. alteración en el funcionamiento los externos
12. obtención de información residual impresa en papel de o en cinta magnética luego de la ejecución de trabajos
13. acceso a áreas informatizadas en forma no autorizada
14. intervención de las líneas de comunicación de datos o teleproceso

✓ **Como fin u objetivo.**

1. programación de instrucciones que producen un bloqueo total al sistema
2. destrucción de programas por cualquier método
3. daño a la memoria
4. atentado físico contra la máquina o sus accesorios
5. sabotaje político o terrorismo en el que se destruya o surja un apoderamiento de los centros neurálgicos computarizados
6. secuestro de soportes magnéticos en los que figure información valiosa con fines de chantaje, que, pago de rescate, etcétera

Robo: forma más común de delito. Se roba dinero, bienes, información y recursos de computación.

Piratería de software: duplicación ilegal de software protegido por derechos de autor.

Sabotaje de software:

1. Caballos de Troya: programa que ejecuta una tarea útil al mismo tiempo que realiza acciones destructivas secretas
2. virus: se propaga de programa en programa, o de disco en disco, y en cada programa o disco infectado crea más copias de ellos. por lo general, esta oculto en un programa de aplicación o en el sistema operativo de un computador
3. gusanos: usan computadores como anfitriones para reproducirse. Viajan de manera independiente por las redes, en busca de estaciones de trabajo no infectadas. Reside en la memoria, no en disco.

Hackers: ingresan ilegalmente en otros computadores usando contraseñas robadas o deficiencias de seguridad en el software del sistema operativo. Se enlazan vía modem, internet y otras redes.

Técnica de salami: Es la desviación del destino de pequeñas cantidades de dinero hacia una cuenta bancaria apócrifa.

• **Diagnóstico, planes y programas nacionales de desarrollo (R)**

La importancia de contar con una estrategia que permita aprovechar el potencial que representan las tecnologías de la información en el marco de los objetivos nacionales, motivó al INEGI, como institución responsable de la formulación de la política nacional en informática, a emprender en coordinación con otras dependencias de la Administración Pública y con distintos grupos sociales, una revisión profunda de la situación nacional que permitiera la planeación de las acciones requeridas para garantizar un desarrollo sostenido y armónico de la informática.

Con el ámbito de llevar a cabo esta recomendación se realizaron algunos documentos para fundamentar antes las autoridades competentes, la importancia de incorporar la informática en el Plan Nacional de Desarrollo 1995-2000. Asimismo, se incluye dentro de los 32 programas sectoriales, regionales, institucionales y especiales, el Programa de Desarrollo Informático.

Para la formulación de este Programa, el INEGI mantuvo el proceso de consulta con diversos sectores a través de la integración de seis grupos de trabajo encargados de revisar las propuestas básicas y proponer el contenido fundamental de las acciones a realizar en materia de recursos humanos, investigación y desarrollo, mercado, usos de la informática en el sector privado, estrategias tecnológicas para el sector público, telecomunicaciones y marco normativo e institucional. Como resultado de estos trabajos, se formuló la estrategia general del Programa de Desarrollo Informático, que define los lineamientos a seguir por el Gobierno Federal para la promoción del uso y desarrollo informático.

Los Proyectos Informáticos Nacionales son un instrumento para coordinar esfuerzos y recursos en la solución de una problemática nacional, en el cual las tecnologías de la información puedan tener una repercusión trascendente.

Los Proyectos Informáticos Regionales son un mecanismo instituido en este Programa, con la finalidad de facilitar esfuerzos de colaboración para llevar a cabo proyectos que responden a una problemática en el ámbito local, pero tienen repercusiones internacionales.

El Plan Nacional de Desarrollo propone cinco objetivos fundamentales para el desarrollo integral del país:

- ✓ **fortalecer el ejercicio pleno de nuestra soberanía,**
- ✓ **consolidar un país de leyes y justicia:** la informática puede contribuir a mejorar la seguridad pública y la procuración e impartición de justicia.
- ✓ **alcanzar un pleno desarrollo democrático:** al acrecentar las posibilidades de acceso a la información, la tecnología permite una mayor participación social en todas las actividades de la vida nacional.
- ✓ **impulsar un desarrollo social con oportunidades de superación para todos:** la informática puede contribuir a la justicia social y la equidad, mejorando la cobertura y la calidad de diversos servicios como educación, salud y seguridad social.
- ✓ **conseguir un crecimiento económico vigoroso, sostenido y sustentable:** la informática posibilita el uso racional y eficiente de los recursos humanos y materiales de que dispone el país, contribuyendo a la asignación del potencial productivo de México a favor de la inversión y del crecimiento económico.

- **Organismos de la Administración Pública Federal responsables de la definición, implantación y regulación de políticas (R)**

Instituto Nacional de Estadística, Geográfica e Informática.

En México, la formulación de la política nacional en materia informática ha estado bajo la responsabilidad del Instituto Nacional de Estadística, Geografía e Informática y las instituciones que lo antecedieron. Algunas atribuciones y responsabilidades complementarias quedan a cargo de otras entidades y dependencias; sin embargo, es importante destacar el papel del INEGI como autoridad nacional en la materia.

Diversos ordenamientos jurídico-administrativos hacen explícitas para el INEGI sus responsabilidades y atribuciones específicas frente a la comunidad informática nacional. En esencia, se le encomiendan dos funciones: fomentar el uso de la informática, concretamente en la Administración Pública Federal, y el desarrollo informático nacional.

Instituto de la Administración Pública Federal con atribuciones vinculadas con la Informática.

En la Administración Pública Federal existen diversas instituciones con atribuciones que directa o indirectamente inciden en el ámbito de la informática, cuya participación es necesaria para promover el desarrollo nacional en la materia.

✓ **Secretaría de Relaciones Exteriores.**

Promover, propiciar y asegurar la coordinación de acciones en el exterior de las dependencias y entidades de la Administración Pública Federal; y sin afectar el ejercicio de las atribuciones que a cada una de ellas corresponda, conducir la política exterior para lo cual intervendrá en toda clase de tratados, acuerdos y convenciones en los que el país sea parte.

✓ **Secretaría de Hacienda y Crédito Público.**

Coordinar y desarrollar los servicios nacionales de estadísticas y de información geográfica, establecer las normas y procedimientos para la organización, funcionamiento y coordinación de los sistemas nacionales estadísticos de información geográfica, así como normar y coordinar los servicios de informática de las dependencias y entidades de la Administración Pública Federal.

✓ **Secretaría de Comercio y Fomento Industrial.**

Normar y registrar la propiedad industrial y mercantil, así como regular y orientar la inversión extranjera y la transferencia de tecnología.

✓ **Secretaría de Comunicaciones y Transportes.**

Otorgar concesiones y permisos previa opinión de la Secretaría de Gobernación para establecer y explotar sistemas de servicios telegráficos, telefónicos, sistemas y servicios de comunicación inalámbrica por telecomunicaciones y satélites, de servicio público de procesamiento remoto de datos, estaciones de radio experimentales, culturales y de aficionados y estaciones de radiodifusión comercial y cultural; así como vigilar el aspecto técnico del funcionamiento de tales sistemas, servicios y estaciones.

✓ **Secretaría de Contraloría y Desarrollo Administrativo.**

Inspeccionar y vigilar, directamente o a través de órganos de control, que las dependencias y entidades de la Administración Pública Federal cumplan con las normas y disposiciones en materia de sistemas de registro y contabilidad, contratación y remuneraciones de personal, contratación de adquisiciones, arrendamientos, servicios y ejecución de obra pública, conservación, uso, destino, afectación, enajenación, y baja de bienes muebles e inmuebles, almacenes y demás activos y recursos materiales de la Administración Pública Federal.

✓ **Secretaría de Educación Pública.**

Promover la creación de institutos de investigación científica y técnica, y el establecimiento de laboratorios, observatorios, planetarios y demás centros que requiera el desarrollo de la educación primaria, secundaria, normal, técnica y superior; orientar, en coordinación con las dependencias competentes del Gobierno Federal y con las entidades públicas y privadas el desarrollo de la investigación científica y tecnológica.

- **Organismos no gubernamentales que apoyan el desarrollo informático (R)**

- **Tecnología informática: transferencia, adaptación, investigación, adecuación, autonomía e independencia (E, RP)**

La información es inherente a la existencia de las personas y de las sociedades. El aprovechamiento de la información propicia la mejoría de los niveles de bienestar y permite aumentar la productividad y competitividad de las naciones.

El importante aporte de la información se ha visto acrecentado por la posibilidad que ha traído consigo la informática, surgida de la convergencia tecnológica de la computación, la microelectrónica y las telecomunicaciones, para producir información en grandes volúmenes, y para consultarla y transmitirla a través de enormes distancias.

Las consecuencias de esta revolución tecnológica serán múltiples y algunas ya son claramente perceptibles. En el ámbito económico, en particular, los avances tecnológicos han permitido reducir, en formas antes inimaginables, el tiempo requerido para producir bienes de toda índole. Así con el apoyo de la informática se han alcanzado niveles muy superiores de productividad y competitividad.

Los servicios que exigen un manejo masivo de información, como los del sector financiero, los seguros y el comercio, pueden prestarse en forma casi instantánea, aumentando su eficiencia, al poder enlazarse oficinas, clientes y proveedores en cualquier parte del mundo a través de redes de computadoras, también, la informática ha hecho posible un mercado mundial capaz de reaccionar prácticamente al instante a los eventos que se suscitan en cualquier parte de nuestro planeta y que permite amplias transacciones de productos y servicios.

La informática está modificando también a las organizaciones: Se eficientizan las estructuras, se redefinen las responsabilidades de los directivos y de los trabajadores. Aparecen nuevos enfoques administrativos que buscan mejorar la productividad y la competitividad, como es la administración mediante la calidad total y la reingeniería, que para su exitosa aplicación se apoyan de manera fundamental en la tecnología informática.

En los ámbitos sociales y culturales, los efectos de la informática son tan importantes como los económicos, porque están cambiando las formas tradicionales de organización y comunicación, transformando las actividades y las condiciones de vida. El trabajo a distancia es ya una realidad en algunos países.

Con el apoyo de la informática, los gobiernos, las instituciones educativas y los organismos asistenciales están en posibilidades de mejorar sustancialmente los mecanismos tradicionales de gestión y de servicio, lo cual se traduce en beneficios reales y tangibles para la población.

Con el uso de esta tecnología, los servicios públicos pueden proporcionarse de forma radicalmente distinta, en lo que toca a su cantidad, y en lo referente a su calidad. Las computadoras y las telecomunicaciones pueden coadyuvar el suministro de estos servicios a comunidades marginadas.

- **Confidencialidad y privacidad de la información (R)**

Protección jurídica de los datos personales.

Las computadoras, al permitir un manejo rápido y eficiente de grandes volúmenes de información, facilitan la concentración automática de datos referidos a las personas, constituyendo así en un verdadero factor de poder. Este tipo de datos son vulnerables según la aplicación de que puedan ser objeto, la cual puede ser variada; de esta forma, dichas informaciones pueden ser empleadas para fines publicitarios, comerciales, fiscales, policíacos, etcétera, convirtiéndose de esta manera en un instrumento de opresión y mercantilismo. La variedad de los supuestos posibles de indefensión frente problema provoca que los individuos estén a merced de un sinnúmero de situaciones que alteren sus derechos fundamentales en sociedad provocados por discriminaciones, manipulaciones, persecuciones, presiones y asedios; todo ello al margen de un control jurídico adecuado..

Son variadas las figuras jurídicas bajo las cuales se ha estudiado y intentado regular dicha cuestión. Figuras como los derechos humanos, derechos personales, derechos patrimoniales, libertades públicas y privadas en el caso de Francia, derecho de la privacidad en los países anglosajones, derecho a la intimidad y al honor de las personas en España, y las garantías individuales y sociales en México, han tendido hacia una sujeción apropiada en cuanto a la concentración y destinación de los datos de carácter personal.

Derechos de la regulación jurídica:

- ✓ derecho de acceso
- ✓ derecho de rectificación
- ✓ derecho de uso conforme al fin
- ✓ derecho para la prohibición de interconexión de archivos

• El "poder informático" y el poder de la información: usos y abusos (R, E)

A lo largo de la historia, las sociedades humanas han tenido 'especialistas en información' y 'tecnologías de la información'; sin embargo, hay dos tendencias relacionadas, una social y otra tecnológica, que apoyan el diagnóstico de que en la actualidad se está produciendo una revolución de la información.

En primer lugar, están los cambios sociales y de organización. El procesado de información se ha vuelto cada vez más visible e importante en la vida económica, social y política. Una prueba es el crecimiento estadístico de las ocupaciones especializadas en actividades de la información. Estas ocupaciones suponen hoy la mayor cuota del empleo en muchas sociedades industrializadas. La categoría más extensa es la de los procesadores de información, distribuidores y trabajadores administrativos, enseguida por la de productores de información, distribuidores y trabajadores de infraestructura.

En segundo lugar, está el cambio tecnológico. Las nuevas tecnologías de la información (IT) basadas en la microelectrónica, junto con otras innovaciones, con los discos ópticos o la fibra óptica, permiten enormes aumentos de potencia y reducciones de costo en toda clase de actividades de procesado de información (el término 'procesado de información' cubre la generación, almacenamiento, transmisión, manipulación y visualización de información, que incluye datos numéricos, de texto, de sonido o de vídeo).

Los actuales dispositivos informáticos y de telecomunicaciones manejan datos en forma digital empleando las mismas técnicas básicas. Estos datos pueden ser compartidos por muchos dispositivos y medios, procesarse en todos ellos y emplearse en una amplia gama de actividades de procesado de información.

Usos.

En la actualidad son múltiples los usos proporcionados por el poder informático:

- ✓ En las oficinas y el surgimiento de la Ofimática.
- ✓ En la Salud, con una mejor preparación de historias clínicas, exámenes y diagnósticos más complejos, mayor exactitud en las pruebas de laboratorio, mejor control en los productos farmacéuticos.
- ✓ En la construcción.
- ✓ Astronomía.
- ✓ En la Comunicación.
- ✓ Diversión y entretenimiento.

Abusos.

Se pueden mencionar los Delitos Informáticos así como los Delitos convencionales.

✓ **Delitos Informáticos:**

1. Piratería del Software: Este problema afecta la industria de cómputo, consiste en el copiado o uso ilegal de programas.
2. Destrucción de Datos: Los daños causados en la red mediante la introducción de virus, bombas lógicas y demás actos de sabotaje informático.
3. Robo de Hardware: Las computadoras notebook y laptop son ahora los objetos de hardware más robados. También son robadas las microcomputadoras de las compañías así como dispositivos periféricos como impresoras y módems.
4. Robo de información: Hay tres maneras en que los datos pueden ser robados. Primera, alguien puede llevarse el medio en que son almacenados los datos. Segunda, alguien puede robarse la computadora y su disco duro. Tercera, alguien puede entrar ilegalmente a los sistemas de cómputo de una organización y obtener acceso a archivos importantes.
5. Hackers.

II. La era de la información

1. Autoría informática

• La autoría y creación del software (R)

Protección de programas por el derecho de marcas.

Facilidad para el realizador en la comercialización de su producto, pues un imitador podría poner en circulación un nuevo programa inspirado del primero o aún plagiarlo bajo un nombre diferente.

Algunos autores favorecen el uso de esta vía para los programas de aplicación en forma especial por ser más fácilmente comercializables e identificables. Sin embargo, el alcance del derecho marcario es evidentemente insuficiente y limitado puesto que no comprende más que los signos exteriores que aseguran la individualización del producto.

La marca registrada con la cual sería reconocido el programa no protegería más que la denominación y no tanto al producto mismo.

La protección de programas por derecho de patentes:

Tres características que deben presentar todo objeto patentable:

- ✓ novedad
- ✓ actividad inventiva
- ✓ carácter industrial

Sólo son patentables los sistemas donde la estructura ha sido integrada en función de la organización lógica que le ha sido incorporada, a tal grado que un gran número de elementos de combinaciones son entonces apropiados al resultado nuevo, es decir, que hayan interconexión entre aquellos y la computadora. La combinación nueva, que constituida por un sistema automático integrado, puede ser protegida por la patente, pues ella satisface el criterio de novedad, actividad inventiva y carácter industrial.

• Derechos de autor: registro, regalías (R)

Derechos de autor.

El derecho a un privilegio sobre el uso de una propiedad científica, literaria o artística, reconocida y sancionada por el derecho positivo. Comprende para su titular, la facultad de disponer de su obra, publicarla, ejecutarla, representarla y exponerla en público, enajenarla, reducirla, adaptarla o autorizar su traducción y reproducción en cualquier forma.

- ✓ **Derechos reconocidos:** la protección de los derechos autor significa que ciertos usos determinados de una obra son reservados al titular y no pueden, fuera de él, ser efectuados más que con su autorización. Las más frecuentes son el derecho de copiar o reproducir todo tipo de obra. Algunos de estos derechos no son exclusivos, sino simples derechos para el titular a efecto de recibir una remuneración.
- ✓ **Adquisición del derecho de autor:** las leyes de varios países establecen que la protección es independiente de toda formalidad. La situación es diferente en aquellos países donde la adquisición de derechos autor está sujeta al cumplimiento de ciertas formalidades como el registro en una oficina gubernamental, el pago del impuesto de registro o de renovación y inserción de la mención de reserva del derecho de autor en los ejemplares publicados.
- ✓ **La duración:** la protección del derecho de autor está limitada en el tiempo. Muchos países han adoptado como regla general de 50 años luego de la muerte del autor de la obra, con algunas excepciones.

Las leyes de un país relativas al derecho de autor se refieren generalmente a los actos realizados en el mismo. La protección del derecho de autor fundada en la legislación nacional no tiene efectos más que en el país en cuestión y la protección a los países extranjeros está garantizada únicamente por la existencia de acuerdos internacionales.

Protección de programas:

- ✓ **Género de la obra:** los programas en su momento de realización utilizan letras que constituyen palabras acompañadas de diagramas.
- ✓ **Forma de expresión:** obras abstractas que encuentran una concretación a nivel de resultados. Todo programa adquiere diversas formas materiales o tangibles tales como escritas, gráficas, símbolos, cartas o bandas, suficientes para hablar de una forma de expresión.

- ✓ **Mérito:** la ley no juzga las obras, las protege ciegamente (fórmula de Pouillet).
- ✓ **Destino:** es indiscutible su fin utilitario y no únicamente bajo el aspecto o función operativa de la máquina sino también en el caso donde el programa es la solución a un problema concreto.
- ✓ **Originalidad:** los programas nuevos fueron nuevos y originales, los programas aparecidos posteriormente pueden ser tenidos como originales pero no como nuevos.

• "Piratería" de programas para computadora (R)

Piratería informática.

Acción que consiste en tener acceso de manera no autorizada a una computadora o a una red de computadoras. Copiado o uso ilegal de programas.

• Contratos y licencias de uso de software; categorías de licencias: organizacionales, académicas, individuales (R, E)

Licencia es, normalmente, un documento entregado por el fabricante del Software, que certifica el pago de los derechos de autoría del programa y que permite el uso de éste en una computadora, de acuerdo a términos contractuales que acompañan a esta licencia.

Al comprar un paquete corriente de software de computación, en realidad no está comprando el software, sino una licencia de software que le permite usar el programa en una máquina. Los acuerdos de licencia varían de una compañía a otra, pero la mayoría incluye limitaciones en cuanto al derecho de usuario de copiar discos, instalados software en discos duros y transferir información a otros usuarios. Casi todos los productos de software comercial en el mercado tienen derechos de autor, de modo que no pueden duplicarse legalmente para ser distribuidos a terceros; algunos discos (en su mayoría juegos) están protegidos contra copias, de manera física, para que no puedan copiarse. La creación de software es increíblemente costosa, debido a su dificultad. La mayoría de los creadores de software usan los derechos autor y la protección contra copias para asegurarse de que vendan suficientes copias de sus productos para recuperar su inversión, seguir operando y escribir más programas.

En lo correspondiente a la adquisición de equipo lógico, se debe hacer distinción entre los diferentes tipos de software existentes:

- ✓ Software de base
- ✓ Software empaquetado para microcomputadoras
- ✓ Software de aplicación

Entre las modalidades de contratación de productos de software disponibles en el mercado, se tienen:

- ✓ **La compraventa o licencia única:** es acordada generalmente con productos finales para computadoras personales y consiste como en el caso de compra, en el pago de una sola cantidad por concepto del producto;
- ✓ **La licencia de uso** (que podrá ser exclusiva o no, cedible o no cedible): es quizás la forma más extendida de contratación, y consiste en el alquiler del código ejecutable del sistema específico a contratar;
- ✓ **el contrato de desarrollo de software**
- ✓ **El mantenimiento:**
 1. Mantenimiento correctivo: tendrá lugar cuando se detecte algún fallo en el software desarrollado;
 2. Mantenimiento adaptativo: se hará presente como consecuencia de cambios en el entorno de operación en el que se haya implantado el sistema, como nuevas plataformas hardware, cambio de software de base, etc.
 3. Mantenimiento perfectivo: es consecuencia de nuevos requerimientos en cuanto a funcionalidad, rendimiento, etc. que podría tener la entidad, fruto de alguna evolución en su organización, requerimientos que van más allá de los objetivos que inicialmente se plantearon para el sistema.
 4. Mantenimiento preventivo: hace referencia a cambios que realiza el desarrollador con el fin de mejorar la confiabilidad o la facilidad de mantenimiento del software.

Acuerdos de Licencia

Un Acuerdo de Licencia es un acuerdo legal entre la compañía de software propietaria de los derechos de autor del software y la organización o el individuo que lo utiliza.

- ✓ **Las licencias individuales** se aplican a un solo usuario del software en una sola computadora. En algunos casos, el software puede ser usado por el mismo individuo en una computadora portátil. Sin embargo, esto

casi siempre se restringe al trabajo en el hogar más que para uso personal. También es característica la restricción que impide a cualquier otra persona usar el software.

- ✓ **Acuerdo de Licencia por Máquina:** puede llegar a permitir que una o más personas utilicen el software siempre que sólo lo haga una persona por vez. **Licenciamiento por Volumen:** permiten que los negocios y las organizaciones mayores obtengan software para un número determinado de computadoras. El valor del descuento aumenta con el volumen de licenciamientos adquiridos.
- ✓ **El Acuerdo de Licencia de "Sitio":** es otra forma de Acuerdo de Licencia por Volumen. Permite un uso ilimitado del software en un solo "sitio", como ser en una sola oficina, departamento o piso de una sola compañía, un edificio o todas las oficinas corporativas de una organización y los hogares de su personal. Los Acuerdos de Licencia Empresariales permiten un uso ilimitado del software en la totalidad de la organización.
- ✓ **Las Licencias para Red:** permiten a la organización adquirir un paquete de software para ser utilizado por todos los usuarios conectados a la Red de Área Local (LAN). En general, sólo se lo puede instalar en un solo servidor de la red y no puede ser usado por más empleados que los que tienen permiso de acceso a la red por parte del sistema operativo de ésta.
- ✓ **Una Licencia Concurrente:** posibilita a un número establecido de usuarios acceder al software al mismo tiempo. Su uso se halla restringido a los ambientes en los que haya instalación de red donde mantiene más bajos los costos de utilización de software.
- ✓ **Acuerdos de Licencia Especiales**
 1. **Software Beta.** Se trata de software que se encuentra en la etapa de prueba de su desarrollo. Típicamente, quien desarrolla el programa le permite a la organización probar software nuevo a cambio de información sobre cualquier problema que pudiera surgir.
 2. **Shareware o "Software de Prueba".** Está hecho según el concepto "probar antes de comprar". Se distribuye gratuitamente una versión del software al que le faltan características o no puede ser usado más que una cierta cantidad de veces o de días. Esto le da al usuario la oportunidad de probar el software y conseguir copias completas en la medida en que se necesiten.
 3. **Freeware (Gratis).** Este tipo de software se distribuye gratuitamente, sin costos de uso. El Freeware tiene sin embargo Acuerdos de Licenciamiento que necesitan ser respetados.
 4. **Dominio Público.** Con este tipo de software, quien detenta los derechos de autor ha resignado todos los derechos al software permitiéndole ser copiado y distribuido libremente. A diferencia del Freeware que no puede ser modificado o revendido a título lucrativo, el software de Dominio Público puede ser modificado o reenvasado para la venta.
 5. **Fuentes.** Habitualmente no son las fuentes mismas las licenciadas sino los métodos utilizados para crearlas. Así, es típico ver licencias basadas en Impresoras (en que la fuente sólo puede ser usada en un solo equipo de salida, independientemente de cuantas computadoras estén conectadas a éste) o licencias basadas en CPU (en las que la licencia es para una sola máquina).

• Propiedad industrial (patentes, marcas) (R)

La propiedad industrial.

Elemento constitutivo de la categoría más amplio del derecho de la propiedad intelectual, presentado ramas mayores especialmente significativas: las marcas y las patentes.

- ✓ **El derecho de marcas:** el derecho marcario norteamericano ofrece una protección a los programas, pero esta protección no se refiere sino al origen aparente del programa como tal, en tanto no haya engaño por parte del consumidor en función del origen del producto.
- ✓ **Derecho de patentes:** en cuanto a la patentabilidad de los programas, dos decisiones pronunciadas en 1980 por la Corte Suprema de Estados Unidos indican que las invenciones reveladoras de un programa constituyen un objeto patentable a condición de que no sea reivindicada la apropiación global de una fórmula o un algoritmo matemático.

Las marcas.

Es un signo distintivo que permiten a su titular (fabricante o comerciante) distinguir sus productos o servicios de aquellos de la competencia. Signo que tiende a procurar a la clientela, la mercancía o el servicio que busca y paralelamente a la empresa, una clientela apegada a la marca.

La organización mundial de la protección intelectual (OMPI) definen: un signo visible protegido por un derecho exclusivo concedido en virtud de la ley, que sirve para diferenciar las mercancías de una empresa de las mercancías de otra empresa.

Funciones:

- ✓ asegurar al producto un carácter distintivo
- ✓ asegurar la protección del producto partiendo de la empresa
- ✓ asegurar una garantía de calidad
- ✓ asegurar una indicación de origen
- ✓ asegurar una publicidad

Las patentes.

ONU definen: un privilegio legal concedido por el gobierno a los inventores y otras personas, donde los derechos derivan para el inventor durante un cierto tiempo para qué otras personas produzcan, utilicen o vendan un producto patentado. Una vez finalizado el tiempo de privilegio al beneficiario, la invención patentada cae al dominio público.

OMPI: un documento emitido a solicitud presentada ante la institución nacional respectiva que describe una invención y crea una situación jurídica en la cual la invención patentada puede ser explotada normalmente bajo la reserva de la autorización del titular de la patente.

La protección que confiere la patente está limitada de 15 a 20 años. La invención debe ser una solución a un problema concreto en la esfera de la tecnología sea como producto o como proceso.

2. Impacto de la tecnología

• Historia tecnológica (R)

• Automatización de procesos (R, E)

Automatización.

Sistema de fabricación diseñado con el fin de usar la capacidad de las máquinas para llevar a cabo determinadas tareas anteriormente efectuadas por seres humanos, y para controlar la secuencia de las operaciones sin intervención humana. El término automatización también se ha utilizado para describir sistemas no destinados a la fabricación en los que los dispositivos programados o automáticos pueden funcionar de forma independiente o semi-independiente del control humano.

La fábrica automatizada.

En ellas emplean robots para pintar, soldar y efectuar otras tareas repetitivas en la línea de montaje. Las computadoras también ayudan a rastrear el inventario, programar la entrega de piezas, controlar la calidad de los productos, vigilar el desgaste de las máquinas y programar el mantenimiento.

Es más eficiente que la fábrica tradicional por dos razones:

- ✓ la automatización permite una mayor integración de la planificación y la manufactura, reduciendo así el tiempo de inactividad que los materiales y las máquinas
- ✓ La automatización reduce el desperdicio en términos de instalaciones, materias primas y mano de obra.

La oficina automatizada.

La automatización de las oficinas inició la época de los macrocomputadores, cuando se usaron los computadores para trabajos internos como la contabilidad de la nómina. El trabajo pasó de los macrocomputadores a los computadores de escritorio y la gente uso con computadores personales para las cuales no estaban programados los macrocomputadores. Hoy en día, casi todas las organizaciones han reconocido la importancia de los computadores personales en estructura de computación.

La choza electrónica.

La tecnología actual de telecomunicaciones abre las puertas a nuevas posibilidades para que los trabajadores modernos regresen a casa para ganarse allí la vida. El término "trabajador desde el hogar" se aplica a todos aquellos individuos cuyas labores se relacionan con la información y las realizan en casa, ya sea que se comuniquen por modem o no. El término choza electrónica se usa para describir un hogar en el cual la tecnología moderna hace posible trabajar en casa.

Argumentos a favor del trabajo desde el hogar:

- ✓ reduce los viajes por automóvil
- ✓ ahora tiempo

- ✓ permite horarios más flexibles
- ✓ puede aumentar la productividad que

• Educación, capacitación y entrenamiento (E, RP)

El papel de la tecnología, en este caso la informática, en la educación es no solamente estratégico, sino urgente. Para poder entonces cernir el campo del trabajo entre computación y educación hay que distinguir dos cuestiones que aunque diferentes, se mezclan entre ellas.

- ✓ **La informática como medio (instrumental y cognitivo) para el proceso de la enseñanza / aprendizaje y la educación en general.**

Este concepto es denominado también como “educación basada en computadora” y comprende a su vez dos aspectos:

1. La “enseñanza asistida por computadora” (E.A.C.), o en términos derivados del enfoque pedagógico más actual el “aprendizaje asistido por computadora”, que comprendería a aquellos aspectos computacionales que de alguna manera directa asisten o ayudan al proceso de enseñanza o aprendizaje.
2. El aspecto de la administración de la educación asistida por computadora, es decir, los recursos de informática que la escuela y el docente se pueden valer para administrar el servicio educativo a los educandos.

- ✓ **La informática como objeto del proceso de enseñanza / aprendizaje; es decir, la enseñanza e investigación de la computadora.**

Los programas de licenciatura en términos generales, se agrupa alrededor de 2 prototipos básicos: un modelo de “ingeniería en computación”, con un contenido técnico proporcionalmente alto en computación, matemáticas e ingeniería y un modelo de “licenciatura en informática” orientado primordialmente hacia la administración y la contabilidad.

Los programas de educación continua en informática de instituciones educativas son escasos y costosos, lo que dificulta que la población acceda a ellos.

Se propone:

1. Evaluación y actualizar los planes de estudio de los programas en informática de los niveles técnico y de licenciatura para elevar su calidad.
2. Elevar la cantidad y calidad de programas de estudio en informática.
3. Fortalecer la infraestructura física y humana de las instituciones educativas públicas y privadas que ofrecen programas de formación en informática.
4. Incorporar contenidos curriculares en informática dentro de los programas de educación básica, media superior y superior en otras disciplinas.

• Robotización (R, E)

Robótica.

Conjunto de estudios y técnicas de concepción y funcionamiento de robots que efectúan labores determinadas adaptadas a su entorno.

En el lenguaje diario, se dice que un proceso está “robotizado” cuando los seres humanos que la llevaban a cabo han sido sustituidos por uno o varios robots.

Robot: su etimología es *robo*ta, palabra checa que significa trabajo forzado.

La visión, audición, el reconocimiento de patrones, la ingeniería de conocimiento, la toma de decisiones, la comprensión del lenguaje natural, el habla, todo se une en los robots actuales.

Leyes de la robótica de Isaac Asimov.

- ✓ Un robot no puede lesionar a un ser humano ni permitir, por su omisión, que un ser humano sea lastimado.
- ✓ Un robot debe obedecer órdenes que reciba de los seres humanos, excepto si estas órdenes entran en conflicto con la primera ley.
- ✓ Un robot debe proteger su propia existencia siempre y cuando dicha protección no entre en conflicto con la primera o segunda leyes.

Las diferencias más importantes de hardware entre los robots y otros computadores son los periféricos de entrada y salida. En vez de enviar la salida a una pantalla o una impresora, un robot envía órdenes a articulaciones, brazos y otras partes móviles. Los primeros robots no tenían dispositivos de entrada correspondientes para supervisar sus movimientos y el ambiente. La mayoría de los robots modernos cuentan con algún tipo de sensores de entrada, con los que pueden corregir o modificar sus acciones con base en la retroalimentación del mundo exterior.

Los robots ofrecen varias ventajas:

- ✓ Se instalan para ahorrar costos de mano de obra.
- ✓ Pueden mejorar la calidad y elevar la productividad.
- ✓ Son ideales para trabajos peligrosos o incómodos para los humanos.

• **Tareas peligrosas para el ser humano y elementos tecnológicos de reemplazo o protección (E)**

La prevención de riesgos laborales, desde el punto de vista social, tanto por las desgracias personales como por el cambio cultural que implica sustituir las tareas peligrosas por las que no lo sean, incitan a apoyarnos en las diferentes áreas de conocimiento de las ciencias y la tecnología para determinar los riesgos inherentes a los puestos de trabajo, evaluar los mismos y planificar su corrección.

Las computadoras frecuentemente permiten construir dispositivos más interactivos, más acoplados, y más susceptibles a los errores, y pueden introducir complejidad innecesaria y peligrosa. El poder de las computadoras es que es una máquina completamente general. Uno puede crear otras máquinas solamente especificando los pasos necesarios para alcanzar los fines. Esta ventaja ha producido un aumento explosivo en el uso de lo las computadoras, que incluye su introducción en sistemas peligrosos.

El papel de las computadoras en accidentes

Hay pocos sistemas hoy que no usan computadoras para proveer control o apoyar el diseño. Las computadoras ahora controlan muchos dispositivos críticos a la seguridad, y frecuentemente reemplazan la poca seguridad de hardware (o sino los reemplazan, los controlan).

En la seguridad se pueden usar computadoras en maneras distintas:

- ✓ Proveer información a un controlador humano a petición.
- ✓ Interpretar los datos y mostrarlos al controlador, que hace las decisiones de control.
- ✓ Emitir los comandos directamente con un monitor humano.
- ✓ Eliminar el humano completamente.

• **Consecuencias (E)**

3. Impacto social de la informática

• **Cultura informática general (R)**

La sociedad de la información.

La sociedad de la información implica el uso masivo de tecnologías de la información y de comunicación para difundir el conocimiento e intercambio en una sociedad. Lo propio de la sociedad de la información es la creación del conocimiento científico, la aplicación de dicho conocimiento, la tecnología y la difusión de la misma entre los actores económicos, por tanto, innovación productiva.

La Informática es un lenguaje que refleja una nueva manera de pensar, de formular problemas, de modelizarlos, concretarlos y procesarlos. La Informática suministra herramientas para manejar y dominar los problemas relacionados con la información del mundo actual y que hasta ahora eran considerados con una simplicidad asombrosa e incluso fueron evitados.

• **La tecnología informática en diferentes niveles de educación (R)**

Educación en la era de la información.

Cuando los computadores personales aparecieron a salón de clases, los estudiantes comenzaron a usar software de ejercicios y prácticas. Un programa tradicional de ejercicios y prácticas presenta una pregunta estudiante y compara su respuesta con la correcta. Los programas de ejercicios y prácticas no enseñan material nuevo, están diseñados para qué los estudiantes repasen temas que ya aprendieron, para que mejoren. El software didáctico (tutor) proporcionan enseñanza directa a cerca de una habilidad o tema específico. Con frecuencia software ejercicios y prácticas y el software didáctico se denominan software de enseñanza asistida por computadora.

Este ofrece varias ventajas:

- ✓ aprendizaje individualizado
- ✓ motivación
- ✓ confianza

Aprendizaje distancia: escuelas virtuales.

Para algunos estudiantes, la aplicación pedagógica más importante de los computadores es el aprendizaje a distancia: el uso de la tecnología para extender el proceso educativo fuera de los confines de la escuela. La tecnología de las telecomunicaciones es muy importante para los estudiantes en lugares remotos. El aprendizaje de distancia también es prometedor para los obreros cuyos puestos o modificados por eliminados por los cambios económicos, puesto que las universidades ofrecen programas electrónicos externos, estas personas pueden actualizar sus habilidades sin abandonar sus comunidades. La escuela en línea es muy atractiva para los estudiantes de mayor edad, cuyos empleos les impiden asistir a universidades más tradicionales. Las escuelas en línea son un paso importante para un sistema educativo que estimula el aprendizaje durante toda la vida.

La revolución tecnológica de la computación.

En una perspectiva macro-económica internacional existe una división entre la industria y el sector gubernamental referente a la investigación y desarrollo, dentro de la cual destaca el sistema educativo. La industria invierte en investigación y desarrollo de tecnología para desarrollar productos nuevos o para mejorarlos, mientras que el sistema educativo se dedica al alcance y difusión general del conocimiento, al entrenamiento de personal para la investigación y el desarrollo y eventualmente a realizar investigación básica.

Lo anterior está cambiando, en el sistema educativo la tecnología se ha desarrollado dirigiéndose hacia distintos enfoques y aún hacia investigar cómo con la tecnología el fenómeno educativo se contamina de un maquinismo desbocado y de un conductismo tecnocrático que lo llevan a la deshumanización. Sin embargo, es claro y aceptado que esas no son las pretensiones de tecnología educativa, sino tal vez algunas de sus consecuencias indeseadas. Con la tecnología educativa se pretende lograr un proceso más productivo e individual, brindar una educación con bases eminentemente científicas, hacer de la enseñanza un fenómeno significativo y en consecuencia, lograr un aprendizaje eficaz a la vez que aboga por una ampliación de la cobertura educativa y a la aplicación de manera sistemática del conocimiento científico y tecnológico a la solución de problemas educativos.

• Bondades y riesgos (R)

Bondades.

- ✓ **Las computadoras pueden facilitar el auto-aprendizaje pasivo.** En el modo de CAI, por ejemplo, las computadoras individualizan el auto - aprendizaje mientras dan apoyo y retroalimentación inmediata.
- ✓ **Las computadoras son interactivas.** Los sistemas de microordenador incorporan varios paquetes de software, son sumamente flexibles y aumentan al máximo el control del aprendizaje.
- ✓ **La tecnología de las computadoras está avanzando rápidamente.** Las innovaciones constantemente están emergiendo, mientras que, los costos relativos disminuyen.
- ✓ **La computadora incrementa el acceso:** redes de trabajo locales, regionales, y nacionales enlazan recursos e individuos, dondequiera que ellos pudieran estar. De hecho, muchas instituciones ahora ofrecen una completa gama de programas de estudio que se basan casi exclusivamente en recursos computacionales.

Riesgos.

- ✓ **Las redes de Computación son costosas de desarrollar.** Aunque las computadoras individuales son relativamente baratas y el mercado de hardware y software es muy competitivo, todavía es costoso desarrollar redes instruccionales y comprar el software necesario para ejecutar el sistema.
- ✓ **La tecnología está cambiando rápidamente.** La tecnología de la computadora evoluciona tan rápidamente que el educador a distancia hace foco solamente en la innovación sin poder satisfacer sus necesidades tangibles; constantemente cambiará su equipamiento en un esfuerzo por estar al tanto de los últimos avances tecnológicos.
- ✓ **El analfabetismo computacional todavía es amplio.** Mientras se han usado computadoras desde los años 60's. Todavía hay muchas personas que no tienen acceso a computadoras o redes.

• Posibilidades de estratificación y separación de la sociedad (R, E)

- **Diferencias generacionales (R)**

Ábaco.

El ábaco fue uno de los primeros instrumentos manuales para calcular, y aunque tiene una gran antigüedad (3,000 años) todavía se sigue utilizando.

Calculadora de pascal.

En 1642, Blaise Pascal desarrolló una calculadora mecánica para facilitarle el trabajo a su padre, un funcionario fiscal. Los números se introducen en las ruedas metálicas delanteras y las soluciones aparecen en las ventanas superiores. Unos 30 años después Leibniz, filósofo, matemático y estadista alemán, mejora la maquina de pascal para que esta pudiera multiplicar, dividir, extraer raíces además de sumar y restar.

Charles Babbage.

La máquina diferencial, que muchos consideran un precursor de las calculadoras modernas, era capaz de calcular tablas matemáticas. La máquina analítica, ideada también por Babbage, habría sido una auténtica computadora programable si el proyecto hubiera contado con la financiación adecuada. Sin embargo, ninguna de las dos máquinas se completó en vida de Babbage, aunque habría sido posible con la tecnología de la época.

Hollerith, Hermann

Inventó un método de codificación de datos en fichas o tarjetas en las que mediante perforaciones se inscriben datos numéricos o alfabéticos. Este sistema resultó ser de gran utilidad en trabajos estadísticos y fue muy importante en el desarrollo de los ordenadores o computadoras digitales. La máquina de Hollerith, utilizada en 1890 para realizar el censo de los Estados Unidos, leía la información a través de unos contactos eléctricos. Creó la Tabulating Machine Company (1896), que está considerada como una predecesora de la IBM (International Business Machines Corporation).

Atanasoff, John Vincent

Pretendió ser el que desarrolló ciertas técnicas básicas utilizadas posteriormente en el diseño del primer ordenador o computadora electrónica digital, (1946)ENIAC. Construyó un sencillo mecanismo de cálculo consistente en un tubo de vacío, que enseñó a diversas personas, incluido uno de los últimos fabricantes de ENIAC; pero no tuvo éxito en la difusión de su mecanismo. ENIAC fue la primera computadora electrónica de aplicación general que entró en funcionamiento. ENIAC podía realizar 300 multiplicaciones por segundo, lo que la hacía 300 veces más rápida que cualquier otro dispositivo de la época. Ocupaba el espacio de una casa de tres recámaras y tenía un peso de 30 toneladas además de programarse por medio de interruptores por tener más de 18,000 bulbos.

La UNIVAC (computadora automática universal).

La UNIVAC fue la primera computadora adquirida por un negocio para procesar y guardar (archivar) información en el año de 1954. Y la IBM 650 introducida a finales de 1954 dio gran liderazgo a la empresa IBM para ser líder en la producción de computadoras en 1955.

Circuito integrado

Pequeño circuito electrónico utilizado para realizar una función electrónica específica, como la amplificación. Se combina por lo general con otros componentes para formar un sistema más complejo y se fabrica mediante la difusión de impurezas en silicio monocristalino, que sirve como material semiconductor, o mediante la soldadura del silicio con un haz de electrones.

La llegada del microprocesador.

En el año de 1965 los componentes de las computadoras se duplicaron haciendo que la gente se preocupara por su tamaño, así en el año de 1969 Víctor Poor y Harry Pyle desarrollaron un modelo de pastilla y lo llevó a dos fabricantes Texas Instrument e INTEL Corporation. A finales de 1969 un ingeniero de INTEL presentó una pastilla de circuitos que se podía programar a una compañía de calculadoras japonesas.

En 1961 se crea el procesador 4004 de Intel que se podía programar, en 1971 producía una pastilla más poderosa la 8008 y en 1974 Intel produjo el procesador 8080 el cual sentó las bases para el desarrollo de las computadoras personales o PCs.

Microprocesador 8086.

Es un descendiente directo del 8080, pero con registros de 16 bits, un bus de datos de 16 bits y direccionamiento de 20 bits, y permite controlar más de un megabyte de memoria. Está disponible con velocidades de 4,77, 8 y 10 MHz. Los modelos 25 y 30 de los equipos PS/2 de IBM, disponen de un 8086 a 8 MHz.

Microprocesador 68881

Coprocesador matemático, o de coma flotante, de la firma estadounidense Motorola para los procesadores 68000 y 68020. Los coprocesadores matemáticos aceleran cualquier función de cálculo matemático siempre que las aplicaciones los admitan. El 68881 acelera los procesos mediante un conjunto adicional de instrucciones aritméticas de coma flotante muy complejas, un conjunto de registros de datos en coma flotante y 22 constantes incorporadas que incluyen potencias de 10.

Microprocesador 80286

Denominado también 286. Se trata de un microprocesador de 16 bits de Intel, presentado en 1982 e incluido desde 1984 en el equipo PC/AT de IBM y compatibles. El 80286 dispone de registros de 16 bits, transfiere información a través del bus de datos a 16 bits simultáneos y utiliza 24 bits para direccionar la memoria. El 80286 puede operar en dos modos, el real (que es compatible con MS-DOS y con los límites de los chips 8086 y 8088) y el protegido (que potencia la funcionalidad del microprocesador). El modo real limita a 1 megabyte la cantidad de memoria que el microprocesador puede direccionar. Por otro lado, en el modo protegido, el 8086 puede acceder directamente a 16 megabytes de memoria.

Microprocesador 80386

Denominado también 386SX en informática. Se trata de un microprocesador de Intel, introducido en 1988 como un producto de bajo costo alternativo al 80386DX. El 80386SX es básicamente un procesador 80386 X limitado por un bus de datos de 16 bits. El diseño basado en 16 bits permite configurar los sistemas 80386SX con componentes menos costosos del tipo AT, reduciendo considerablemente el precio total del sistema. El 80386SX proporciona además prestaciones superiores al 80286 y compatibilidad con todo el software diseñado para el 80386DX. Incorpora también características del 80386DX, como la multitarea y el modo 8086 virtual.

Microprocesador 80387

Denominado también 387 en el campo de la informática. Se trata de un coprocesador matemático, también denominado de coma flotante, diseñado por Intel para la familia de procesadores 80386. Está disponible a velocidades de 16, 20, 25 y 33 MHz. El coprocesador 80387 puede aumentar de forma considerable el rendimiento del sistema, siempre que el software de aplicación haga uso de él, ya que pone a disposición de la aplicación instrucciones aritméticas, trigonométricas, exponenciales y logarítmicas con las que no cuenta el 80386. El 80387 también incorpora operaciones fundamentales para el cálculo de senos, cosenos, tangentes, arcotangentes y logaritmos.

Microprocesador 80486

Denominado también 80486DX o 486DX, microprocesador Intel de 32 bits lanzado al mercado en 1989. Su característica principal es la incorporación de un coprocesador matemático integrado. El 486 es un procesador de 32 bits en el bus de datos y 32 bits en el bus de direcciones. Integra 1.200.000 transistores y se fabricó para tres velocidades de reloj (25 MHz, 33 MHz y 50 MHz).

Microprocesador Pentium

Microprocesador lanzado al mercado por Intel en 1993, sucesor del 486. Según la sucesión lógica, debería haberse llamado 586 o 80586, pero Intel lo denominó Pentium debido a razones de copyright. El Pentium es un microprocesador superescalar de 32 bits con un bus de datos externo de 64 bits, que contiene 3.100.000 transistores y coprocesador matemático incorporado. Las primeras versiones de este procesador tenían una frecuencia de reloj de 60 MHz, con una alimentación eléctrica de 5 voltios, un bus de direcciones de 32 bits y un bus de datos externo de 64 bits.

• Aplicaciones en diversas áreas de conocimiento y en distintas actividades humanas (R, E)

- ✓ La recepcionista utiliza la computadora para grabar mensajes, localizar empleados y para tareas administrativas.
- ✓ El departamento de ventas coteja la disponibilidad del producto y el crédito del cliente.

- ✓ En envío y recibo utilizan la computadora para entrar transacciones manteniendo actualizados los registros de inventario y venta.
- ✓ En el área de manufactura la utilizan para hacer el itinerario de producción y registrar los costos de los artículos producidos.
- ✓ El departamento de recursos humanos mantiene la pista de los empleados pasados o actuales, además de los adiestramientos y destrezas de los empleados.
- ✓ En la educación la computadora es un medio que fortalece el proceso enseñanza-aprendizaje. Se están utilizando los programas de aplicaciones como, por ejemplo: procesadores de palabras, hojas electrónicas y base de datos. También, se ha hecho popular el uso de Internet. El uso de multimedia y correo electrónico. Otro componente que está tomando mucha popularidad es el de educación a distancia.
- ✓ La profesión médica utiliza la computadora en el diagnóstico y monitoreo de los pacientes y para regular los tratamientos. Está utilizando bases de datos médicos de investigaciones recientes con hallazgos y tratamientos.
- ✓ Los científicos usan la computadora para analizar el sistema solar, seguir los patrones del tiempo y llevar a cabo experimentos. Los científicos formulan las hipótesis y luego las prueban a través de la observación y colección de datos. En ocasiones tienen que simular el comportamiento del mundo real y su medio ambiente para comprobar la veracidad de sus teorías.
- ✓ En el área de publicaciones la computadora se está utilizando en los medios de impresión, tales como: revistas, magazines y periódicos. Las páginas son creadas en la computadora, se pueden añadir gráficas o fotos de diferentes medios, que son guardadas en disco flexible, y luego se imprimen en una impresora láser a color.
- ✓ En el gobierno la computadora es usada en todos los niveles. La utilizan para un funcionamiento más eficiente, efectivo y democrático.
- ✓ En los deportes el uso de las telecomunicaciones nos permite el disfrute de los deportes en el momento en que estén ocurriendo no importa el lugar del mundo.
- ✓ En el arte han provisto al artista de menos herramientas para crear arte tradicional como nuevas formas de crear arte.
- ✓ En la fotografía el uso de cámaras digitales y sus programas han permitido al fotógrafo digitalizar, almacenar y presentar las fotos en una computadora.
- ✓ En la música la grabación digitalizada ha sustituido la grabación análoga de cintas (tapes). Se están utilizando los sintetizadores que pueden reproducir los tonos complejos de cualquier instrumento musical.
- ✓ En el baile programas como "Life Forms" han ayudado a los coreógrafos a crear pasos de baile en 3D, y luego son guardados para el futuro.
- ✓ En el teatro las computadoras juegan un papel importante en el diseño de los escenarios, control de luces y efectos especiales.
- ✓ En la producción de películas, comerciales y programas de TV las computadoras han revolucionado la creación de éstos. Las computadoras generan gráficas, animación y efectos especiales.

• Herramientas y ambientes para el usuario final (E)

Los programas de aplicación, o aplicaciones, son las herramientas de software que permiten usar un computador para fines específicos.

- ✓ **Procesamiento de texto y publicación electrónica.**

El procesamiento de textos es fundamental para cualquier persona que se comunique en forma escrita, y es la aplicación principal de que se valen los estudiantes. En la publicación electrónica se usa computadora personal para transformar cualquier texto en una publicación refinada, con atractivo visual.

- ✓ **Hojas de cálculo y otras aplicaciones de procesamiento numérico.**

En las empresas, la hoja de cálculo electrónica es la aplicación de computadores personales que paga el alquiler o que al menos lo calcula. Las hojas de cálculo y el software para estadísticas puedan ayudar a convertir cantidades abstractas en conceptos concretos.

- ✓ **Bases de datos para almacenamiento y recuperación información.**

Las bases de datos predominan en el mundo de los macrocomputadores. Cada vez más bibliotecas, bancos y otras instituciones adopta las bases de datos para almacenar información.

- ✓ **Telecomunicaciones y redes.**

Cada día hay más computadores conectados en redes, de manera que pueden intercambiar información, estamos entrando en una era en que las redes se han convertido en la norma. Una conexión en red es una

puerta a un mundo de buzones electrónicos, tableros de noticias, servicios de bases de datos comerciales y otras herramientas de comunicación.

✓ **Graficador por computador.**

Los computadores también pueden generar todo tipo de gráficos, desde los diagramas y gráficas de las hojas de cálculo hasta animaciones tridimensionales realistas.

✓ **Multimedia e hipermedia.**

La herramienta multimedia para computadores personales permiten combinar audio y video con los textos y gráficos tradicionales, añadiendo nuevas dimensiones a la comunicación por computador. Las herramientas de hipermedia se centran en las capacidades interactivas de los computadores. Con la hipermedia los usuarios pueden explorar diversas trayectorias por las fuentes de información. La combinación de multimedia e hipermedia tienen un potencial casi inimaginable para transformar la manera en que vemos la información y trabajamos con ella.

✓ **Inteligencia artificial.**

Es la rama de las ciencias de la computación que explora el uso de los computadores en tareas que requieren inteligencia, imaginación e introspección, tareas que tradicionalmente han sido realizadas por seres humanos y no por máquinas.

✓ **Resolución de problemas generales.**

Las computadoras se usan para resolver problemas. La mayoría emplea software de aplicación escrito por programadores profesionales, pero algunos tipos de problemas requieren programación especializada. Los lenguajes de programación no son aplicables, sino herramientas que permiten construir y adecuar aplicaciones.

• **Sociología de la informática (R)**

- ✓ El acceso a la información y el libre flujo de la información son derechos humanos fundamentales.
- ✓ Las tecnologías de la información y la comunicación son capitales para la creación de la sociedad mundial de la información y desempeñan un importante papel en la lucha contra la pobreza y la desigualdad a escala mundial.
- ✓ En el entorno cambiante que caracteriza a la sociedad de la información, el desarrollo de recursos humanos es un requisito continuo y fundamental.
- ✓ La diversidad cultural y lingüística constituyen un pilar dentro de una sociedad de la informática bien lograda; se puede estimular y apoyar la creatividad en la concepción, tratamiento, divulgación y conservación de contenidos locales mediante un equilibrio adecuado entre los derechos de propiedad intelectual y las necesidades de los usuarios de la información.
- ✓ La sociedad civil, las empresas y el espíritu empresarial deben desempeñar el papel que les corresponde para aplicar el poder de la información y el conocimiento con el fin de promover un desarrollo económico, social y cultural sostenible.
- ✓ La sociedad de la información debe contribuir a mejorar la vida de todos los ciudadanos.
- ✓ La participación de múltiples personas y organismos interesados en alianzas nacionales, regionales y mundiales es un ingrediente esencial para alcanzar los objetivos de la sociedad de la información.
- ✓ La habilitación e integración son características y objetivos fundamentales de la sociedad de la información.
- ✓ La confianza y la seguridad son esenciales para el funcionamiento cabal de la sociedad de la información.
- ✓ Deben elaborarse estrategias para evaluar los aspectos ambientales de las tecnologías de la información y comunicación y hacer frente a los mismos.
- ✓ El fomento al desarrollo de las tecnologías de la información y comunicación debe tener en cuenta las características particulares de todas las regiones y todos los países, ya sean de carácter geográfico, demográfico o cultural.

MATEMÁTICAS

A. Matemáticas básicas

I. Cálculo

1. Diferencial en una variable

• Derivada de una función de una variable (R, E, RP)

El concepto de derivada lo podemos encontrar en la actualidad en prácticamente cualquier ciencia. La derivada, junto con la práctica totalidad del cálculo infinitesimal básico, fue inventada prácticamente al mismo tiempo aunque de manera independiente por Newton y Leibnitz.

Definición de la Derivada:

Sea una función $f(x)$ y consideremos la siguiente fracción:

$$\frac{f(x_0) - f(x)}{x_0 - x} = \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

Esta fracción, además del significado geométrico que veremos en seguida, se puede interpretar como "lo que crece la función por unidad de x ", o la rapidez de crecimiento de la función a lo largo del eje x . El lector que tenga problemas para visualizar esto podrá verlo con facilidad si sustituye funciones concretas, por ejemplo en lugar de $f(x)$ se puede utilizar $s(t)$ (la función que

$$\frac{s(t + \Delta t) - s(t)}{\Delta t} = \frac{\Delta s}{\Delta t}$$

dice el camino recorrido por un cuerpo en el instante t), y en lugar de x la propia variable t . Con esta sustitución, la fracción quedaría como:

que es precisamente la velocidad *media* del cuerpo en el intervalo $[t, t + \Delta t]$.

Pues bien, se define la **derivada** de la función $f(x)$ en el punto x_0 [se escribe $f'(x_0)$ - con un apóstrofe siguiendo al nombre de la función] como el límite

$$f'(x_0) = \lim_{x \rightarrow x_0} \frac{f(x_0) - f(x)}{x_0 - x} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

Intuitivamente, esto significa que vamos reduciendo cada vez más el intervalo en el que efectuamos el promedio de crecimiento de la función, hasta el punto de dejar reducido ese intervalo a un punto. La derivada representa, pues, **el crecimiento instantáneo de una función en un punto**. Es importante señalar que nada obliga a que este límite exista, pero si existe, se dice que la función es **derivable** en el punto x_0 .

En muchas ocasiones veremos la derivada expresada de las siguientes formas:

$$f'(x) = \frac{dy}{dx}$$

$$f'(x_0) = \left(\frac{dy}{dx} \right)_{x=x_0}$$

donde dy y dx se llaman *infinitésimos*, y se asumen infinitamente pequeños. Tanto dy como dx son un único símbolo, es decir, no podemos "cancelar" la parte "d" del numerador y denominador para que solamente quede y/x .

Cuando la función que se está derivando es una función que depende del tiempo, se ha

$$\dot{s} = s'(t) = \frac{ds}{dt}$$

extendido indicar la derivada con un punto encima de la función:

Reglas para Encontrar Derivadas:

$$\frac{d}{dx}(c) = 0$$

$$\frac{d}{dx}(x) = 1$$

$$\frac{d}{dx}(u+v+\dots) = \frac{d}{dx}(u) + \frac{d}{dx}(v) + \dots$$

$$4. \frac{d}{dx}(cu) = c \frac{d}{dx}(u)$$

$$5. \frac{d}{dx}(uv) = u \frac{d}{dx}(v) + v \frac{d}{dx}(u)$$

$$6. \frac{d}{dx}(uvw) = uv \frac{d}{dx}(w) + uw \frac{d}{dx}(v) + vw \frac{d}{dx}(u)$$

$$7. \frac{d}{dx}\left(\frac{u}{c}\right) = \frac{1}{c} \frac{d}{dx}(u), c \neq 0$$

$$8. \frac{d}{dx}\left(\frac{c}{u}\right) = c \frac{d}{dx}\left(\frac{1}{u}\right) = -\frac{c}{u^2} \frac{d}{dx}(u), u \neq 0$$

$$9. \frac{d}{dx}\left(\frac{u}{v}\right) = v \frac{d}{dx}(u) - u \frac{d}{dx}(v), v \neq 0$$

$$\frac{v^2}{v^2}$$

$$10. \frac{d}{dx}(x^m) = mx^{m-1}$$

$\frac{dx}{dx}$

$$11. \frac{d}{dx}(u^m) = mu^{m-1} \frac{d}{dx}(u)$$

- Problemas que requieren el concepto de la derivada y de sus interpretaciones geométricas y físicas (R, E, RP)

Encuentre la pendiente de la recta tangente a la gráfica de f en el punto $P(a, f(a))$.

$$\begin{aligned} 1. \quad f(x) &= 2-x^3 \\ &= \underline{d}(2) - \underline{d}(x^3) \\ &\quad dx \quad dx \\ &= 0 - 3x^2 \\ &= -3x^2 \end{aligned}$$

$$\begin{aligned} 2. \quad f(x) &= 3x-5 \\ &= 3 \underline{d}(x) - \underline{d}(5) \\ &\quad dx \quad dx \\ &= 3(1) - 0 \\ &= 3 \end{aligned}$$

$$\begin{aligned} 3. \quad f(x) &= \sqrt{x+1} \\ &= \frac{1}{2}(x)^{-1/2} \underline{d}(x) + \underline{d}(1) \\ &\quad dx \quad dx \\ &= \frac{1}{2}(x)^{-1/2} \\ &\quad 2 \\ &= \frac{1}{2\sqrt{x}} \end{aligned}$$

$$\begin{aligned} 4. \quad f(x) &= \frac{1}{x} - 1 \\ &= \underline{d}\left(\frac{1}{x}\right) - \underline{d}(1) \\ &\quad dx \quad dx \\ &= x \underline{d}\left(\frac{1}{x}\right) - (1)\underline{d}(x) - 0 \\ &\quad dx \quad dx \\ &= -\frac{1}{x^2} \end{aligned}$$

La posición de un punto P moviéndose sobre una recta coordenada l está dada por $f(t)$ donde t está medido en segundos y $f(t)$ en centímetros. Encuentre la velocidad media de P en el siguiente intervalo de tiempo:

$$\begin{aligned} f(t) &= 4t^2 + 3t \text{ en el intervalo } (1, 1.2) \\ &= 4 \underline{d}(t^2) + 3 \underline{d}(t) \\ &\quad dx \quad dx \\ &= 4(2t) + 3(1) \end{aligned}$$

$$= 8t + 3$$

Sumando los puntos del intervalo y dividiendo entre 2

$$\frac{1+1.2}{2} = 1.1$$

Sustituyendo en la derivada resultante

$$\begin{aligned} &= 8(1.1) + 3 \\ &= \mathbf{11.8} \end{aligned}$$

Hallar $f'(x)$

$$\begin{aligned} f(x) &= \sqrt{3x+1} = (3x+1)^{1/2} \\ f'(x) &= \underline{1} (3x+1)^{-1/2} \underline{d} (3x+1) \\ &= \underline{\frac{2}{2}} (3x+1)^{-1/2} (3) \\ &= \underline{\frac{3}{2\sqrt{3x+1}}} \end{aligned}$$

Derivar las siguientes funciones:

$$1. g(w) = \underline{1}$$

$$\begin{aligned} &\underline{-w^4} (0) \underline{-1} (4w^3) \\ &\underline{-4w^3} = -4w^{3-8} \\ &\underline{w^8} \\ &= \mathbf{-4w^{-5}} \end{aligned}$$

$$2. g(x) = (x^3-7)(2x^2+3)$$

$$\begin{aligned} &= (x^3-7) \underline{d} (2x^2+3) + (2x^2+3) \underline{d} (x^3-7) \\ &\quad dx \qquad \qquad \qquad dx \\ &= (x^3-7)(4x) + (2x^2+3)(3x^2) \\ &= 4x^4 - 28x^3 + 6x^4 + 9x^2 \\ &= \mathbf{10x^4 + 9x^2 - 28x} \end{aligned}$$

$$3. f(x) = \underline{4x-5}$$

$$\begin{aligned} &- \frac{(3x+2)\underline{d} (4x-5) - (4x-5)\underline{d} (3x+2)}{(3x+2)^2} \\ &- \frac{(3x+2)(4)-(4x-5)(3)}{(3x+2)^2} \\ &- \frac{12x+8-12x+15}{(3x+2)^2} = \frac{23}{(3x+2)^2} \end{aligned}$$

$$4. h(z) = \frac{8-z+3z^2}{2-9z}$$

$$\begin{aligned}
 &= \frac{(2-9z)\underline{d}(8-z+3z^2)-(8-z+3z^2)\underline{d}(2-9z)}{(2-9z)^2} \\
 &= \frac{(2-9z)(-1+6z)-(8-z+3z^2)(-9)}{(2-9z)^2} \\
 &= \frac{(-2+12z+9z-54z^2)-(-72+9z-27z^2)}{(2-9z)^2} \\
 &= \frac{-2+12z+9z-54z^2+72-9z+27z^2}{(2-9z)^2} \\
 &= \frac{\underline{-27z^2+12z+70}}{(2-9z)^2}
 \end{aligned}$$

Usar $\Delta y = f(x_2) - f(x_1) = f(x_1 + \Delta x) - f(x_1)$ para encontrar Δy usando los valores iniciales de x y Δx indicados.

1. $y = 2x^2 - 4x + 5$, $x=2$, $\Delta x=-0.2$

$$\begin{aligned}
 &= f(1.8) - f(2) \\
 &= \{2(1.8)^2 - 4(1.8) + 5\} - \{2(2)^2 - 4(2) + 5\} \\
 &= (6.48 - 7.2 + 5) - (8 - 8 + 5) \\
 &= 4.28 - 5 \\
 &= \mathbf{-0.72}
 \end{aligned}$$

2. $y = 1/x^2$, $x=3$, $\Delta x=0.3$

$$\begin{aligned}
 &= f(3.3) - f(3) \\
 &= \left\{ \frac{1}{(3.3)^2} \right\} - \left\{ \frac{1}{(3)^2} \right\} \\
 &= \frac{1}{10.89} - \frac{1}{9} \\
 &= \mathbf{-0.0192837}
 \end{aligned}$$

Derivar las funciones definidas.

1. $f(x) = (x^2 - 3x + 8)^3$

$$\begin{aligned}
 &= 3(x^2 - 3x + 8)^2 \underline{d}(x^2 - 3x + 8) \\
 &= \mathbf{3(x^2 - 3x + 8)^2 (2x - 3)}
 \end{aligned}$$

2. $g(x) = (8x-7)^{-5}$

$$\begin{aligned}
 &= -5(8x-7)^{-6} \underline{d}(8x-7) \\
 &= -5(8x-7)^{-6} (8) \\
 &= \mathbf{-40(8x-7)^{-6}}
 \end{aligned}$$

3. $f(x) = \underline{x}$

$$\begin{aligned}
 &= \frac{(x^2-1)^4}{\frac{(x^2-1)^4 \frac{d}{dx}(x) - x \frac{d}{dx}(x^2-1)^4}{((x^2-1)^4)^2}} \\
 &= \frac{((x^2-1)^4(1)) - ((x)((4(x^2-1)^3(2x))))}{((x^2-1)^4)^2} \\
 &= \frac{((x^2-1)^4(1)) - (x(8x(x^2-1)^3))}{((x^2-1)^4)^2} \\
 &= \frac{(x^2-1)^4 - 8x^2(x^2-1)^3}{(x^2-1)^8} \\
 &= \frac{-7x^2(x^2-1)^3}{(x^2-1)^8}
 \end{aligned}$$

$$\begin{aligned}
 4. \quad g(z) &= \left(z^2 - \frac{1}{z^2} \right)^6 \\
 &= 6 \left(z^2 - \frac{1}{z^2} \right)^5 \frac{d}{dx} \left(z^2 - \frac{1}{z^2} \right) \\
 &= 6 \left(z^2 - \frac{1}{z^2} \right)^5 \left(2z - \frac{((z^2)(0)-(1)(2z))}{(z^2)^2} \right) \\
 &= 6 \left(z^2 - \frac{1}{z^2} \right)^5 \left(2z + \frac{2z}{z^4} \right) \\
 &= 6 \left(z^2 - \frac{1}{z^2} \right)^5 \left(2z + \frac{2}{z^3} \right)
 \end{aligned}$$

$$\begin{aligned}
 4. \quad K(u) &= \frac{(u^2+1)^3}{(4u-5)^5} \\
 &= \left(u^2+1 \right)^3 \left(-\frac{20}{(4u-5)^6} \right) + (4u-5)^{-5} (6u(u^2+1)^2) \\
 &= \frac{-20(u^2+1)^3}{(4u-5)^6} + 6u (4u-5)^{-5} (u^2+1)^2 \\
 &= \frac{-20(u^2+1)^3 + 6u(4u-5)^{-5} (u^2+1)^2 (4u-5)^6}{(4u-5)^6} \\
 &= \frac{-20(u^2+1)^3 + 6u(4u-5) (u^2+1)^2}{(4u-5)^6} \\
 &= \frac{-20(u^2+1)^3}{(4u-5)^6} + \frac{6u (4u-5)(u^2+1)^2}{(4u-5)^6} \\
 &= \frac{-20(u^2+1)^3}{(4u-5)^6} + \frac{6u (u^2+1)^2}{(4u-5)^6} \\
 &= \frac{-20(u^2+1)^3 + 6u (u^2+1)^2 (4u-5)}{(4u-5)^6}
 \end{aligned}$$

$$\begin{aligned}
 &= \frac{(4u-5)^6}{(u^2+1)^2[-20(u^2+1)+6u(4u-5)]} \\
 &= \frac{(4u-5)^6}{(u^2+1)^2[-20u^2-20+24u^2-30u]} \\
 &= \frac{(4u-5)^6}{(u^2+1)^2[4u^2-30u-20]} \\
 &= \frac{(4u-5)^6}{(4u-5)^6}
 \end{aligned}$$

Encontrar al menos una función implícita f determinada por la ecuación dada.
 $3x-2y+4 = 2x^2+3y-7x$

$$\begin{aligned}
 -2y-3y &= 2x^2-7x-3x-4 \\
 -5y &= 2x^2-10x-4 \\
 y &= \frac{2x^2}{-5} - \frac{10x}{-5} - \frac{4}{5} \\
 y &= \frac{2x^2}{-5} + 2x + \frac{4}{-5}
 \end{aligned}$$

Derive la función definida

$$\begin{aligned}
 1. \quad f(x) &= \sqrt[3]{x^2} + 4\sqrt{x^3} \\
 &= x^{2/3} + 4x^{3/2} \\
 &- \left[\frac{2}{3} \right] X^{-1/3} + \left[\frac{3}{2} \right] (4x)^{1/2} \\
 &- \left[\frac{2}{3} \right] X^{-1/3} + \frac{12x^{1/2}}{2} \\
 &- \left[\frac{2}{3} \right] X^{-1/3} + 6x^{1/2}
 \end{aligned}$$

$$\begin{aligned}
 2. \quad k(r) &= \sqrt[3]{8r^3 + 27} \\
 &= (8r^3 + 27)^{1/3} \cdot 3(8r^2) \\
 &= \frac{1}{3} (8r^3 + 27)^{-2/3} \cdot 24r^2 \\
 &= \frac{24}{3} r^2 (8r^3 + 27)^{-2/3} \\
 &= 8r^2
 \end{aligned}$$

• Máximos y mínimos de una función de una variable (R, E, RP)

Si la función es creciente a la izquierda del punto, y de creciente a la derecha, lo llamaremos máximo. Si a la izquierda del punto la función es decreciente y a la derecha creciente, diremos que se trata de un mínimo.

Definición: Sea $f(x)$ con primera derivada y segunda derivada alrededor de un punto x_0 . Diremos que hay un máximo en x_0 , si $f'(x_0) = 0$ y si $f''(x_0) < 0$.

Definición: Sea $f(x)$ con primera y segunda derivada alrededor de un punto x_0 , se dice que en x_0 hay un mínimo si $f'(x) = 0$ y si $f''(x) > 0$.

Ejemplos:

$$1. f(x) = 3x^3 + 2x - 1 \\ f'(x) = 9x^2 + 2 \\ f''(x) = 18x$$

• Problemas que requieren el concepto de la diferencial (R, E, RP)

Definición: Sea $y = f(x)$ donde f es derivable y sea Δx un incremento de x . Entonces:

- (i) la diferencial dy de la variable dependiente y está dada por

$$dy = f'(x) \Delta x.$$
- (ii) la diferencial dx de la variable independiente x está dada por

$$dx = \Delta x.$$

Definición: Sea $w = f(x, y)$. Las diferenciales dx y dy de las variables independientes x y y se definen como

$$dx = \Delta x \text{ y } dy = \Delta y,$$

donde Δx y Δy son incrementos de x y y . La diferencial dw de la variable dependiente w se define por medio de

$$dw = f_x(x, y) dx + f_y(x, y) dy = \frac{\partial w}{\partial x} dx + \frac{\partial w}{\partial y} dy$$

Definición: Sea $w = f(x, y)$. Decimos que f es diferenciable o que tiene una diferencial en (x_0, y_0) si Δw se puede expresar en la forma

$$\Delta w = f_x(x_0, y_0) \Delta x + f_y(x_0, y_0) \Delta y + \eta_1 \Delta x + \eta_2 \Delta y$$

donde η_1 y η_2 tienden a cero cuando $(\Delta x, \Delta y) \rightarrow (0, 0)$.

Use diferenciales para estimar el cambio en $f(x, y, z) = x^2z^3 - 3yz^2 + x^{-3} + 2y^{1/2}z$ cuando (x, y, z) cambia de

(1, 4, 2) a (1.02, 3.97, 1.96).

$$\begin{aligned}
 &= (2xz^3 + (-3x^{-4})) dx + (-3z^2 + y^{-1/2}z) dy + (x^23z^2 + (-3y2z)+ 2y^{1/2}) dz \\
 &\quad x_1 = 1 \qquad \qquad \qquad y_1 = 4 \qquad \qquad \qquad z_1 = 2 \\
 &\quad x_2 = 1.02 \qquad \qquad \qquad y_2 = 3.97 \qquad \qquad \qquad z_2 = 1.96 \\
 &\quad \Delta x = 1.02 - 1 \qquad \qquad \qquad \Delta y = 3.97 - 4 \qquad \qquad \qquad \Delta z = 1.96 - 2 \\
 &\quad dx = \Delta x = 0.02 \qquad \qquad \qquad dy = \Delta y = -0.03 \qquad \qquad \qquad dz = \Delta z = -0.04 \\
 \\
 &= (2(1)(2)^3 + (-3(1)^{-4})) (0.02) + (-3(2)^2 + 4^{-1/2} (2)) (-0.03) + ((1)^2 3(2)^2 + (-3 (4)(2(2)) + 2 (4)^{1/2}) (-0.04) \\
 &= (2(8) + (-3)) (0.02) + (-3(4) + (.5)(2)) (-0.03) + ((1)(12) + (-12)(4) + 2(2)) (-0.04) \\
 &= (16 - 3)(0.02) + (-12 + 1) (-0.03) + (12 + (-48) + 4) (-0.04) \\
 &= (13)(0.02) + (-11)(-0.03) + (-32) (-0.04) \\
 &= 0.26 + 0.33 + 1.28
 \end{aligned}$$

$$= 1.87$$

Use diferenciales para estimar el cambio en $f(x, y) = x^2 - 3x^3y^2 + 4x - 2y^3 + 6$ cuando (x, y) cambia de $(-2, 3)$ a

(-2.02, 3.01).

$$= (2x + 9x^2y^2 + 4) dx + (-3x^3 - 2y - 6y^2) dy$$

$$= (2x + 9x^2y^2 + 4) dx - (6x^3y - 6y^2) dy$$

$$x_1 = -2$$

$$y_1 = 3$$

$$x_2 = -2.02$$

$$y_2 = 3.01$$

$$\Delta x = -2.02 - (-2)$$

$$\Delta y = 3.01 - 3$$

$$\Delta x = -2.02 + 2$$

$$\Delta y = 0.01$$

$$dx = \Delta x = -0.02$$

$$dy = \Delta y = 0.01$$

$$= (2(-2) + 9(-2)^2(3)^2 + 4)(-0.02) + (-6(-2)^3(3) - 6(3)^2)(0.01)$$

$$= (-4 - 324 + 4)(-0.02) + (144 - 54)(0.01)$$

$$= (-324)(-0.02) + (90)(0.01)$$

$$= 6.48 + 0.9$$

$$= 7.38$$

Encuentre dw.

$$1. w = x^3 - x^2y + 3y^2$$

$$dw = \frac{\partial w}{\partial x} dx + \frac{\partial w}{\partial y} dy$$

$$= (3x^2 - 2xy) dx + (x^2 + 6y) dy$$

$$2. w = x^2 \operatorname{sen} y + 2y^{3/2}$$

$$dw = \frac{\partial w}{\partial x} dx + \frac{\partial w}{\partial y} dy$$

$$= (2x \operatorname{sen} y) dx + x^2 \frac{\partial}{\partial y} (\operatorname{sen} y) + \operatorname{sen} y \frac{\partial}{\partial y} (x^2) + 2 \left[\frac{3}{2} y^{-1/2} \right]$$

$$= (2x \operatorname{sen} y) dx + (x^2 \cos y + 3y^{1/2}) dy$$

$$3. w = x^2 e^{xy} + (1/y^2)$$

$$= x^2 \frac{\partial}{\partial x} (e^{xy}) + e^{xy} \frac{\partial}{\partial x} (x^2) + \frac{\partial}{\partial x} (1/y^2) + x^2 \frac{\partial}{\partial y} (e^{xy}) + e^{xy} \frac{\partial}{\partial y} (x^2) +$$

$$\frac{\partial}{\partial y} (1/y^2)$$

$$= x^2(xe^{xy}) + e^{xy}(2x) + 0 + x^2(ye^{xy}) + e^{xy}(0) + y^2 \frac{\frac{\partial}{\partial y} (-1)}{y^4} \frac{\frac{\partial}{\partial y} (y^2)}{y^4}$$

$$= x^3 e^{xy} + 2x e^{xy} + x^2 y e^{xy} + \left[\frac{-2y}{y^4} \right]$$

$$= x^3 e^{xy} + 2x e^{xy} + x^2 y e^{xy} - \frac{2}{y^3}$$

$$= x^3 e^{xy} + 2x e^{xy} + x^2 y e^{xy} - 2y^{-3}$$

$$= e^{xy} (x^2 y + 2x) dx + (x^3 e^{xy} - 2y^{-3}) dy$$

4. $w = x^2 \ln(y^2 + z^2)$

$$\begin{aligned} &= x^2 \frac{\partial}{\partial x} \ln(y^2 + z^2) + \ln(y^2 + z^2) \frac{\partial}{\partial x} (x^2) + x^2 \frac{\partial}{\partial y} \ln(y^2 + z^2) + \ln(y^2 + z^2) \frac{\partial}{\partial y} (x^2) \\ &\quad + x^2 \frac{\partial}{\partial z} \ln(y^2 + z^2) + \ln(y^2 + z^2) \frac{\partial}{\partial z} (x^2) \\ &= \ln(y^2 + z^2)(2x) + x^2 \left[\ln \frac{1}{y^2 + z^2} \right] \left[\frac{\partial}{\partial y} y^2 + z^2 \right] + x^2 \left[\ln \frac{1}{y^2 + z^2} \right] \left[\frac{\partial}{\partial z} y^2 + z^2 \right] \\ &= 2x \ln(y^2 + z^2) + x^2 \left[\frac{1}{y^2 + z^2} \right] (2y) dy + x^2 \left[\frac{1}{y^2 + z^2} \right] (2z) dz \\ &= 2x \ln(y^2 + z^2) dx + x^2 \left[\frac{2y}{y^2 + z^2} \right] dy + x^2 \left[\frac{2z}{y^2 + z^2} \right] dz \\ &= 2x \ln(y^2 + z^2) dx + \frac{2x^2 y}{y^2 + z^2} dy + \frac{2x^2 z}{y^2 + z^2} dz \end{aligned}$$

2. Cálculo integral en una variable

• Problemas geométricos aplicando el concepto de integral (R, E, RP)

Definición: Sea f una función definida en un intervalo cerrado $[a, b]$. La integral definida de f desde a hasta b denotada por $\delta_a^b f(x) dx$, está dada por:

$$\delta_a^b f(x) dx = \lim_{\|P\| \rightarrow 0} \sum_i f(w_i) \Delta x_i$$

siempre que el límite exista.

Ejemplos:

1. δ_{-3}^2

$$= \delta_{-3}^2 x dx = 2x + 6 \Big|_{-3}^2$$

$$= \delta_{-3}^2 2x dx + \delta_{-3}^2 6 dx = 2\delta_{-3}^2 x dx + 6\delta_{-3}^2 dx$$

$$\begin{aligned} &= 2 \left[\frac{x^{1+1}}{1+1} \Big|_{-3}^2 \right] + 6x \Big|_{-3}^2 \\ &= 2 \left[\frac{x^2}{2} \Big|_{-3}^2 \right] + 6x \Big|_{-3}^2 \end{aligned}$$

$$\begin{aligned}
 & 2 \quad -3 \quad -3 \\
 & = x^2 \left| \begin{array}{c} 2 \\ -3 \end{array} \right. + 6x \left| \begin{array}{c} 2 \\ -3 \end{array} \right. \\
 & = \{(2)^2 - (-3)^2\} + \{(6)2 - 6(-3)\} \\
 & = 4 - 9 + 12 + 18 = 25 \\
 2. \quad & \int_0^3 \sqrt{9-x^2} dx \\
 & = \frac{1}{2} x \sqrt{9-x^2} + \frac{1}{2} \cdot 9 \arcsen^* \frac{x}{3} \\
 & = \frac{1}{2} x \sqrt{9-x^2} + \frac{9}{2} \arcsen \frac{x}{3} \\
 & = \frac{1}{2} (3) \sqrt{9-x^2} + \frac{9}{2} \arcsen \frac{3}{3} - \frac{1}{2} (0) \sqrt{9-(0)^2} + \frac{9}{2} \arcsen \frac{0}{3} \\
 & = \frac{9}{2} (90) \\
 & = \frac{9}{2} \left(\frac{\pi}{2} \right) \\
 & = \frac{9\pi}{4}
 \end{aligned}$$

*NOTA: Ejemplo
 $\arcsen^{-1} = \arcsen$
 $\arcsen \theta = 10$
 $\theta = \arcsen^{-1}(10)$
 $\theta = \arcsen(10)$

- Problemas físicos aplicando el concepto de integral (R, E, RP)

3. Cálculo de varias variables

- Gradiente de una función (R, E, RP)

Definición: Si f es una función de dos variables, entonces el gradiente de f se define como

$$\nabla f(x, y) = \langle f_x(x, y), f_y(x, y) \rangle = f_x(x, y)\mathbf{i} + f_y(x, y)\mathbf{j}.$$

Derivada Direccional de una Función:

Definición: Si f es una función de x y y y $\mathbf{u} = \langle \cos\theta, \sin\theta \rangle$ es un vector unitario, entonces la derivada direccional de f en la dirección de \mathbf{u} , denotada por $D_{\mathbf{u}}f(x, y)$, está dada por

$$D_{\mathbf{u}}f(x, y) = \lim_{t \rightarrow 0} \frac{f(x+t \cos\theta, y+t \sin\theta) - f(x, y)}{t}$$

Derivadas de Funciones Vectoriales:

Definición: Una función vectorial r es continua en a si $\lim_{t \rightarrow a} r(t) = r(a)$.

Definición: Si r es una función vectorial, entonces la derivada de r es la función vectorial r' definida por

$$r'(t) = \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} [r(t + \Delta t) - r(t)]$$

para todo t para el cual el límite existe.

- Divergencia y rotacional (R, E, RP)
- Derivada direccional de una función (R, E, RP)
- Máximos, mínimos y puntos silla de funciones de dos o más variables (R, E, RP)
- Derivadas de funciones vectoriales (R, E, RP)
- Integrales de funciones vectoriales (R, E, RP)
- Integrales múltiples (R, E, RP)
- Integrales de línea (R, E, RP)

4. Ecuaciones diferenciales

- Problemas de aplicación (R, E, RP)

Clasificación:

La primera clasificación de las ecuaciones diferenciales se hace utilizando el orden de la ecuación.

El orden de una ecuación diferencial es el de la derivada de mayor orden de la ecuación diferencial.

Ecuaciones de Primer Orden Lineales:

✓ De Variables Separadas:

Son de la forma $P(x) dx + Q(y)dy = 0$

Son las más sencillas de integrar. Sólo tenemos que pasar al otro lado del signo igual uno de los sumandos e integrar en los dos lados.

$$\begin{aligned} xdx + 2y^2dy &= 0 \\ xdx &= -2y^2dy \end{aligned}$$

Integrando en los dos lados, nos queda:

$$x^2/2 = -2/3y^3 + C$$

✓ Ecuaciones Homogéneas:

Son aquellas en las que y' es una función homogénea de grado cero de x e y (es decir, el grado de todos los términos es el mismo).

$$(x^2 - y^2)dx + 2xydy = 0.$$

Dividiendo por x^2 nos queda

$$(1 - y^2/x^2)dx + 2y/x dy = 0.$$

Haciendo el cambio $y/x = u$ y derivando ($y = ux$) nos queda
 $y' = u + u'x(1 - u^2)dx + 2udy = 0$ $1 - u^2 + 2u(u + u'x) = 0$

Operando nos queda

$$u + u'x = (u^2 - 1)/(2u)x \ du + (1 + u^2)/(2u)dx = 0$$

Esta ecuación diferencial es del tipo de variables separadas.

✓ **Ecuaciones Reducibles a Homogéneas:**

Son aquellas que mediante un cambio de variable se convierten en homogéneas.

✓ **Diferenciales Exactas:**

Dada la ecuación diferencial $P(x,y)dx + Q(x,y)dy = 0$. Si se cumple $P'_y = Q'_x$ la ecuación es una diferencial exacta.

✓ **Reducibles a Diferencial Exacta:**

Se convierten a diferencial exacta haciendo una transformación.

✓ **Ecuación Lineal:**

Son las ecuaciones de la forma $y' + X(x)y = F(x)$

✓ **Ecuación de Bernoulli:**

Son las ecuaciones de la forma $y' + X(x)y = F(x)y^n$

✓ **Ecuación de Riccati:**

Son del tipo $y' = X_1(x) + X_2(x)y + X_3(x)y^2$

Ecuaciones de Primer Orden No Lineales:

✓ **Resolvibles en y' :**

Son de la forma:

$$a_0(x,y)y^n + a_1(x,y)y^{n-1} + a_2(x,y)y^{n-2} + \dots + a_n(x,y) = 0$$

✓ **Resolvibles en x :**

Cuando se puede despejar la x . Obtenemos $x = f(y,y')$ y derivando respecto de x

$$1 = f'_y y' + f''_{yy} y''$$

haciendo el cambio

$$y' = p \text{ e } y'' = dp/dx = dp/dy dy/dx = dp/dy p$$

tenemos una ecuación de primer orden lineal.

✓ **Resolvibles en y :**

Cuando se puede despejar y . Obtenemos $y = f(x,y')$. Hacemos $y' = p$. Entonces

$$\begin{aligned} y'' &= p' \\ y' &= f'_x + f'_y y'' \\ p &= f'_x + f'_y p' \end{aligned}$$

Que es una ecuación de primer orden y primer grado que puede ser más fácil de integrar que la original.

✓ **Ecuación de Lagrange:**

Son de la forma

$$y = x f(y') + g(y')$$

donde $f(y')$ no puede ser igual y' .

Se resuelven derivando y llamando $y' = p$ con lo que obtenemos
 $p = f(p) + [x f'(p) + g'(p)]p'$
esta ecuación es lineal y se integra tomando x como función de p.

✓ **Ecuación de Clairaut:**

Es como la de Lagrange pero con $f(y') = y'$
 $y = x y' + g(y')$

La solución es

$$y = Cx + g(C)$$

✓ **Reducción de Ecuaciones Carentes de Términos en y:**

Son del tipo $F(x, y', y'', \dots y^{(n)}) = 0$

Se resuelven reduciéndolas a otras de orden $n - 1$ haciendo el cambio $y' = p$

✓ **Ecuaciones sin x:**

Son del tipo $F(y, y', y'', \dots y^{(n)}) = 0$

✓ **Ecuaciones Carentes de x e y:**

Son del tipo $F(y', y'', \dots y^{(n)}) = 0$

✓ **Ecuaciones del Tipo $y^{(n)} = F(y^{(n-2)})$:**

Se reducen a otras de segundo orden haciendo el cambio $z = y^{(n-2)}$ y quedan de la forma $z'' = F(z)$. Si multiplicamos por $z' dx$ nos queda $z'' z' dx = F(z) dz$ cuyo primer miembro es la diferencial de $z'^2/2$ por lo que integrando obtendremos una ecuación de primer orden de variables separadas.

5. Series de Fourier

• Propiedades de ortogonalidad (R, E, RP)

Sea una función $f(t)$ una función periódica de periodo T , la cual se puede representar por la serie trigonométrica

$$\begin{aligned} f(t) &= \frac{1}{2}a_0 + a_1 \cos \omega_0 t + a_2 \cos 2\omega_0 t + \dots + b_1 \sin \omega_0 t + b_2 \sin 2\omega_0 t + \dots \\ &= \frac{1}{2}a_0 + \sum_{n=1}^{\infty} (a_n \cos n\omega_0 t + b_n \sin n\omega_0 t) \end{aligned}$$

donde $\omega_0 = 2\pi/T$.

Una serie como la representada se llama serie trigonométrica de Fourier. Esta serie también se puede representar así:

$$f(t) = C_0 + \sum_{n=1}^{\infty} C_n \cos(n\omega_0 t - \theta_n)$$

Ejemplo 1: Deducir la forma

$$f(t) = C_0 + \sum_{n=1}^{\infty} C_n \cos(n\omega_0 t - \theta_n) \quad \text{de} \quad \frac{1}{2}a_0 + \sum_{n=1}^{\infty} (a_n \cos n\omega_0 t + b_n \sin n\omega_0 t)$$

y expresar C_n y θ_n en términos de a_n y b_n . Se puede expresar así

$$a_n \cos n\omega_0 t + b_n \sin n\omega_0 t = \sqrt{a_n^2 + b_n^2} \left(\frac{a_n}{\sqrt{a_n^2 + b_n^2}} \cos n\omega_0 t + \frac{b_n}{\sqrt{a_n^2 + b_n^2}} \right)$$

se utiliza la entidad trigonométrica

$$\begin{aligned} a_n \cos n\omega_0 t + b_n \sin n\omega_0 t &= C_n (\cos \theta_n \cos n\omega_0 t + \sin \theta_n \sin n\omega_0 t) \\ &= C_n \cos(n\omega_0 t - \theta_n) \end{aligned}$$

donde

$$\begin{aligned} C_n &= \sqrt{a_n^2 + b_n^2} \\ \cos \theta_n &= \frac{a_n}{\sqrt{a_n^2 + b_n^2}} \quad \sin \theta_n = \frac{b_n}{\sqrt{a_n^2 + b_n^2}} \end{aligned}$$

por consiguiente,

$$\tan \theta_n = \frac{b_n}{a_n} \quad \text{ó} \quad \theta_n = \tan^{-1} \left(\frac{b_n}{a_n} \right)$$

También si se hace

$$C_n = \frac{1}{2} a_0$$

Se Obtiene

$$f(t) = \frac{1}{2} a_0 + \sum_{n=1}^{\infty} (a_n \cos n\omega_0 t + b_n \sin n\omega_0 t) = C_0 + \sum_{n=1}^{\infty} C_n \cos(n\omega_0 t - \theta_n)$$

Es obvio que la representación de Fourier de una función periódica, representa la función como la suma de componentes sinusoidales que tienen diferentes frecuencias. La componente sinusoidales de frecuencia $\omega_n = n\omega_0$ se denomina la enésima armónica de la función periódica. La primera armónica comúnmente se conoce como la componente fundamental porque tiene el mismo período de la función y $\omega_0 = 2\pi f_0 = 2\pi/T$ se conoce como la frecuencia angular fundamental. Los coeficientes C_n y los ángulos θ_n se conocen como amplitudes armónicas y ángulos de fase, respectivamente.

Propiedades de Ortogonalidad:

Existen muchos tipos de series de Fourier. Las mas sencillas son las Series de Fourier Trigonométricas. Un ejemplo es la serie de Fourier del seno.

$$\sum_{n=1}^{\infty} \frac{1}{2} \sin nx = \sin x + \frac{1}{2} \sin 2x + \frac{1}{3} \sin 3x + \dots$$

Se vera que las series de Fourier tienen interpretaciones físicas importantes en las aplicaciones. Sin embargo, las series de Fourier están basadas en un tipo distinto de teoría a las familiares series de potencias.

De manera equivalente, una función diferenciable $f(x)$ es una función tal que en cualquier intervalo finito se puede dividir en un número de partes, cada una de las cuales es continua y tiene derivada continua. Además, las únicas discontinuidades de $F(x)$ y $f'(x)$ son discontinuidades de salto.

Ejemplo Funciones Ortogonales

Las dos funciones $f(x) = x$ y $g(x) = x^2$ son ortogonales en el intervalo $[-1,1]$ puesto que

$$\begin{aligned} (x, x^2) &= \int_{-1}^1 x x^2 dx \\ &= \int_{-1}^1 x^3 dx \\ &= \frac{x^4}{4} \Big|_{-1}^1 = \frac{1}{4} - \frac{1}{4} = 0 \end{aligned}$$

Conjunto Ortogonal de Funciones:

Un conjunto de funciones $\{\phi_1(x), \phi_2(x), \dots\}$ es un conjunto ortogonal de funciones en el intervalo $[a,b]$ si cualesquiera dos funciones son ortogonales entre si.

$$(\phi_n, \phi_m) = \int_a^b \phi_n(x) \phi_m(x) dx = 0 \quad (n \neq m).$$

se considerará solo conjuntos ortogonales en los que ninguna de las funciones son idénticamente iguales a cero en $[a,b]$.

Los coeficientes de una serie respecto a un conjunto ortogonal tiene una forma útil, que se deducirá ahora. Suponga que $\{\phi_1(x), \phi_2(x), \dots\}$ es un conjunto ortogonal de funciones en el intervalo $[a,b]$ y que

$$F(x) = \sum_{n=1}^{\infty} C_n \phi_n(x)$$

Se quiere obtener una formula para los coeficientes C_n en términos de $f(x)$ y de las funciones ortogonales $\phi_n(x)$. Se selecciona un miembro del conjunto ortogonal, digamos, $\phi_n(x)$, y tome el

$$F(x) = \sum_{n=1}^{\infty} C_n \phi_n(x)$$

producto interno con $f(x)$. es decir, se multiplican ambos lados de por $\phi_n(x)$, y se integra sobre el intervalo para obtener

$$\begin{aligned} (f, \phi_m) &= \int_a^b f(x) \phi_m(x) dx \\ &= \int_a^b \left[\sum_{n=1}^{\infty} C_n \phi_n(x) \right] \phi_m(x) dx \end{aligned}$$

suponga que la integración y la suma se puede intercambiar para dar

$$(f, \phi_m) = \sum_{n=1}^{\infty} C_n \int_a^b \phi_n(x) \phi_m(x) dx = \sum_{n=1}^{\infty} C_n (\phi_n, \phi_m)$$

Pero ϕ_n forma un conjunto ortogonal, de manera que $(\phi_n, \phi_m) = 0$ si $n \neq m$. Entonces se convierte en

$$(f, \phi_m) = C_m (\phi_m, \phi_m)$$

Teorema Fundamental de una Función por una Serie de Funciones Ortogonales:

Suponga que $f(x)$ es diferenciable por partes en el intervalo $[a,b]$ y que

$$f(x) = \sum_{n=1}^{\infty} C_n \phi_n(x)$$

donde $\{\phi_n(x)\}$ es un conjunto ortogonal de funciones en $[a,b]$. Entonces

$$C_n = \frac{(f, \phi_n)}{(\phi_n, \phi_n)} = \frac{\int_a^b f(x) \phi_n(x) dx}{\int_a^b \phi_n^2(x) dx}$$

Una prueba rigurosa del teorema incluye consideraciones técnicas que están más allá del nivel

$$F(x) = \sum_{n=1}^{\infty} C_n \phi_n(x)$$

de esta investigación. Estas consideraciones se refieren a la convergencia de y a la demostración de que la suma y la integral se pueden intercambiar. Además cuando se

$$F(x) = \sum_{n=1}^{\infty} C_n \phi_n(x)$$

escribe , de hecho, no se requiere que la serie converja a $f(x)$ para toda x . Las condiciones suficientes para garantizar el intercambio de Fourier del seno y del coseno,

$$F(x) = \sum_{n=1}^{\infty} C_n \phi_n(x)$$

también se analizan en que sentido es igual a $f(x)$. Sólo se necesita la continuidad por las partes de f y las ϕ_n para este teorema.

• Condiciones de Dirichlet y propiedades de convergencia (R, E, RP)

Es importante establecer criterios simples para determinar cuando una serie de Fourier converge. En esta sección desarrollaremos las condiciones que debe cumplir $f(x)$ para que se pueda encontrar la suma de la serie de Fourier. Un método bastante útil para analizar las propiedades de convergencia es expresar las sumas parciales de Fourier como integrales. Riemann y Fejer establecieron otros métodos para el cálculo de las series de Fourier. En esta sección nos limitaremos al estudio de la convergencia de las funciones que son continuas a trozos en un intervalo dado.

Función Continua a Trozos:

Una función f es **continua a trozos** dentro de un intervalo si ambas f y f' son continuas a trozos en dicho intervalo.

Supongamos que f es continua a trozos y periódica. Entonces la serie (1) con coeficientes (2) converge a :

1. $f(x)$ si x es un punto de continuidad.

2.

$\frac{1}{2}(f(x+0) + f(x-0))$
si x es un punto de discontinuidad.

Esto significa que, en cualquier x dentro del intervalo $-L$ y L , la serie de Fourier converge al promedio entre el límite izquierdo y el derecho de $f(x)$ en x . Si f es continua en x , entonces los límites izquierdo y derecho son iguales a $f(x)$, y la serie de Fourier converge a la misma $f(x)$. Si f tiene una discontinuidad en x entonces la serie de Fourier converge al punto medio en la discontinuidad.

Sea f una función continua a trozos. Se dice que f está *estandarizada* si los valores en los x_i de discontinuidad estan dados por

$$f(x_i) = \frac{1}{2}[f(x_i+) + f(x_i-)]$$

Las condiciones impuestas a $f(x)$ son suficientes mas no necesarias, i.e. si las condiciones se satisfacen la convergencia esta garantizada. Sin embargo si no se satisfacen la serie puede o no converger.

Desigualdad de Bessel:

Sea f una función integrable en el intervalo $[0, T]$. Sean a_n, b_n, c_n los coeficientes de Fourier de f . Entonces:

$$\frac{|a_0|^2}{2} + \sum_{n=1}^{\infty} (|a_n|^2 + |b_n|^2) = 2 \sum_{k=-\infty}^{\infty} |c_k|^2 \leq \frac{2}{T} \int_0^T |f(t)|^2 dt$$

Lema de Riemann:

Sea f integrable y a_n y b_n los coeficientes de Fourier de f . Entonces:

$$\lim_{n \rightarrow \infty} a_n = \lim_{n \rightarrow \infty} b_n = 0$$

lo que implica que

$$\lim_{n \rightarrow \infty} \int_{-\pi}^{\pi} f(t) \cos nt dt = \lim_{n \rightarrow \infty} \int_{-\pi}^{\pi} f(t) \sin nt dt = 0$$

Identidad de Parseval:

$$\frac{2}{T} \int_0^T |f(x)|^2 dx = \frac{a_0^2}{2} + \sum_{n=1}^{\infty} (a_n^2 + b_n^2)$$

Si a_n y b_n son coeficientes de Fourier correspondientes a $f(x)$ y si $f(x)$ satisface las condiciones Dirichlet.

- **Integración y diferenciación de las series de Fourier (R, E, RP)**

6. Transformadas de Fourier

- **Integral de Fourier (R, E, RP)**

Una serie de Fourier puede usarse algunas veces para representar una función dentro un intervalo. Si una función esta definida sobre toda la recta real, puede ser representarse con una serie Fourier si es periódica. Si no es periódica, entonces no puede representarse con una serie Fourier para todo x . Aun en este caso es posible representar la función en términos de senos y cosenos, pero la serie de Fourier se convierte en una integral de Fourier. La motivación proviene de considerar formalmente las series de Fourier como funciones con período $2T$ y hacer tender T al infinito. Suponiendo:

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{inx}$$

y

$$c_n = \frac{1}{2T} \int_{-T}^T e^{-in\frac{\pi}{T}t} f(t) dt$$

Tomando

$$\omega_n = \frac{n\pi}{T} \quad \text{and} \quad \Delta\omega = \omega_n - \omega_{n-1} = \frac{\pi}{T}$$

y reemplazando la fórmula de la integral por los coeficientes de Fourier:

$$\frac{1}{2\pi} \sum_{n=-\infty}^{\infty} \left(e^{i\omega_n x} \int_{-T}^T e^{-i\omega_n t} f(t) dt \right) \Delta\omega$$

La sumatoria se asemeja a una suma de Riemann de una integral definida, y en el límite $T \rightarrow \infty$ ($\Delta\omega \rightarrow 0$) tendríamos:

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} \left(e^{i\omega x} \int_{-T}^T e^{-i\omega t} f(t) dt \right) d\omega \quad x \in \mathbb{R}$$

Este razonamiento informal sugiere la siguiente definición:

Transformadas de Fourier:

Una función $F(\omega)$ se denomina la **transformada de Fourier** de $f(x)$, si:

$$F(\omega) := \mathcal{F}\{f(t)\} := \int_{-\infty}^{\infty} e^{-i\omega t} f(t) dt$$

existe.

$$\mathcal{F}^{-1}\{F(\omega)\} := \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\omega x} F(\omega) d\omega$$

$$F(\omega)$$

Se denomina la **transformada inversa de Fourier** de $\mathcal{F}\{f(t)\}$. La transformada de Fourier de f es por lo tanto una función $F(\omega)$ de una nueva variable ω . Esta función, evaluada en ω , es

Las constantes $1/2\pi$ que preceden la integral pueden ser reemplazadas con cualesquier dos constantes cuyo producto sea $1/2\pi$.

Ejemplo:

La transformada de Fourier de f definida como:

$$f(t) := \begin{cases} 1 & |t| < 1 \\ 0 & |t| > 1 \end{cases}$$

es

$$\mathcal{F}\{f(t)\} = \int_{-1}^1 e^{-i\omega t} dt = \begin{cases} 2 \frac{\sin \omega}{\omega} & \omega \neq 0 \\ 2 & \omega = 0 \end{cases}$$

La transformada inversa de Fourier se calcula así:

$$\begin{aligned} \frac{1}{\pi} \int_{-\infty}^{\infty} e^{i\omega x} \frac{\sin \omega}{\omega} d\omega &= \frac{2}{\pi} \int_0^{\infty} \cos \omega x \frac{\sin \omega}{\omega} d\omega \\ &= \frac{1}{\pi} \int_0^{\infty} \frac{\sin \omega(x+1)}{\omega} d\omega - \frac{1}{\pi} \int_0^{\infty} \frac{\sin \omega(x-1)}{\omega} d\omega \\ &= \begin{cases} 1 & |x| < 1 \\ 1/2 & |x| = 1 \\ 0 & |x| > 1 \end{cases} \end{aligned}$$

La integral de Fourier:

1. Si $f(x)$ y $f'(x)$ son funciones definidas continuas a trozos en un intervalo finito .

y

2.

$$\int_{-\infty}^{\infty} |f(x)| dx \quad (-\infty, \infty)$$

converge, i.e. $f(x)$ es completamente integrable en

Entonces:

$$\frac{1}{2}[f(x-) + f(x+)] = \frac{1}{2\pi} \lim_{\Omega \rightarrow \infty} \int_{-\Omega}^{\Omega} e^{i\omega x} \left(\int_{-\infty}^{\infty} e^{-i\omega t} f(t) dt \right) d\omega$$

Las condiciones arriba citadas son suficientes pero no necesarias. La similitud con los resultados obtenidos para la serie de Fourier es aparente.

• Propiedades de las transformadas de Fourier (R, E, RP)

Linealidad:

$$\alpha, \beta \in \mathbb{C}$$

Si , entonces:

$$\mathcal{F}\{\alpha f(t) + \beta g(t)\} = \alpha \mathcal{F}\{f(t)\} + \beta \mathcal{F}\{g(t)\} = \alpha F(\omega) + \beta G(\omega)$$

Dado que la transformada de Fourier de $f(t)$ y $g(t)$ existe.

Escalado:

$$\mathcal{F}\{f(t)\} = F(\omega) \quad c \in \mathbb{R}$$

Si y , entonces:

$$\mathcal{F}\{ct\} = \frac{1}{|c|} F\left(\frac{\omega}{c}\right), \quad c \neq 0$$

Corrimiento en Tiempo:

$$\mathcal{F}\{f(t)\} = F(\omega) \quad t_0 \in \mathbb{R}$$

Si y , entonces:

$$\mathcal{F}\{f(t-t_0)\} = e^{-i\omega t_0} F(\omega), \quad t_0 \in \mathbb{R}$$

Corrimiento en frecuencia:

Si $\mathcal{F}\{f(t)\} = F(\omega)$ y $\omega_0 \in \mathbb{R}$, entonces:

$$\mathcal{F}\{e^{i\omega_0 t} f(t)\} = F(\omega - \omega_0), \quad \omega_0 \in \mathbb{R}$$

Simetría:

Si $\mathcal{F}\{f(t)\} = F(\omega)$, entonces:

$$\mathcal{F}\{F(t)\} = 2\pi f(-\omega)$$

Usando la formula de la transformada inversa de Fourier

$$f(t) = \mathcal{F}^{-1}\{F(\omega)\} = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(x) e^{ixt} dx$$

por lo tanto:

$$2\pi f(-\omega) = \int_{-\infty}^{\infty} F(x) e^{-ix\omega} dx = \int_{-\infty}^{\infty} F(t) e^{-it\omega} dt = \mathcal{F}\{F(t)\}$$

Modulación:

Si $\mathcal{F}\{f(t)\} = F(\omega)$ y $\omega_0 \in \mathbb{R}$, entonces:

$$\begin{aligned} \mathcal{F}\{f(t) \cos(\omega_0 t)\} &= \frac{1}{2}[F(\omega + \omega_0) + F(\omega - \omega_0)] \\ \mathcal{F}\{f(t) \sin(\omega_0 t)\} &= \frac{1}{2i}[F(\omega + \omega_0) - F(\omega - \omega_0)] \end{aligned}$$

Derivada en Tiempo:

$n \in \mathbb{N}$

Sea y asumiendo que $f^{(n)}$ es continua a trozos. Suponiendo

$\lim_{t \rightarrow \infty} f^{(k)}(t) = \lim_{t \rightarrow -\infty} f^{(k)}(t) = 0$ que . Entonces:

$$\mathcal{F}\{f^{(n)}(t)\} = (i\omega)^n F(\omega)$$

En particular:

$$\mathcal{F}\{f'(t)\} = i\omega F(\omega)$$

y

$$\mathcal{F}\{f''(t)\} = -\omega^2 F(\omega)$$

Derivada en Frecuencia:

$n \in \mathbb{N}$

Dado y asumiendo que f sea continua a trozos. Entonces:

$$\mathcal{F}\{t^n f(t)\} = i^n F^{(n)}(\omega)$$

En particular:

$$\mathcal{F}\{tf(t)\} = iF'(\omega)$$

y

$$\mathcal{F}\{t^2 f(t)\} = -F''(\omega)$$

Probaremos el teorema para $n=1$. El argumento para un n más grande es una repetición de este.

$$F'(\omega) = \int_{-\infty}^{\infty} [f(t)e^{-i\omega t}]dt = -i \int_{-\infty}^{\infty} (tf(t))e^{-i\omega t}dt = -i\mathcal{F}\{tf(t)\}$$

- **Transformadas de Fourier de derivadas (R, E, RP)**

- **Teorema de convolución (R, E, RP)**

Si f y g ambas tienen transformadas de Fourier, entonces la convolución $f*g$ de las funciones f y g se define como:

$$f * g = \int_{-\infty}^{\infty} f(u)g(x-u)du$$

La transformada de Fourier de la convolución de $f(x)$ y $g(x)$ es igual al producto de las transformadas de Fourier de $f(x)$ y $g(x)$.

$$\mathcal{F}\{f * g\} = \mathcal{F}\{f\}\mathcal{F}\{g\} \text{ (convolución en tiempo)}$$

y

$$\mathcal{F}\{f(t)g(t)\} = \frac{1}{2\pi}[F * G](\omega) \text{ (convolución en frecuencia)}$$

II. Álgebra

1. Clásica

- **Funciones, tipos y propiedades (R, E, RP)**

Álgebra, rama de las matemáticas en la que se usan letras para representar relaciones aritméticas. Al igual que en la aritmética, las operaciones fundamentales del álgebra son adición, sustracción, multiplicación, división y cálculo de raíces. La aritmética, sin embargo, no es capaz de generalizar las relaciones matemáticas, como el teorema de Pitágoras, que dice que en un triángulo rectángulo el área del cuadrado que tiene como lado la hipotenusa es igual a la suma de las áreas de los cuadrados cuyos lados son los catetos. La aritmética sólo da casos particulares de esta relación (por ejemplo, 3, 4 y 5, ya que $3^2 + 4^2 = 5^2$). El álgebra, por el contrario, puede dar una generalización que cumple las condiciones del teorema: $a^2 + b^2 = c^2$.

El álgebra clásica, que se ocupa de resolver ecuaciones, utiliza símbolos en vez de números específicos y operaciones aritméticas para determinar cómo usar dichos símbolos. El álgebra moderna ha evolucionado desde el álgebra clásica al poner más atención en las estructuras matemáticas. Los matemáticos consideran al álgebra moderna como un conjunto de objetos con reglas que los conectan o relacionan. Así, en su forma más general, se dice que el álgebra es el idioma de las matemáticas.

Función:

Es una relación que asigna a cada elemento del dominio uno y sólo un elemento del contradominio. Este último se llama el valor de la función para el elemento dado del dominio.

Una función f de A en B , se escribe $f: A \rightarrow B$ es la relación de A en B .

Propiedades:

- (a) $\text{Dom}(f) = A$
- (b) Si (a, b) y (a, c) pertenecen a f , entonces $b = c$.

La propiedad (b) dice que, si $(a, b) \in f$, entonces b está determinada únicamente por a . Por esta razón, también se escribe $b = f(a)$ y se enlista la relación f como $\{(a, f(a)) | a \in A\}$. Las funciones son también llamadas **aplicaciones** o **transformaciones**.

Ejemplos:

1. Sean $A = \{1, 2, 3, 4\}$ y $B = \{a, b, c, d\}$ y sea $f = \{(1, a), (2, a), (3, d), (4, c)\}$ **

Entonces f es una función, ya que ningún elemento de A aparece como primer elemento de dos pares ordenados diferentes. Aquí se tiene

$$\begin{aligned}f(1) &= a \\f(2) &= a \\f(3) &= d \\f(4) &= c\end{aligned}$$

El codominio de f , $\text{Cod}(f) = \{a, d, c\}$.

Una función puede tomar el mismo valor en dos elementos diferentes de A .

2. Sean $A = \{1, 2, 3\}$ y $B = \{x, y, z\}$
 $R = \{(1, x), (2, x)\}$ y $S = \{(1, x), (2, z), (3, y)\}$

Ninguna de estas relaciones es una función de A en B , por diferentes razones. La relación S no es una función ya que contiene los pares coordenados $(1, x)(1, y)$ lo que viola la propiedad (b) de la definición de una función.

La relación R no es una función de A en B , ya que el $\text{Dom}(R) \neq A$.

Tipos:

Una función $f: A \rightarrow B$ se llama **inyectiva**, o uno a uno si para toda a, a' en A ,
 $a \neq a'$ implica que $f(a) \neq f(a')$

La función f definida en el ejemplo 1 (**) no es inyectiva ya que
 $f(1) = f(2) = a$

- ♦ Sea $A = B = \mathbb{Z}$ y sea $f: A \rightarrow B$ definida por
 $f(a) = a + 1$ para $a \in A$

f consta de todos los pares ordenados $(a, a+1)$ para $a \in \mathbb{Z}$. Entonces cada $a \in A$ aparece como el primer elemento de algún par, por lo cual $\text{Dom}(f) = A$.

También, si $(a, b) \in f$ y $(a, c) \in f$, de modo que

$$\begin{aligned}b &= f(a) = a + 1 && \text{y} \\c &= f(a) = a + 1 && \text{entonces} \\b &= c\end{aligned}$$

Por consiguiente, f es una función. Supóngase que

$$f(a) = f(a') \quad \text{para } a \text{ y } a' \text{ en } A. \text{ Entonces}$$

$$\begin{array}{ll} a+1 = a'+1 & \text{por lo cual} \\ a = a' & \text{De aquí que } f \text{ sea inyectiva.} \end{array}$$

A una función $f: A \rightarrow B$ se le llama **suprayectiva** si $f(A) = B$, esto es, si el $\text{Cod}(f)=B$. f es suprayectiva si todo elemento $b \in B$ es el segundo elemento en algún par ordenado $(a, b) \in f$.

f es suprayectiva si para cada $b \in B$ se puede encontrar alguna $a \in A$ tal que $b=f(a)$.

Tomando el ejemplo con ♦ referencia. Sea b un elemento arbitrario de B .

Es posible encontrar un elemento $a \in A$ tal que

$$\begin{array}{ll} f(a) = b & \text{ya que} \\ f(a) = a+1 & \text{es necesario un elemento } a \text{ en } A \text{ tal que} \\ a+1 = b & \text{Por supuesto,} \\ a = b-1 & \end{array}$$

lo que satisface la ecuación deseada ya que $b-1$ está en A . De aquí que f sea suprayectiva.

Cuando una función es inyectiva y suprayectiva, se dice que f es una **biyección** o una correspondencia uno a uno.

A una función $A \rightarrow B$ se le llama **invertible** si su relación inversa, f^{-1} es también una función. Solo si f es inyectiva y suprayectiva (biyección) entonces es invertible.

• Números primos (R, E, RP)

El conjunto de los **números primos** es un subconjunto de los números naturales que engloba a todos los elementos de este conjunto que son divisibles exactamente tan sólo por dos números naturales (el 1, que sólo tiene un divisor natural, no es primo). Los veinte primeros números primos son: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67 y 71.

Nótese el hecho de que todos los números naturales son divisibles por si mismos y 1 (excepto 0 en el caso de que se considere en este conjunto, pues ningún número es divisible entre 0).

El Teorema fundamental de la Aritmética establece que cualquier entero positivo puede representarse siempre como un producto de números primos, y esta representación (factorización) es única.

¿Cuántos números primos existen?

Existen infinitos números primos. Euclides realizó la primera demostración alrededor del año 300 antes de nuestra era. Otros matemáticos han demostrado la infinitud de los números primos con métodos diversos, e incluso hay una demostración topológica.

A pesar de que sabemos que hay infinitos números primos, aún quedan preguntas en el aire sobre la distribución de los mismos o la lista de primos que hay por debajo de cierto número.

Un procedimiento empleado para hallar todos los números primos menores que un entero dado es el de la criba de Eratóstenes. Además se sabe que no hay límite para la distancia entre dos primos consecutivos, esto es, dado un número N , se puede encontrar dos números primos tales que entre ellos dos no hay otros números primos y su diferencia es mayor que N .

Aunque no se ha podido probar hasta la fecha, se conjectura que existen infinitos números primos de la forma $p_1=p_2 + 2$ (siendo p_1 y p_2 primos) o primos gemelos. Sí se ha probado que los únicos "primos trillizos" (primos de la forma $p_1 = p_2 + 2$ y $p_2 = p_3 + 2$) son 3, 5 y 7; y esto es así

porque uno de los números p_1 , p_2 y p_3 así definidos es múltiplo de 3, y por tanto compuesto cuando $p_3 > 3$.

Propiedades de los números primos:

- ✓ Si p es un número primo y divisor del producto de números enteros ab , entonces p es divisor de a o de b . (Lema de Euclides)
- ✓ Si p es primo y a es algún número entero, entonces $a^p - a$ es divisible por p (Pequeño Teorema de Fermat)
- ✓ Un número p es primo si y solo si el factorial $(p - 1)! + 1$ es divisible por p . (Teorema de Wilson)
- ✓ Si n es un número natural, entonces siempre existe un número primo p tal que $n < p < 2 \cdot n$. (Postulado de Bertrand)
- ✓ En toda progresión aritmética $a_n = a + n \cdot q$, donde los enteros positivos $a, q \geq 1$ son primos entre sí, existen infinitos números (Teorema de Dirichlet).
- ✓ El número de primos menores que un x dado sigue una función asintótica a $f(x) = \frac{x}{\ln x}$ (Teorema de los números primos).

Clases de Primos:

- ✓ Número primo de Fermat (de forma $2^2n + 1$)
- ✓ Número primo de Mersenne (de forma $M_p = 2^p - 1$ donde p es primo)
- ✓ Número primo de Sophie Germain (un p primo tal que $2p + 1$ es primo)
- ✓ Números primos gemelos (p y $p+2$ primos)

Conjeturas Sobre los Números Primos:

- ✓ Todo número par mayor o igual que 4 es suma de dos números primos. (Conjetura de Goldbach)
- ✓ Existen infinitos pares de números primos gemelos.
- ✓ Existen infinitos números primos de Fermat.
- ✓ Existen sólo 5 números primos de Fermat.
- ✓ Para cada n natural, existe algún número primo entre n^2 y $(n + 1)^2$.
- ✓ Existen infinitos números primos de la forma $n^2 + 1$
- ✓ La sucesión de Fibonacci contiene infinitos números primos.

• Ecuaciones y sistemas de ecuaciones módulo n (R, E, RP)

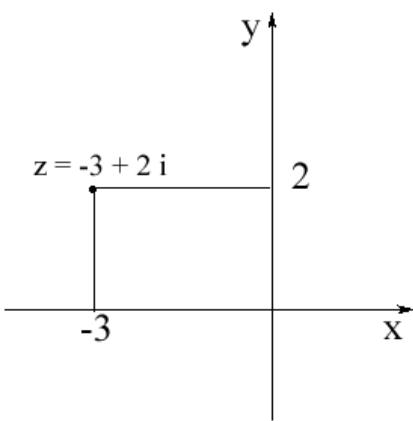
• Números complejos: operaciones y propiedades (R, E, RP)

Los números de la forma $z = a+bi$, donde a, b son números reales, y ademas i es la unidad imaginaria, siendo $i^2 = -1$, se denominan números complejo. La parte formada por a se la denomina **parte real** de z , y a la formada por b **parte imaginaria** de z .

Representación de los Números Complejos:

Se consideran dos ejes ortogonales, cada eje es una recta real, ambas rectas se intersectan en el origen, entonces al número complejo $z = a+bi$, le asignamos el punto del plano, cuya distancia orientada al eje horizontal (el eje x) es a , cuya distancia al eje vertical (eje y) es b .

Ejemplo: el número complejo $z = -3+2i$ esta representado por el punto de coordenadas (-3,2).



Operaciones con Números Complejos:

✓ Suma

Sea z y w números complejos, donde $z = a+bi$ y $w=c+di$
 $z+w = (a+bi) + (c+di) = (a+c) + (b+d)i$

Ejemplo : sean $z = -3+4i$ y $w = 1+2i$.
 $(-3+4i) + (1+2i) = (-3+1) + (4+2)i = -2+6i$

✓ Multiplicación

Sea z y w números complejos, donde $z = a+bi$ y $w=c+di$
 $z \cdot w = (a+bi)(c+di) = (a*c - b*d) + (a*d + b*c)i$

Ejemplo: $z = 2+3i$ y $w = 1-4i$
 $(2+3i)(1-4i) = (2*1 - 3*(-4)) + (2*(-4) + 3*1)i = 14-5i$

✓ División

Sea z y w números complejos, donde $z = a+bi$ y $w=c+di$, y w' conjugado de w , es decir $w' = c-di$
 $(a+bi)/(c+di) = (a+bi)(c-di)/(c+di)(c-di) = ((a*c+b*d) - (a*d-b*c)i)/(c*c+d*d)$

Ejemplo: $z = 2+i$ y $w = 3+2i$
 $(2+i)/(3+2i) = ((2+i)(3-2i))/((3+2i)(3-2i)) = 8-i/13$

✓ Norma

Sea $z = a+bi$, llamaremos norma de z a
 $|z|^2 = a^2 + b^2$

Ejemplo: $z = 3-2i$
 $|z|^2 = 9 + 4 = 13$

✓ Valor absoluto o módulo

Llamaremos valor absoluto de z , a
 $|z| = \sqrt{a^2 + b^2}$

Ejemplo: $z = 3-2i$
 $|z| = \sqrt{9 + 4} = \sqrt{13} = 3.6055..$

✓ **Forma polar o forma trigonométrica**

Un número complejo $z = a+bi$, se puede escribir en la forma trigonométrica.

$$z = r*(\cos(\rho) + i \sin(\rho))$$

donde $r = |z| = (\sqrt{a^2 + b^2})^{1/2}$ es la distancia desde el punto al origen.

$$\rho = \arctg(b/a)$$

ρ es el argumento de z , es el ángulo que forma con el eje x.

La siguiente rutina convierte un número complejo de la forma binomica $Z = x + yi$ a la trigonométrica, devolviendo r y el argumento en este caso lo llame theta

✓ **Potencia de un número complejo**

Para hallar la potencia de un número complejo aplicamos el Teorema De Moivre, donde $z = a+bi$ y n es número entero positivo.

$$z^n = r^n * (\cos(n*\rho) + i \sin(n*\rho))$$

donde $r = |z|$ es el módulo y ρ es el argumento de z .

La siguiente rutina halla la potencia de un número complejo $Z = x + yi$, elevado a una potencia N , la parte real y la imaginaria son retornados en los parametros a y b .

Propiedades de los Números Complejos:

- ✓ Dos números complejos $a+bi$ y $c+di$ son iguales si y solo si $a=b$, $y=c=d$

Se podría considerar a los números reales como un subconjunto de los números complejos, esto es cuando $b=0$

Ejemplo: $0+0i$ y $-3+0i$ representan los números reales 0 y -3 respectivamente.

- ✓ Si $a=0$ se considera un número imaginario puro a $0+bi$

Conjugado Complejo, o conjugado simplemente de un número $a + bi$ es el numero $a-bi$. Siendo z un numero complejo, su conjugado se presenta como: $Z = z^*$

2. Lineal

• **Espacios vectoriales sobre un campo (R, E, RP)**

• **Sistemas de ecuaciones lineales (R, E, RP)**

Un par de ecuaciones lineales se puede resolver trazando la gráfica de ambas sobre los mismos ejes y determinando las coordenadas del punto de intersección.

Cualquier sucesión de valores $x_1 = s_1$ y $x_2 = s_2$ tales que

$$a_{11}s_1 + a_{12}s_2 = b_1$$

$$a_{21}s_1 + a_{22}s_2 = b_2$$

le llamamos una solución del sistema de ecuaciones lineales.

Si el sistema de ecuaciones lineales tiene solución se le llama compatible o consistente. Si no tiene solución le llamamos incompatible o inconsistente.

Sistema con solución única.

Considérese el sistema

$$x - y = 7$$

$$x + y = 5$$

Al sumar las dos ecuaciones se obtiene, por el resultado A, la ecuación siguiente: $2x = 12$ (es decir, $x = 6$). Entonces, de la segunda ecuación, $y = 5 - x = 5 - 6 = -1$. Por lo tanto, el par $(6, -1)$ satisface el sistema. Por la forma en que se encontró la solución, se ve que no existe ningún otro par que satisfaga ambas ecuaciones. Por tanto, el sistema tiene una solución única.

Sistema con un número infinito de soluciones

Considérese el sistema

$$\begin{aligned} x - y &= 7 \\ 2x - 2y &= 14 \end{aligned}$$

Es obvio que estas dos ecuaciones son equivalentes. A fin de comprobar esto, multiplíquese la primera por 2. $x - y = 7$ o $y = x - 7$. Por tanto, el par $(x, x-7)$ es una solución del sistema para todo número real x . El sistema tiene un número infinito de soluciones. Por ejemplo, los pares siguientes son soluciones: $(7, 0), (0, -7), (8, 1), (1, -6), (3, -4)$ y $(-2, -9)$.

Sistema sin solución

Considérese el sistema

$$\begin{aligned} x - y &= 7 \\ 2x - 2y &= 13 \end{aligned}$$

Multiplicando la primera ecuación por 2, se obtiene $2x - 2y = 14$. Esto contradice a la segunda ecuación. Entonces el sistema no tiene solución.

Ejemplo: Resolver el sistema

$$\begin{aligned} 2x_1 + 4x_2 + 6x_3 &= 18 \\ 4x_1 + 5x_2 + 6x_3 &= 24 \\ 3x_1 + x_2 - 2x_3 &= 4 \end{aligned}$$

Dividiendo la primera ecuación entre 2.

$$\begin{aligned} x_1 + 2x_2 + 3x_3 &= 9 \\ 4x_1 + 5x_2 + 6x_3 &= 24 \\ 3x_1 + x_2 - 2x_3 &= 4 \end{aligned}$$

Multiplicando por -4 ambos lados de la primera ecuación y sumando esta nueva ecuación a la segunda. Se obtiene entonces:

$$\begin{array}{r} -4x_1 - 8x_2 - 12x_3 = -36 \\ 4x_1 + 5x_2 + 6x_3 = 24 \\ \hline -3x_2 - 6x_3 = -12 \end{array}$$

El sistema ahora es:

$$\begin{aligned} x_1 + 2x_2 + 3x_3 &= 9 \\ -3x_2 - 6x_3 &= -12 \\ 3x_1 + x_2 - 2x_3 &= 4 \end{aligned}$$

La primera ecuación se multiplica por -3 y el resultado se suma a la tercera ecuación:

$$\begin{aligned} x_1 + 2x_2 + 3x_3 &= 9 \\ -3x_2 - 6x_3 &= -12 \\ -5x_2 - 11x_3 &= -23 \end{aligned}$$

La segunda ecuación se divide entre -3 :

$$\begin{aligned} x_1 + 2x_2 + 3x_3 &= 9 \\ x_2 + 2x_3 &= 4 \\ -5x_2 - 11x_3 &= -23 \end{aligned}$$

La segunda ecuación se multiplica por -2 y el resultado se suma a la primera, y luego la segunda ecuación se multiplica por 5 y el resultado se suma a la tercera:

$$\begin{array}{r} x_1 - x_3 = 1 \\ x_2 + 2x_3 = 4 \\ - - x_3 = -3 \end{array}$$

La tercera ecuación se multiplica por -1:

$$\begin{array}{rcl} x_1 & - & x_3 = 1 \\ x_2 & + & 2x_3 = 4 \\ & & x_3 = 3 \end{array}$$

Por último, la tercera ecuación se suma a la primera y luego la tercera Ecuación se multiplica por -2 y el resultado se suma a la segunda, obteniéndose el sistema siguiente [el cual es equivalente al primer sistema]:

$$\begin{array}{rcl} x_1 & = 4 \\ x_2 & = -2 \\ x_3 & = 3 \end{array}$$

• Triangulación y diagonalización (R, E, RP)

Se dice que una matriz cuadrada es **triangular superior** si todos los elementos situados debajo de su diagonal principal son cero. Se dice que una matriz cuadrada es **triangular inferior** si todos los elementos situados arriba de su diagonal principal son cero.

Ejemplo:

$$\begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & -5 \\ 0 & 0 & 1 \end{pmatrix}$$

triangular superior

$$\begin{pmatrix} 1 & 0 & 0 \\ -5 & 1 & 0 \\ 2 & 3 & 1 \end{pmatrix}$$

triangular inferior

• Producto hermitiano (R, E, RP)

• Norma (R, E, RP)

Longitud o norma de un vector.- Si $v \in \mathbb{R}^n$, entonces la longitud o norma de v , denotada por $|v|$, está dada por:

$$|v| = \sqrt{v \cdot v}$$

Ejemplo:

Norma de un vector en \mathbb{R}^2 Sea $v = (x, y) \in \mathbb{R}^2$. Entonces $|v| = \sqrt{x^2 + y^2}$ es la definición ordinaria de la longitud de un vector en el plano.

Norma de un vector en \mathbb{R}^3 Si $v = (x, y, z) \in \mathbb{R}^3$, entonces $|v| = \sqrt{x^2 + y^2 + z^2}$

Norma de un vector en \mathbb{R}^5 Si $v = (2, -1, 3, 4, -6) \in \mathbb{R}^5$, entonces $|v| = \sqrt{4 + 1 + 9 + 16 + 36} = \sqrt{66}$.

• Proyecciones (R, E, RP)

Sean u y v en \mathbb{R}^2 diferentes de cero. La proyección de u en (sobre) v es el vector, denotado por proy_v^u , definido por:

$$\text{proy}_v^u = \frac{\underline{u} \cdot \underline{v}}{\underline{v}} \underline{v}$$

Al escalar $\frac{\underline{u} \cdot \underline{v}}{\underline{v}}$ le llamamos la componente de u en la dirección de v .

Ejemplos:

$$u = (2, 5)$$

$$v = (7, 3)$$

Encontrar proy_v^u

Encontramos la componente de u en la dirección de v
 $u \cdot v = (2)(7) + (5)(3) = 29$

$$\|v\|^2 = \sqrt{v_1^2 + v_2^2}$$

$$\|v\|^2 = 7^2 + 3^2 = 58$$

$$\frac{u \cdot v}{\|v\|^2} = \frac{29}{58} = \frac{1}{2}$$

$$\text{entonces } \text{proy}_v^u = \frac{1}{2} (7, 3)$$

$$= \left(\frac{7}{2}, \frac{3}{2} \right) = (3.5, 1.5)$$

¿Cuál es la distancia de u a proy_v^u ?

$$\begin{aligned} d(u, \text{proy}_v^u) &= \|u - \text{proy}_v^u\| \\ &= \|(-1.5, 3.5)\| \\ &= \sqrt{(-1.5)^2 + (3.5)^2} \\ &= \sqrt{14.5} \\ &= 3.8 \end{aligned}$$

• Bases ortogonales y ortonormales (R, E, RP)

Conjunto ortonormal en \mathbb{R}^n El conjunto de vectores $S = \{u_1, u_2, \dots, u_k\}$ en \mathbb{R}^n se llama conjunto ortonormal si

$$\begin{aligned} u_i \cdot u_j &= 0 && \text{si } i \neq j \\ u_i \cdot u_j &= 1 && \end{aligned}$$

Si sólo se satisface la ecuación (1), se dice que el conjunto es ortogonal.

Un conjunto de vectores es ortonormal si un par cualquiera de ellos es ortogonal y si cada uno tiene longitud 1.

• Valores y vectores propios (R, E, RP)

3. Teoría de grupos

• Grupos (R, E)

La teoría de grupos estudia las propiedades de los grupos, y uno de sus objetivos fundamentales es la clasificación de éstos.

En matemática, un grupo es un magma (i.e. un conjunto, con una operación binaria), que satisface ciertos axiomas, detallados abajo. La rama de las matemáticas que estudia los grupos es llamada teoría de grupos.

Sea una estructura formada por un conjunto, X , sobre cuyos elementos se ha definido una operación, $*$, o ley interna:

$$\begin{array}{ccc} \mathbf{X} \times \mathbf{X} & \mathbf{X} \\ (x,y) & x^*y \end{array}$$

Si la operación verifica las siguientes propiedades, entonces se dice que la estructura es un **Grupo** con respecto a la operación *.

1. **Asociativa:** para todo x, y, z verifica $x^*(y^*z) = (x^*y)^*z$
2. **Existencia del Elemento Neutro:** $x^*1 = 1*x = x$ para todo x de \mathbf{X} , siendo 1 el elemento neutro.
3. **Existencia de Inverso** (en caso de que la operación se denote aditivamente y no multiplicativamente, el término que se usa es Opuesto y el elemento neutro se denota 0): Para todo elemento x de \mathbf{X} , existe otro elemento y de \mathbf{X} , tal que: $x^*y = y^*x = 1$.

Grupo Comutativo o Abeliano:

Se denomina grupo commutativo o abeliano a aquel grupo que verifica la propiedad commutativa, es decir que para todo x, y de \mathbf{X} : $x^*y = y^*x$

Un grupo es un magma (i.e. un par $(A, *)$), donde A es un conjunto no vacío y * una ley de composición interna, esto es $*: A \times A \rightarrow A$), verificando:

1. $a^*(b^*c) = (a^*b)^*c$ para cualesquiera a, b, c de A (asociatividad)
2. En A existe un elemento denotado por 1 que cumple $1*a = a*1 = a$ (elemento neutro)
3. Para todo a de A existe a' tal que $a^*a' = a'^*a = 1$ (elemento inverso)

Ejemplos:

- ✓ $(\mathbb{R}, +)$ es grupo abeliano. \mathbb{R} es el conjunto de los números reales y $+$ la suma usual.
- ✓ $(\mathbb{R} - \{0\}, \cdot)$ es grupo abeliano. (Notar que el cero no tiene inverso multiplicativo, por eso se lo excluye).
- ✓ $(\mathbb{Z}_n, +)$ es grupo.

Un grupo es finito o infinito si el conjunto es finito o infinito. En nuestro ejemplo, los formados con \mathbb{R} son infinitos y el formado con \mathbb{Z}_n es finito.

Magma:

En álgebra abstracta, un magma es un tipo especialmente elemental de estructura algebraica. Específicamente, un magma consiste de un conjunto X con una sola operación binaria en él. Es, usualmente (pero no siempre), interpretado como una forma de multiplicación. Ningún axioma es impuesto a la operación al definir un magma. Los magmas no son usualmente estudiados como tales; sino que se consideran diferentes tipos de magmas, dependiendo de que axiomas se requieran de la operación.

Hay lo que podemos llamar un magma libre sobre cualquier conjunto X . Puede ser descrito, en términos familiares en ciencias de la computación como el magma de los árboles binarios con operación dada por la yuxtaposición (ordenada) de los árboles por la raíz. Tiene por tanto un rol fundacional en sintaxis.

Conjunto:

En matemáticas, un **conjunto** es una colección de objetos, tales que dos conjuntos son iguales si, y sólo si, contienen los mismos objetos. Se puede obtener una descripción mas detallada en la Teoría de conjuntos.

Los conjuntos son uno de los conceptos básicos de las matemáticas. Como ya se ha dicho, un conjunto es, más o menos, una colección de objetos, denominados **elementos**. La notación estándar utiliza llaves ('{ ' y ' }') alrededor de la lista de elementos para indicar el contenido del conjunto.

Operación:

Dar una operación en un conjunto X es asignar a cada par ordenado de elementos de X un único elemento de X.

Si una operación se denota con un símbolo, por ejemplo +, entonces el elemento de X asignado a un par (x,y) se denota x+y. Lo más usual es utilizar el símbolo + ó un punto. En tales caso se dice que se usa la notación aditiva ó multiplicativa.

Nótese que el conjunto X es parte de la operación. Así, la suma de números naturales es una operación diferente de la suma de números enteros (en ésta última es cierta la existencia de elemento opuesto, mientras que en la primera es falsa). Además, por definición, la operación ha de estar definida para cualquier par ordenado. Por ejemplo, la división de números racionales no es una operación, porque no está definida cuando el divisor es nulo, mientras que la división sí es una operación en el conjunto de los números racionales no nulos.

Axioma:

Parece claro que los axiomas se refieren a los principios que se aceptan en el estudio de todas las ciencias, mientras que los postulados se refieren a una ciencia en particular, la geometría en este caso. Por eso el axioma 9 parece que está intercalado y, según Enriques, debe colocarse entre los postulados, complementando al primero en el sentido de que por dos puntos pasa una única recta. Actualmente se llaman sencillamente axiomas a los principios que se aceptan sin demostración al desarrollar una teoría. O mejor aún, los axiomas caracterizan y definen la estructura que estudie la teoría en cuestión. Así, hablamos de los axiomas de grupo, los axiomas de anillo, los axiomas de la geometría proyectiva, de la geometría afín o de la geometría euclídea, etc.

III. Geometría analítica

1. Lineal

• Problemas relacionados con planos y rectas (R, E, RP)

Sistema de Referencia en el Plano:

Es un conjunto $R = \{O, (x,y)\}$ O punto fijo, llamado origen
(x,y) base

A cada punto p del plano se le asocia su vector posición op que tiene unas coordenadas.

El punto P da lugar al vector OP

El vector OP tiene de coordenadas (4,3) respecto de la base B(x,y)

El punto P tiene de coordenadas (4,3) respecto del sistema de referencia R
 Cambiando los valores de a y b puedes ver que a otro punto P, corresponde otro vector OP
 Las coordenadas de OP(a,b), siempre serán las coordenadas de P(a,b)

Aplicaciones de los Vectores a Problemas Geométricos:

✓ Coordenadas del vector que une dos puntos dados por sus coordenadas:

En esta escena hay tres vectores que cumplen:

$$\mathbf{OA} + \mathbf{AB} = \mathbf{OB}$$

Por tanto: $\mathbf{AB} = \mathbf{OB} - \mathbf{OA}$

Coordenadas de \mathbf{OA} = coordenadas de A

Coordenadas de \mathbf{OB} = coordenadas de B

De esta forma, coordenadas del vector \mathbf{AB} = coordenadas de su *extremo* B - coordenadas de su *origen* A

✓ Comprobación de que tres puntos están alineados:

Los puntos $A(x_1,y_1)$, $B(x_2,y_2)$, $C(x_3,y_3)$ están alineados siempre que los vectores \mathbf{AB} y \mathbf{BC} tengan la misma dirección.

Esto ocurre cuando sus coordenadas son proporcionales:

$$\begin{aligned}\mathbf{AB} &= (x_2-x_1, y_2-y_1) \\ \mathbf{BC} &= (x_3-x_2, y_3-y_2)\end{aligned}\quad \frac{x_2-x_1}{x_3-x_2} = \frac{y_2-y_1}{y_3-y_2}$$

✓ Punto medio de un segmento :

En esta escena aparece una suma de vectores: $\mathbf{OA} + \mathbf{OB} = \mathbf{OS}$, OS es la diagonal del paralelogramo OASB.

Las diagonales se cortan en sus puntos medios. Por tanto:

$$\mathbf{OM} = \frac{1}{2} \mathbf{OS} = \frac{1}{2} (\mathbf{OA} + \mathbf{OB}) = \left(\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2} \right)$$

donde $A=(x_1,y_1)$ y $B(x_2,y_2)$.

Las coordenadas del punto medio, M, de un segmento de extremos $A=(x_1,y_1)$, $B(x_2,y_2)$ son:

$$M = \left(\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2} \right)$$

Moviendo con el ratón los puntos A y/o B, podrás comprobar cuáles son las coordenadas del punto medio M, del segmento AB en cada caso.

2. No lineal

• Traslación y rotación en R2 y R3 (R, E, RP)

• Transformación de las coordenadas de un punto de un sistema cartesiano a un sistema esférico y viceversa (R, E, RP)

• Ecuaciones paramétricas y polares de las cónicas (R, E, RP)

$x = \cos \theta$, $y = \sin \theta$ Ecuaciones paramétricas o representación paramétrica de la circunferencia.

$x = f(t)$, $y = g(t)$ Ecuaciones paramétricas de la curva C

$$y = \pm \sqrt{1 - t^2}$$

Las ecuaciones paramétricas de un lugar geométrico específico no son únicas, ya que el lugar geométrico puede representarse por diferentes pares de ecuaciones.

$$\begin{aligned} x &= tv_0 \cos \alpha, & y &= tv_0 \sin \alpha - \frac{1}{2} gt^2 \\ x &= p \operatorname{ctg} 2\alpha, & y &= 2p \operatorname{ctg} \alpha \end{aligned}$$

Ecuaciones paramétricas de la parábola

$$\begin{aligned} x &= a \sec \theta, & y &= b \sin \theta \end{aligned}$$

Ecuaciones paramétricas cónicas de la parábola

$$\begin{aligned} x &= a \sec \theta, & y &= b \tan \theta \end{aligned}$$

Ecuación de la representación paramétrica de la hipérbola

$$\begin{aligned} x &= a(\theta - \sin \theta), & y &= a(1 - \cos \theta) \end{aligned}$$

Ecuaciones paramétricas de la cicloide

$$\begin{aligned} x &= (a + b) \cos \theta - b \cos \frac{a+b}{b} \theta, & y &= (a + b) \sin \theta - b \sin \frac{a+b}{b} \theta, \end{aligned}$$

Ecuaciones paramétricas de la epicicloide

$$\begin{aligned} y &= (a + b) \sin \theta - b \sin \frac{a+b}{b} \theta, \\ x &= 2a \cos \theta - a \cos 2\theta, \\ y &= 2a \sin \theta - a \sin 2\theta \end{aligned}$$

Ecuaciones paramétricas de la cardioide

$$\begin{aligned} x &= (a - b) \cos \theta + b \cos \frac{a-b}{b} \theta, & y &= (a - b) \sin \theta - b \sin \frac{a-b}{b} \theta, \end{aligned}$$

Ecuaciones paramétricas de la hipocicloide

$$\begin{aligned} x &= a \cos^3 \theta, & y &= a \sin^3 \theta \end{aligned}$$

Ecuaciones paramétricas de la astroide

$$\begin{aligned} x &= r \cos \theta, & y &= r \sin \theta \end{aligned}$$

Ecuación polar del lugar geométrico

$$p = r \cos(\theta - w)$$

Ecuación polar de la recta

$$a^2 = r^2 - 2cr \cos(\theta - \alpha) + c^2$$

Ecuación polar de una circunferencia de centro el punto (c, a) y radio igual a a

$$r = a$$

Ecuación polar si centro está en el polo

$$r = \pm 2a \cos \theta$$

Ecuación polar si circunferencia pasa por el polo y el centro está sobre el eje polar

$$r = \pm 2a \sin \theta$$

Ecuación polar si circunferencia pasa por el polo y el centro está sobre el eje a 90°

Se debe tomar el signo positivo o negativo
Según el centro esté arriba o abajo del polo

$$r = \frac{ep}{1 - e \cos \theta}$$

Ecuación de la cónica con directriz a la izquierda del polo

$$r = \frac{ep}{1 + e \cos \theta}$$

Ecuación de la cónica con directriz a la derecha del polo y a p unidades de él

• Características de una superficie cuadrática con ejes paralelos a los coordenados, a partir de su ecuación (R, E)

Ecuación Cuadrática y Forma Cuadrática:

- i. Una ecuación cuadrática con dos variables sin términos lineales es una ecuación de la forma

$$ax^2 + bxy + cy^2 = d$$

donde $|a| + |b| + |c| \neq 0$. Es decir, por lo menos uno de los números a , b o c es diferente de cero.

- ii. Una forma cuadrática con dos variables es una expresión de la forma

$$F(x, y) = ax^2 + bxy + cy^2$$

donde $|a| + |b| + |c| \neq 0$.

Teorema de los Ejes Principales en \mathbb{R}^2 :

Sea $ax^2 + bxy + cy^2 = d^{(*)}$ una ecuación cuadrática en las variables x y y . Entonces existe un solo número θ en $[0, 2\pi]$ tal que la ecuación $(*)$ se puede escribir en la forma

$$a'x'^2 + c'y'^2 = d$$

donde x' , y' son los ejes que se obtienen al rotar los ejes x y y un ángulo θ en sentido antihorario. Por otra parte, los números a' y c' son los valores característicos de la matriz

$$A = \begin{pmatrix} a & b/2 \\ b/2 & c \end{pmatrix}$$

Los ejes x' y y' reciben el nombre de **ejes principales** de la gráfica de la ecuación cuadrática *.

Si $A = \begin{pmatrix} a & b/2 \\ b/2 & c \end{pmatrix}$, entonces la ecuación cuadrática $ax^2 + bxy + cy^2 = d$ con $d \neq 0$ es

la ecuación de:

- i. Una hipérbola si $\det A < 0$.
- ii. Una elipse, circunferencia o sección cónica degenerada si $\det A > 0$.
- iii. Un par de líneas rectas o una sección cónica degenerada si $\det A = 0$.
- iv. Si $d = 0$, entonces la ecuación $ax^2 + bxy + cy^2 = d$ es la de dos líneas rectas si $\det A \neq 0$, y es la ecuación de una sola recta si $\det A = 0$.

Ecuaciones Estándar:

Circunferencia: $x^2 + y^2 = r^2$

Elipse: $x^2 + y^2 = 1$

Hipérbola:
$$\left\{ \begin{array}{l} x^2 - y^2 = 1 \\ \text{o} \end{array} \right.$$

$$y^2 + x^2 = 1$$

Hay una gran variedad de superficies tridimensionales de la forma $A\mathbf{v} \cdot \mathbf{v} = d$, siendo $\mathbf{v} \in \mathbb{R}^n$. A dichas superficies se les llama **superficies cuadráticas**. Las formas cuadráticas pueden definirse con un número cualquiera de variables.

Forma cuadrática Sean $\mathbf{v} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{pmatrix}$ una matriz simétrica de $n \times n$.

Entonces una **forma cuadrática** en x_1, x_2, \dots, x_n es una expresión de la forma

$$F(x_1, x_2, \dots, x_n) = A\mathbf{v} \cdot \mathbf{v}$$

• Ecuación de una superficie cuadrática (E)

C. Matemáticas discretas

I. Lógica

1. Lógica proposicional

• Reglas de inferencia (R, E, RP)

Se describe el proceso de derivación, mediante el cual se demuestra que una formula particular es la consecuencia válida de un conjunto dado de premisas. Pero antes de hacerlo, se dan dos reglas de inferencia que se conocen como P y T

Regla P: una premisa puede introducirse en cualquier punto de la deducción

Regla T: una formula S puede introducirse en una deducción si S está tautológicamente implícito en cualquiera de una o más formulas precedentes en la deducción.

Eje.

Demuestre que R es una inferencia válida de las premisas P Q, Q R y P

Solución:

{1}	(1)	P Q	REGLA P
{2}	(2)	P	REGLA P
{1,2}	(3)	Q	REGLA T (1),(2) y In
{4}	(4)	Q R	REGLA P
{1,2,4}	(5)	R	REGLA T (3),(4) y In

La segunda columna de números designa la Formula así como la línea de deducción en la que esta ocurre. El conjunto de números entre corchetes (la primera columna) de cada línea muestra las premisas de las que depende la formula en esa línea. A la derecha P o T representa la regla de inferencia, seguida por un comentario que muestra de qué formulas y tautologías, se ha obtenido esa formula en particular. Por ejemplo, si se sigue esta notación, la tercera línea muestra que la formula en ella está numerada (3) y se ha obtenido de las premisas (1) y (2). El

comentario a la derecha dice que la Q se ha introducido usando la regla T y también indica los detalles de la aplicación de la regla T.

Caso Práctico:

“Cuando había un juego de pelota, transportarse era difícil, si llegaban a tiempo, entonces transportarse no era difícil, llegaron a tiempo por lo tanto no había juego de pelota” Muestre que estos enunciados constituyen un argumento válido

Solución:

P: Había un juego de pelota

Q: Transportarse era difícil

R: Llegaron a tiempo

Se debe demostrar que de las premisas $P \wedge Q, R \rightarrow Q$, y R se obtiene la conclusión P esto es igual a:

R P

Condición P Q si había Juego de pelota entonces el transporte era difícil

Negación B el transporte no era difícil

V Indica dos opciones P V Q ≡ P o Q ≡ P||Q

Λ Indica la conjunción “Y” si $P \wedge Q \equiv P$ y Q

◀ ▶

Modus Ponens

$$[p \wedge (p \rightarrow q)] \Rightarrow q \quad \text{Modus Ponens}$$

$$[(p \rightarrow q) \wedge \neg q] \Rightarrow \neg p$$

Modus Tollens o demostración por contradicción

$$[(p \rightarrow q) \wedge (q \rightarrow r)] \Rightarrow (p \rightarrow r) \quad \text{Ley del silogismo}$$

2. Lógica de predicados

• Reglas para fórmulas bien formadas (R, E, RP)

Una Formula bien formada es una expresión que consta de una cadena de variables (letras mayúsculas con o sin subíndices), paréntesis símbolos conectivos.

Una fórmula esta bien formada cuando esta bien estructurada de acuerdo con el conjunto de reglas establecidas y con el dominio de discurso.

Una Formula bien formada puede generarse mediante las siguientes reglas:

1. Si una variable enunciativa individual es una formula bien formada
 2. Si “A” es una formula bien formada, entonces A es una formula bien formada
 3. Si A y B son formulas bien formadas entonces las siguientes también son formulas bien formadas: $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$ y $(A \leftarrow B)$
 4. Una cadena de símbolos que contenga variables enunciativas, conectivas y paréntesis es una formula bien formada si esta puede obtenerse mediante un número finito de aplicaciones de las reglas 1,2 y 3.

- Formas clausales: resolución, unificación y PROLOG (R, E, RP)

La **resolución** es una técnica para demostrar teoremas en el lenguaje de la lógica. La resolución considera 2 ó más cláusulas e intenta deducir hechos o relaciones no explícitas en las cláusulas originales del problema.

Las demostraciones por resolución involucran los siguientes pasos:

1. Poner las premisas o axiomas en forma clausular
2. Agregar al conjunto de axiomas la negación de lo que se demostrará en forma clausular
3. Resolver las cláusulas entre sí, produciendo nuevas cláusulas que se sigan lógicamente de ellas
4. Producir una contradicción, generando la cláusula vacía

Ejemplo:

Si estudio obtengo buenas calificaciones, si no estudio me divierto

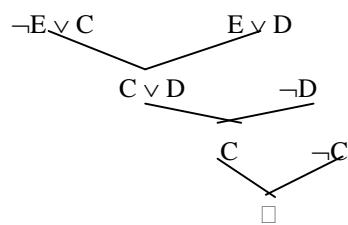
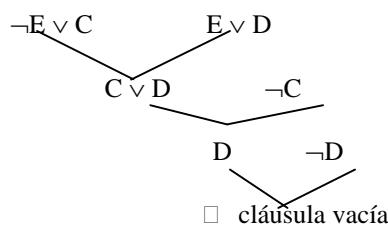
Conclusión: Por lo tanto u obtengo buenas calificaciones o me divierto

E – Estudio

C – Calificaciones buenas

D – Divierto

$$\frac{\neg E \vee C \\ E \vee D}{C \vee D}$$



II. Combinatoria

1. Análisis combinatorio

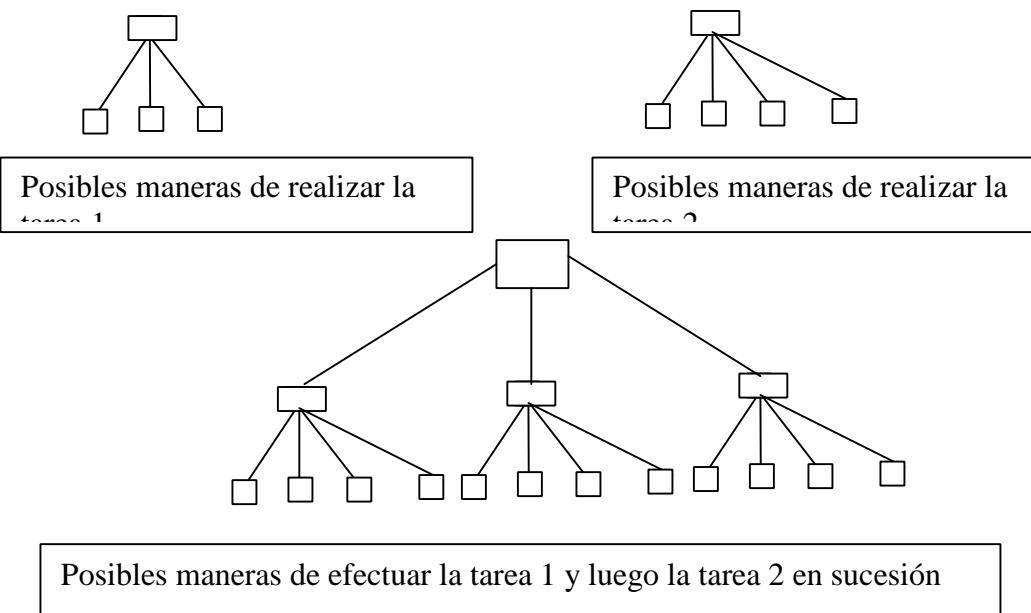
• Teoría de conteo (R, E, RP)

Las técnicas de conteo son importantes en matemáticas y en las ciencias de la computación particularmente en el análisis de algoritmos,

- ✓ Técnica de adición
- ✓ Permutaciones

Teorema 1 o principio de multiplicación del conteo. Supóngase que se va a efectuarse 2 trabajos T1 y T2 en secuencia si T1 puede realizarse de n_1 maneras y para una de estas maneras T2 puede llevarse a cabo de n_2 maneras, entonces la secuencia t_1t_2 puede efectuarse de n_1n_2 maneras.

Eje1: Cada elección de método para realizar T1 dará lugar a una manera diferente de efectuar la secuencia de trabajos. Existen n_1 de estos métodos y para cada uno de ellos puede elegirse n_2 maneras de llevar a cabo T2. Así en total habrá n_1n_2 maneras efectuar la sucesión T1T2 véase las siguientes. Figura en caso de que n_1 sea 3 y n_2 sea 4:



Teorema 2 y Eje 2: Un identificador de etiqueta para un programa de computadora, consta de una letra seguida por tres dígitos, si se permite repeticiones Cuantos identificadores distintos de etiqueta será posible tener?

Solución:

Hay 26 posibilidades para la letra inicial y 10 posibilidades para cada uno de los tres dígitos. En consecuencia por el principio extendido de la multiplicación hay $26 \times 10 \times 10 \times 10$, o sea, 26,000 identificadores posibles de etiqueta

Teorema 3: Sea A u conjunto de n elementos y $1 \leq r \leq n$, entonces el número de secuencias de longitud r que puede formarse con elementos de A permitiendo repeticiones, es n^r

Eje 3: Cuantas palabras de tres letras puede formarse a partir de letras del conjunto {a, b, y, z} si se permite repetir letras?

Solución:

En este caso n es 4 y r es 3 de manera que el número de tales palabras es 4^3 es decir 64 por el teorema 3

Teorema 4: Si $1 \leq r \leq n$, entonces ${}_nP_r$ el número de permutaciones de n objetos tomados r a la vez, es $n(n-1)(n-2).....(n-r+1)$

Eje 4: Sea A el conjunto {a,b,c}. entonces, las permutaciones posibles de A son las secuencias abc, acb, bca, cab y cba

Si se conviene definir $0!$ Igual a 1, entonces, para cada $n \geq 0$ se ve que el número de permutaciones de n objetos es $n!$ si $n = 0$ y $1 \leq r \leq n$, se puede dar ahora una forma más compacta para: ${}_nP_r$ como sigue:

$$\begin{aligned} {}_n P_r &= n(n-1)(n-2)\dots(n-r+1) \\ &= \underline{n(n-1)\dots(n-r+1)(n-r)(n-r-1)\dots(2)(1)} \\ &\quad (n-r)(n-r-1)\dots(2)(1) \\ &= \frac{n!}{(n-r)!} \end{aligned}$$

Definición.- Un conjunto de objetos en donde el orden de los objetos no importa se llama combinación.

Siempre que el enunciado tenga la palabra “elegir, escoger, tomar, sacar una muestra, de cuántas formas” será conteo o combinaciones.

Principio de conteo

Para encontrar el número total de opciones para un evento, se multiplica el número de opciones para cada parte.

Ejemplos:

Una cantante tiene 4 blusas y 5 pantalones para usar en conciertos. ¿De cuántas maneras diferentes se puede vestir para un concierto?

$$R = 4 \times 5 = 20$$

Un identificado tipo etiqueta de un programa de computadora, se forma con una letra seguida de tres dígitos. Si se permiten las repeticiones, ¿cuántos identificadores distintos son posibles?

Solución: Hay 26 posibilidades para la letra inicial y 10 para cada uno de los tres dígitos. Entonces, por el principio de multiplicación generalizado, hay

$$26 \times 10 \times 10 \times 10 = 26\,000 \text{ identificadores posibles}$$

El nombre tradicional combinatorio para un subconjunto de r elementos de un conjunto S de n elementos es una combinación de n elementos tomados de r en r .

Fórmula:

$$C_r^n = \frac{n!}{r!(n-r)!}$$

• Relaciones de recurrencia (R, E, RP)

Definición.- Una relación de recurrencia para una sucesión a_0, a_1, \dots , es una ecuación que relaciona a_n , con alguno de sus antecesores a_0, a_1, \dots, a_{n-1} .

Una relación de recurrencia define indirectamente el término n -ésimo de una sucesión.

Ejemplo:

Una serie geométrica es una sucesión infinita de números, como 5, 15, 45, 135, ..., donde la división de cualquier término, distinto del primero, entre su predecesor inmediato es una constante llamada razón común.

Para el ejemplo anterior esta razón común es 3, ya que $15 = 3(5)$, $45 = 3(15)$, etc. Si a_0, a_1, a_2, \dots , es una serie geométrica, entonces $a_1/a_0 = a_2/a_1 = \dots = a_{n+1}/a_n = \dots = r$, la razón común. En esta serie geométrica particular, resulta $a_{n+1} = 3a_n$, $n \geq 0$.

La relación de recurrencia $a_{n+1} = 3a_n$, $n \geq 0$ no define una serie geométrica única. Para especificar la sucesión particular descrita por $a_{n+1} = 3a_n$, es necesario conocer uno de los términos de dicha sucesión.

De ahí que $a_{n+1} = 3a_n$, $n \geq 0$, $a_0 = 5$.

Cuando el problema es encontrar una formula explícita para una sucesión definida por recurrencia, la formula recursiva es llamada relación recurrente, Debe recordarse que para

definir una sucesión de forma recursiva, una fórmula recursiva debe ir acompañada de información acerca del comienzo de la sucesión. A esta información se le llama condición o condiciones iniciales para la sucesión.

Eje1.

La relación de recurrencia $a_n = a_{n-1} + 3$ con $a_1 = 4$ define de manera recursiva la sucesión 4,7,10,13 la condición inicial es $a_1 = 4$

Fibonacci 1,1,2,3,5,8,13,21 $f_n = f_{n-1} + f_{n-2}$, $f_1 = 1$ y $f_2 = 1$

Las relaciones de recurrencia aparecen en forma natural en muchos problemas de conteo y en el análisis de problemas de programación.

Eje. 2

La relación de recurrencia $a_n = a_{n-1} + 3$ con $a_1 = 2$ define la sucesión 2, 5, 8, ... se efectúa un análisis hacia atrás al valor de a_n . Sustituyendo la definición de a_{n-1} , a_{n-2} y así sucesivamente, hasta tener un esquema claro

$$\begin{array}{lll} a_n = a_{n-1} + 3 & \text{o bien} & a_n = a_{n-1} + 3 \\ = (a_{n-2} + 3) + 3 & & = a_{n-2} + (2)(3) \\ = ((a_{n-2} + 3) + 3) + 3 & & = a_{n-2} + (3)(3) \end{array}$$

En algún momento este proceso producirá:

$$\begin{aligned} a_n &= a_{n-(n-1)} + (n-1)(3) \\ &= a_1 + (n-1)(3) \\ &= 2 + (n-1)(3) \end{aligned}$$

III. Relaciones y grafos

1. Relaciones

• Órdenes parciales (R, E, RP)

Definición.- Una relación puede considerarse como un cuadro que muestra las correspondencias de unos elementos con respecto a otros.

Definición.- Una relación R en un conjunto A recibe el nombre de relación de orden parcial si R es reflexiva, antisimétrica y transitiva.

Una relación R en un conjunto A es un orden parcial si R es reflexiva, antisimétrica y transitiva. El conjunto A, junto con el orden parcial R, es un conjunto parcialmente ordenado y se denotará este conjunto por (A, R) si no existe posibilidad de confusión acerca del orden parcial, será posible referirse al conjunto parcialmente ordenado como A, en vez de (A, R).

Eje1.

Sea "A" una colección de subconjuntos de un conjunto S, la relación \subseteq de inclusión de conjuntos es un orden parcial en A, de modo que (A, \subseteq) es un conjunto parcialmente ordenado

Eje2.

Sea Z^+ el conjunto de enteros positivos. La relación usual (menor o igual que) es un orden para el Z^+ , como (mayor o igual que)

Eje3.

La relación de divisibilidad ($a R b$ si y solo si $a|b$) es un orden parcial en Z^+

Eje4.

Sea R el conjunto de todas las relaciones de equivalencia en un conjunto A como R consta de subconjuntos de A x A, R es un conjunto parcialmente ordenado bajo el orden parcial de contención de conjuntos, si R y S son relaciones

de equivalencia en A, puede expresarse la misma propiedad en notación relacional como sigue:

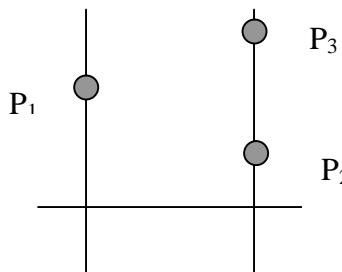
$R \subseteq S$ si y solo si $x R y$ implica $x S y$ para toda x, y en E

Entonces (R, \subseteq) es un conjunto parcialmente ordenado

Eje 8.

Sea $A=R$, con el orden usual entonces el plano $R^2 = RxR$ puede tener un orden lexicográfico, la figura siguiente ilustra esto. Se observa que el plano es ordenado linealmente por el orden lexicográfico. Cada recta vertical tiene el orden usual y los puntos en una recta son menores que cualesquiera de los puntos en una recta mas a la derecha así en la figura

$$P_1 < P_2, P_1 < P_3 \text{ y } P_2 < P_3$$



• Relaciones de equivalencia (R, E, RP)

Definición.- Una relación que es reflexiva, simétrica y transitiva en un conjunto X se conoce con el nombre de relación de equivalencia sobre X.

Sean A y B dos formulas enunciativas y seas P_1, P_2, \dots, P_n todas las variables aparecen tanto en A como en B. Considérese una asignación de valores de verdad para P_1, P_2, \dots, P_n y los valores de verdad resultantes de A y B, si el valor de A es igual al valor de verdad de B para cada uno de los 2^n posibles conjuntos de valores de verdad asignados a P_1, P_2, \dots, P_n se dice entonces que A y B son equivalentes suponiendo que las variables y la asignación de los valores de verdad a las variables aparecen en el mismo orden en las tablas de verdad de A y B son equivalentes

1. P es equivalente a P
2. PvP es equivalente a P
3. $(P \rightarrow P)vQ$ es equivalente a Q
4. $P \vee P$ es equivalente a $Q \vee Q$

En la definición de equivalencia de dos formulas no es necesario suponer que ambas contienen las mismas variables. Este punto se ilustra en los ejemplos (3) y (4) anteriores sin embargo puede notarse que si dos formulas son equivalentes y una variable particular aparece en solo una de ellas, entonces los valores de verdad de esta formula son independientes de esta variable. Por ejemplo en (3) el valor de verdad de $(P \rightarrow P)vQ$ es independiente del valor de verdad de P . De manera similar en (4), los valores de verdad de $P \vee P$ y de $Q \vee Q$ son, cada uno independientes de P y Q

Formulas Equivalentes

$PvP \rightarrow P$	$P \rightarrow P$	Leyes de indempotencia	1
$(PvQ) \vee R \rightarrow Pv(Q \vee R)$	$(P \rightarrow Q) \wedge R \rightarrow P \wedge (Q \rightarrow R)$	Leyes asociativas	2
$P \times Q \rightarrow Q \vee P$	$P \wedge Q \rightarrow Q \wedge P$	Leyes commutativas	3
$Pv(Q \rightarrow R) \rightarrow (PvQ) \wedge (PvR)$	$P \wedge (QvR) \rightarrow (P \wedge Q) \vee (P \wedge R)$	Leyes distributivas	4
$P \vee G \rightarrow P$	$P \rightarrow T \rightarrow P$		5
$P \vee T \rightarrow T$	$P \rightarrow F \rightarrow F$		6
$P \vee P \rightarrow T$	$P \rightarrow P \rightarrow P$		7
$P \vee (P \rightarrow Q) \rightarrow P$	$(P \rightarrow Q) \rightarrow P \vee Q$	Leyes de absorción	8
$(P \vee Q) \rightarrow P \wedge Q$		Leyes de Morgan	9

Eje3. Demuestre que $(P \rightarrow (Q \rightarrow R)) \vee (Q \rightarrow R) \vee (P \rightarrow R) \rightarrow R$

Solución:

$$\begin{aligned}
 & (\neg P \wedge (\neg Q \wedge R)) \vee (Q \wedge R) \vee (\neg P \wedge \neg R) \\
 \rightarrow & (\neg P \wedge (\neg Q \wedge R)) \vee ((Q \vee P) \wedge R) & 4 \\
 \rightarrow & ((\neg P \wedge \neg Q) \vee ((Q \vee P) \wedge R)) & 2 \\
 \rightarrow & ((\neg P \wedge \neg Q) \vee (Q \vee P) \wedge R) & 4 \\
 \rightarrow & ((\neg P \vee Q) \wedge (\neg Q \vee P)) \wedge R & 9,3 \\
 \rightarrow & T \wedge R & 7 \\
 \rightarrow & R & 5
 \end{aligned}$$

2. Gráficas y árboles

- Problemas clásicos: recorridos, números cromáticos, coloración de aristas y vértices, apareamientos (R, E, RP)

Recorridos o Trayectorias.

Una trayectoria (camino) en una grafica es una sucesión de (v_1, v_2, v_k) de vértices, cada uno adyacente al siguiente y una elección de aristas entre v_i y v_{i+1} de modo que ninguna arista es elegida mas de una vez. En términos geométricos, esto significa que es posible inicial en v_1 y viajar a través de las aristas hasta v_k y nunca utilizar la misma arista dos veces.

Un circuito es una trayectoria que inicia y termina con el mismo vértice. La palabra circuito es más común en la teoría general de graficas. Trayectoria v_1, v_2, v_k es simple si los vértices v_1, v_2, v_{k-1} , son todos distintos

Definición.- Sea $G = (V, A)$ un grafo no dirigido. G se denomina árbol si es conexo y no contiene ciclos.

Como un lazo es un ciclo de longitud uno, un árbol no tiene lazos.

Cuando un grafo es no conexo no puede ser un árbol, pero cada componente del grafo es un árbol y se denomina bosque.

Cuando un grafo es un árbol se escribe R en lugar de G para destacar su estructura.

Recorrido en árboles

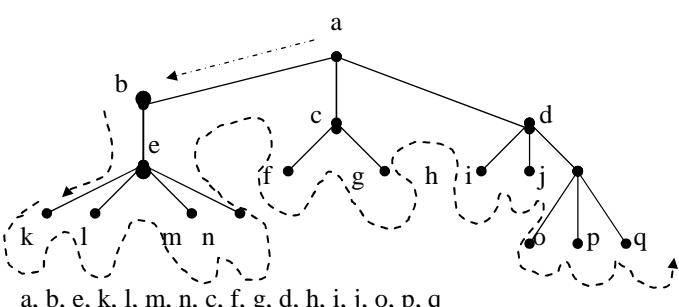
Los algoritmos encaminados a visitar los vértices de un árbol ordenado con raíz reciben el nombre de recorrido y los más comunes son: recorrido preorden, postorden e inorden.

Sea $R = (V, E)$ un árbol ordenado y con raíz en r , si R no tuviera ningún otro vértice la raíz r constituirá el recorrido preorden o postorden de R si $|V| > 1$, sean R_1, R_2, R_n Subárboles de R tomados de izquierda a derecha:

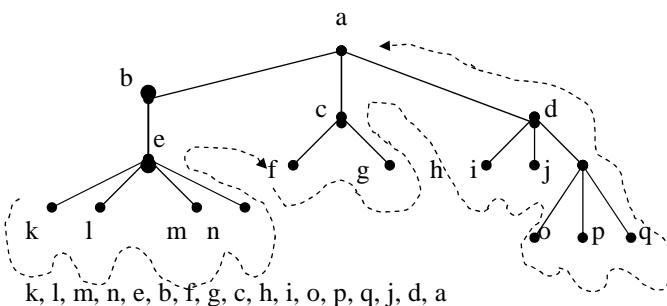
El **Reccorrido preorden** de R visita, en primer lugar, r y luego los vértices R_1 en preorden, luego lo de R_2 , también en preorden y así sucesivamente hasta recorrer en preorden los vértices de R_n .

El **Reccorrido postorden** de R , comienza visitando, en primer lugar, los vértices de R_1 en postorden y luego del de R_2 también es postorden y así sucesivamente hasta recorrer en postorden los de R_n se finaliza visitando la raíz

Reccorrido en **preorden**, para ello visitaremos primero a y luego los subárboles cuyas raíces estén en el nivel 1 y de izquierda a derecha: raíces b, c, y d

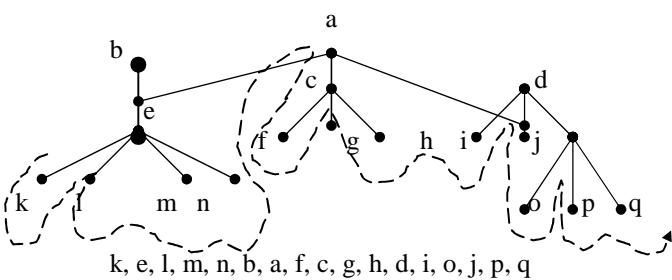


Ahora aplicaremos al mismo árbol el recorrido **postorden**

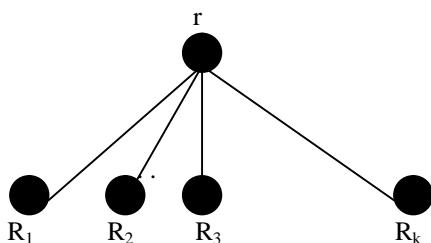


Por último definiremos y estudiaremos el algoritmo correspondiente al recorrido **inorden**

Definición: Sea $R = (V, E)$ un árbol ordenado y con raíz en r si R no tuviera ningún otro vértice, entonces la raíz r constituiría el recorrido inorden de R , si $|V| > 1$, sea R_1, R_2, \dots, R_n subárboles de R tomados de izquierda a derecha. Entonces el recorrido inorden comienza recorriendo el primer subárbol R_1 en inorden e inmediatamente visitando la raíz r , se continua recorriendo en inorden R_2 , etc. Para terminar recorriendo en inorden el último subárbol R_n .

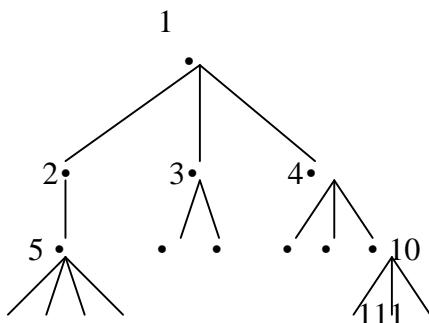


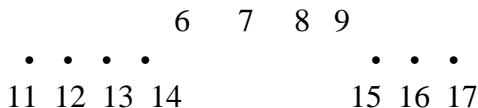
Definición.- Sea $R = (V, A)$ un árbol con raíz r . Si R no tiene otros vértices, entonces la raíz misma constituye el recorrido en orden previo, simétrico y posterior de R . Si $|V| > 1$, sean $R_1, R_2, R_3, \dots, R_k$ los subárboles de R según se va de izquierda a derecha.



- El recorrido en orden previo de R comienza en r y después pasa por los vértices de R_1 en orden previo, a continuación por los vértices de R_2 en orden previo, y así sucesivamente hasta que se pasa por los vértices de R_k en orden previo.
- El recorrido en orden simétrico de R primero, se pasa por los vértices de R_1 en orden simétrico, después por la raíz r y a continuación por los vértices de los subárboles R_2, R_3, \dots, R_k en orden simétrico.
- El recorrido en orden posterior de R pasa por los vértices de los subárboles R_1, R_2, \dots, R_k en orden posterior y a continuación por la raíz.

Ejemplo:





- a) Recorrido en orden previo: 1, 2, 5, 11, 12, 13, 14, 3, 6, 7, 4, 8, 9, 10, 15, 16, 17.
- b) Recorrido en orden simétrico: Comenzando en el vértice 1, se recorren los vértices del subárbol R1, con raíz del vértice 2 en orden simétrico. Esto lleva al vértice 5 y después al 11, una hoja. De ahí que al pasar en orden simétrico por los vértices del subárbol con raíz del vértice 5, se comience listando los vértices visitados como 11, 5, 12, 13, 14. A continuación, se pasa por el vértice 2, la raíz del subárbol R1 y después por la raíz del vértice 1. Para completar este recorrido, ahora se debe pasar por los vértices de los subárboles R2 y R3 en orden simétrico. Para R2 esto lleva del vértice 6 al vértice 3 (la raíz de R2) y después al vértice 7. Por último, para el subárbol R3 con raíz en el vértice 4 primero, se pasa por el vértice 8, después por la raíz en el vértice 4 y s continuación, por el 9. Después de que se recorre el subárbol con raíz del vértice 10; resulta el listado en orden simétrico 15, 10, 16 y 17. En consecuencia, el recorrido en orden simétrico del árbol determina la sucesión (en orden simétrico) 11, 5, 12, 13, 14, 2, 1, 6, 3, 7, 8, 4, 9, 15, 10, 16, 17 para los vértices.
- c) Recorrido en orden posterior: Para el recorrido en orden posterior de un árbol se comienza en la raíz y se construye el camino más largo, yendo al hijo situado más a la izquierda de cada vértice interno al que se llegue. Al llegar a una hoja h se pasa por este vértice y se retrocede hasta su padre p. Sin embargo, no se pasa por p hasta después de pasar por todos sus descendientes. El siguiente vértice por el que se pasa se halla aplicando el mismo procedimiento a p que a r, para obtener h. Nunca se pasa por un vértice más de una vez o antes que por sus descendientes. El recorrido en orden posterior pasa por los vértices en el orden 11, 12, 13, 14, 5, 2, 6, 7, 3, 8, 9, 15, 16, 17, 10, 4, 1.

Número Cromático

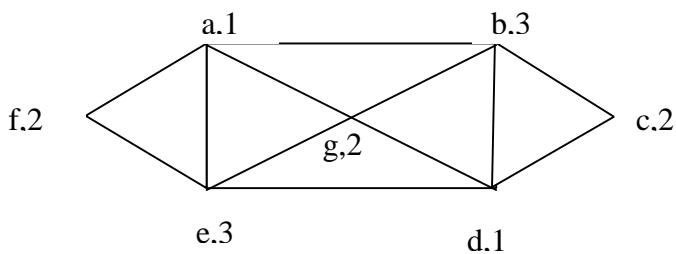
Definición.- Si $G = (V, A)$ es un grafo no dirigido, aparece una coloración apropiada de G cuando se colorean los vértices de G de modo que si $\{a, b\}$ es una arista de G, entonces a y b se pintan con colores distintos. (De ahí que vértices adyacentes tengan colores distintos).

El número mínimo de colores necesarios para colorear de forma apropiada G se denomina **número cromático** de G y se escribe $\gamma(G)$.

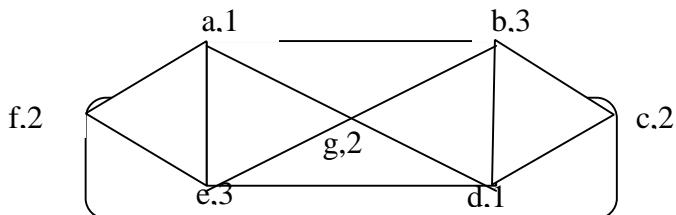
El **número cromático** de un grafo corresponde al **número** de vértices que posee el grafo, El número cromático de un grafoplano es igual o menor que cuatro.

Hay que señalar que aunque son suficientes cuatro colores para colorear las regiones de un mapa plano, serían necesarios, sin embargo mas de cuatro para colorear algunos grafos no planos, los cuales poseen números cromáticos grandes y arbitrarios.

El grafo que se representa a continuación puede ser coloreado empleando únicamente tres colores, es decir, será $X(g) = 3$



Para colorear el siguiente grafo plano, modificado del anterior serán necesario 4 colores: $X(G) = 4$



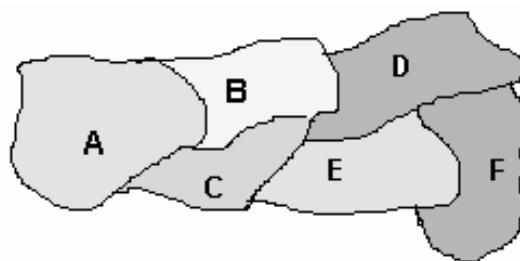
Coloración de Graficas (vértices)

Suponga que $G = (V, E)$ es una grafica sin aristas múltiples y $C = \{C_1, C_2, C_x\}$ es cualquier conjunto de x “colores” cualquier función $f: V \rightarrow C$ es una coloración de la grafica G con x colores (o utilizando los colores de C). Para cada vértice v , $f(v)$ es el color de v . Como por lo general se muestra una imagen de una grafica, se piensa la coloración de manera intuitiva pintando cada vértice con un color de C , sin embargo los problemas de coloración de graficas tienen varias aplicaciones practicas donde “color” puede tener casi cualquier significado por ejemplo si la grafica representa una red de ciudades conectadas entre sí, cada ciudad puede marcarse con el nombre de la linea aérea con mas vuelos hacia y desde esa ciudad. En este caso, los vértices son las ciudades y los colores son los nombres de las líneas aéreas.

Una coloración es propia si cualesquiera dos vértices adyacentes tienen colores diferentes.

La coloración de un mapa es una forma de colorear cada región (país, estado condado, provincia etc.) de modo que no existan dos regiones distintas que comparten una frontera con el mismo color. El problema de coloración de un mapa consiste en determinar el menor número de colores que puede utilizarse. Se puede ver este caso como el problema de coloración propia de una grafica, de la siguiente manera

Dado un mapa M , se construye una grifa G_M con un vértice para cada región y una arista que conecta cualesquiera dos vértices cuyas regiones correspondientes comparten una frontera común. Entonces, la coloración propia de G_M corresponde exactamente a la coloración de M .



• Aristas de corte y vértices de corte (R, E, RP)

Definición.- La división, o separación, de un vértice en un grafo da lugar a un nuevo grafo con más componentes, ese vértice se denomina **vértice de corte** o punto de articulación.

Se llaman vértices los círculos o puntos de un grafo o árbol y se les llama aristas a las líneas que unen un vértice con otro vértice.

D. Teoría matemática de la computación

I. Autómatas y lenguajes formales

1. Máquinas finitas

La física de los sistemas complejos puede suponer el fin del mecanicismo ingenuo de Newton y Descartes y la aceptación dentro del alcance del análisis científico de aquello que Leibnitz llamaba "las máquinas infinitas". En el mundo mecanicista de Newton y Descartes sólo caben las máquinas finitas, aquéllas en las que cada parte tiene una función simple y precisa. Los sistemas complejos, por el contrario, son máquinas en donde cada parte no explica nada: el vagar de una sola hormiga alrededor del hormiguero no nos dice nada acerca de la compleja estrategia que éste ha puesto en marcha para alcanzar una fuente de comida, ni la actividad eléctrica de una neurona nos dice nada acerca de los pensamientos y los recuerdos que en ese preciso instante están recorriendo la mente de un gato o un ser humano. No hay funciones sino sólo destellos que forman algo coherente pero de forma ciega, un flujo de posibilidades de funciones, constante e impredecible, adaptándose permanentemente al entorno y al propio estado del sistema.

- **Teoremas de equivalencia entre lenguajes producidos por gramáticas y lenguajes reconocidos por autómatas (R, E, RP)**

- **Jerarquización de autómatas: finitos, autómatas de pila y equivalencias de autómatas (R, E, RP)**

AUTÓMATA FINITO.

Modelo de computación muy restringido, sin embargo tiene una gran aplicación en reconocimiento de patrones.

Representa la maquina más simple, y no contiene MEMORIA AUXILIAR. El AF, consiste de una cinta de entrada y una unidad de control.

Funcionamiento: Un AF recibe su entrada como una cadena de caracteres; dicha cadena le es entregada en la cinta de entrada. El AF no genera ninguna salida, excepto alguna SEÑAL que indique que la cadena de caracteres ha sido aceptada.

Un AF es considerado un dispositivo reconocedor de lenguaje. El lenguaje aceptado por un AF es el conjunto de cadenas que acepta.

Para describir formalmente el funcionamiento de un AF sobre una cierta cadena, debemos extender la definición de nuestra función de transición S, para que se aplique a un estado, y a una cadena de caracteres, en lugar de a un solo símbolo. Definimos así, la función: $K \times \Sigma^* \rightarrow K$, la intención es que $(q,w) \mapsto q'$ represente el estado del AF después de haber leído w, a partir del estado q.

Definición formalmente:

- 1) $(q, \epsilon) = q$
- 2) $(\forall w \in String)(\forall a \in \Sigma) (q, wa) = ((q, w), a)$

Así:(1) afirma que sin leer un símbolo, el AF no puede cambiar de estado.

Definición: Un string x se dice que es aceptado por un autómata finito M(K, Σ, q_0, F) si $S(q_0, x) = p \in F$.

Definición: El lenguaje aceptado por M, designado $L(M)$, es el conjunto $\{x | S(q_0, x) \in F\}$.

Definición: Un lenguaje es un conjunto regular(o regular) si consiste en el conjunto aceptado por algún autómata finito.

- ✓ **Autómata Finito Deterministico:** Es un autómata finito, y es deterministico cuando su operación esta completamente determinada por su entrada.

Autómata Finito Deterministico: Es un quíntuplo: $M=(K, \Sigma, S, s_0, F)$.

Donde:

- K es un conjunto finito de estados (identificadores).
- Σ es un alfabeto finito (entrada).
- $s_0 \in K$ es el estado inicial.
- $F \subseteq K$ es el conjunto de estados finales.
- $S: K \times \Sigma \rightarrow K$, es una función de transición de estados.

Una *configuración* es la situación en que se encuentra la máquina en un momento dado.

Definición. $[q_1, w_1] \xrightarrow{M} [q_2, w_2]$ ssi $w_1 = s$ w_2 para un $s \in S$, y existe una transición en M tal que $d(q_1, s) = q_2$

Definición. Una palabra $w \in S^*$ es aceptada por una máquina $M=(K, \Sigma, d, s_0, F)$ si existe un estado $q \in F$ tal que $[s, w] \xrightarrow{M^*} [q, e]$

Definición. Un cálculo en una máquina M es una secuencia de configuraciones c_1, c_2, \dots, c_n tales que $c_i \xrightarrow{M} c_{i+1}$.

Teorema. Dados una palabra $w \in S^*$ y una máquina M , sólo hay un cálculo $[s, w] \xrightarrow{M} \dots \xrightarrow{M} [q, e]$.

Definición. Dos autómatas M_1 y M_2 son equivalentes, $M_1 \equiv M_2$, cuando aceptan exactamente el mismo lenguaje.

Definición. Dos estados son compatibles si ambos son finales o ninguno es final.

Definición. Dos estados q_1 y q_2 son equivalentes, $q_1 \equiv q_2$, si al intercambiarlos en cualquier configuración, no se altera la aceptación o rechazo de toda palabra.

- ✓ **Autómata Finito No Determinístico (AFN):**

Definición. Un AFN es un quíntuplo (K, Σ, D, s_0, F) donde K, Σ, s_0, F tienen el mismo significado que para el caso de los AFD, y D , llamada la relación de transición, es un subconjunto finito de $K \times \Sigma^* \times K$.

Definición. Una palabra w es aceptada por AFN si existe una trayectoria en su diagrama de estados, que parte del estado inicial y llega a un estado final, tal que la concatenación de las etiquetas de las flechas es igual a w .

Definición. La cerradura al vacío de cerr-e (q) de un estado q es el conjunto más pequeño que contiene al estado q , a todo estado r y una transición $(p, e, r) \in D$, con cerr-e (q).

Definición. Para todo AFN existe algún AFD tal que $L(N)=L(D)$.

Modificando el modelo del AF para que acepte cero, una o más transiciones para un mismo símbolo de entrada, obtenemos el AFN.

Una secuencia de entrada a_1, a_2, \dots, a_n , es aceptada por un AFN si existe una secuencia de transiciones correspondientes a la secuencia de entrada, que conduce del estado inicial (a un estado final).

Como se vera mas adelante, cualquier conjunto aceptado por un autómata no determinístico puede ser aceptado por un AFN.

Un AFN se denota formalmente por un quíntuplo: (K, Σ, S, q_0, F) donde:

K, Σ, q_0, F son estados, alfabeto, estado inicial y estados finales y tienen el mismo significado que en el AF pero $S: K \times \Sigma \rightarrow 2^k$

AUTÓMATA DE PILA (PUSHDOWN):

Un autómata de pila (APN) es una máquina de estados finito al que se le añade una memoria externa en forma de pila (stack), es decir que sólo acepta operaciones meter (push) y sacar (pop).

Definición: Un autómata de pila es un autómata finito más una pila. Se definen por la tupla $M = (Q; P; \Sigma; \delta; q_0; Z_0; F)$

donde Q, P, Σ, q_0 y F se definen igual que en un AFN. G es un alfabeto de los caracteres que pueden introducirse a la pila, y Z_0 es el símbolo inicial en la misma, generalmente λ (significa pila vacía), pero en ocasiones es útil usar otro símbolo. La función de transición es definida como:

$$\delta : Q \times (S \times \{1\}) \times (G \times \{1\}) \rightarrow [Q \times (G \times \{1\})]^*$$

es decir que el cambio de estado ya no sólo depende del estado y del símbolo en la entrada, sino además del contenido de la pila, específicamente del símbolo en el tope de la pila. El contenido de la pila puede cambiar después de cada transición.

Para clarificar a la función de transición, analicemos la parte por parte:

$Q \times (S \times \{1\}) \times (G \times \{1\})$: Esta nos dice que los elementos del dominio son tercias (estado, alfabeto, pila)

$[Q \times (G \times \{1\})]^*$: Indica que el codominio son varios pares del tipo (estado, pila)

Ejemplo

$(q_i, a, A) = \{[q_j, B]\}$ Para realizar esta transición es necesario estar en el estado q_i , estar analizando una a en la cadena de entrada y tener en el tope de la pila una A . Al realizarse la transición pasamos del estado q_i al q_j , se procesa el símbolo a , sacamos A de la pila y metemos B .

Transición $(q_i, a, A) = \{(q_j, B)\}$

En el diagrama de transición, A/B significa que se reemplaza una A en el tope de la pila por una B . El significado de λ , es de cuidado especial, ya que en ocasiones puede significar "pila vacía", pero más generalmente significa "no importa lo que hay en la pila". Existen tres transiciones especiales que generan una acción en particular, estas son:

- $(q_i, \lambda, A) = [q_i, \lambda]$ En esta transición la única acción que ocurre es desapilar A .
- $(q_i, \lambda, \lambda) = [q_i, \lambda]$ La única acción que ocurre es apilar A .
- $(q_i, a, \lambda) = [q_i, a]$ Equivale a una transición de AF normal. (como si no tuviera pila)

Clasificación de Autómatas Push Down

1. Autómatas a pila embebidos
 2. Autómatas a pila embebidos ascendentes
 3. Autómatas lógicos a pila restringidos
 4. Autómatas lineales de índices
 5. Autómatas con dos pilas
- ✓ **Autómatas a pila embebidos:** En los cuales la estructura principal de almacenamiento la constituye una pila de pilas. Junto a la definición clásica se presenta una nueva formulación en la cual se elimina el control de estado finito y se simplifica la forma de las transiciones al tiempo que se mantiene la potencia expresiva. Esta nueva formulación permite diseñar una técnica de tabulación para la ejecución eficiente de los diversos esquemas de compilación para gramáticas de adjunción de árboles y gramáticas lineales de índices.
- ✓ **Autómatas a pila embebidos ascendentes:** Constituyen los autómatas a pila embebidos ascendentes. En este capítulo se realiza una definición formal de los mismos, algo que no se había logrado hasta el momento. La eliminación del control de estado finito permite simplificar la forma de las transiciones, lo cual facilita la definición de una técnica de tabulación para este modelo de autómata.
- ✓ **Autómatas lógicos a pila restringidos:** Las gramáticas lineales de índices constituyen un tipo específico de gramáticas de cláusulas definidas en el cual los predicados tienen un único argumento en forma de pila de índices. Aprovechamos esta característica para definir una versión restringida de los autómatas lógicos a pila adecuada al tratamiento de este tipo de gramáticas y de las gramáticas de adjunción de árboles. Dependiendo de la forma de las transiciones permitidas, podemos distinguir tres tipos diferentes de autómata, uno que permite el análisis ascendente de los índices o adjunciones, otro que permite el análisis descendente y otro que permite estrategias mixtas. En los dos últimos casos es preciso establecer restricciones en la combinación de las transiciones para garantizar que dichos autómatas aceptan exactamente la clase de los lenguajes de adjunción de árboles. Se presentan esquemas de compilación y técnicas de tabulación para los tres tipos de autómata.
- ✓ **Autómatas lineales de índices:** Los autómatas lineales de índices, que utilizan la misma estructura de almacenamiento que los autómatas lógicos a pila restringidos pero con un juego diferente de transiciones. Distinguimos tres tipos diferentes de autómata: los autómatas lineales de índices orientados a la derecha para estrategias en las cuales las pilas de índices se evalúan de modo ascendente, los autómatas lineales de índices orientados a la izquierda en los cuales las pilas de se evalúan de modo descendente y los autómatas lineales de índices fuertemente dirigidos que permiten definir estrategias mixtas de análisis para el tratamiento de las pilas de índices.
- ✓ **Autómatas con dos pilas:** Se opta por un modelo de autómata con una nueva estructura de almacenamiento. Se preserva la pila de los autómatas a pila tradicionales, a la que acompaña una pila auxiliar cuyo contenido restringe el conjunto de transiciones aplicables es un momento dado. Los autómatas con dos pilas fuertemente dirigidos permiten definir esquemas de compilación arbitrarios para gramáticas de adjunción de árboles y gramáticas lineales de índices. Por su parte, los autómatas con dos pilas ascendentes sólo permiten describir esquemas de compilación que incorporan estrategias ascendentes en lo referente al tratamiento de las adjunciones y de las pilas de índices. Se presentan las técnicas de tabulación que permiten una ejecución eficiente de ambos modelos de autómata.

2. Lenguajes formales

En Matemáticas, Lógica, y las Ciencias Computacionales, un **lenguaje formal** es un conjunto de palabras (cadenas de caracteres) de longitud finita formadas a partir de un alfabeto (conjunto de caracteres) finito.

Informalmente, el término *lenguaje formal* se utiliza en muchos contextos (en las ciencias, en derecho, etc.) para referirse a un modo de expresión más cuidadoso y preciso que el habla cotidiana. Hasta finales de la década de 1990, el consenso general era que un lenguaje formal, en el sentido que trata este artículo, era en cierto modo la versión "límite" de este uso antes mencionado: un lenguaje tan formalizado que podía ser usado en forma escrita para describir métodos computacionales. Sin embargo, hoy en día, el punto de vista de que la naturaleza esencial de los lenguajes naturales (sin importar su grado de "formalidad" en el sentido informal antes descrito) difiere de manera importante de aquella de los verdaderos lenguajes formales (en el sentido estricto de este artículo) gana cada vez más adeptos.

Un posible alfabeto sería, digamos, $\{a, b\}$, y una cadena cualquiera sobre este alfabeto sería, por ejemplo, **ababba**. Un lenguaje sobre este alfabeto, que incluyera esta cadena, sería: el conjunto de todas cadenas que contienen el mismo número de símbolos **a** que **b**.

La **palabra vacía** (esto es, la cadena de longitud cero) es permitida y frecuentemente denotada mediante \emptyset o λ . Mientras que el alfabeto es un conjunto finito y cada palabra tiene una longitud también finita, un lenguaje puede bien incluir un número infinito de palabras.

Algunos ejemplos varios de lenguajes formales:

- ✓ el conjunto de todas las palabras sobre $\{a, b\}$
- ✓ el conjunto $\{a^n \mid n \text{ es un número primo}\}$
- ✓ el conjunto de todos los programas sintácticamente válidos en un determinado lenguaje de programación
- ✓ el conjunto de entradas para las cuales una particular máquina de Turing se detiene.

Los lenguajes formales pueden ser especificados en una amplia variedad de maneras, como:

- ✓ cadenas producidas por una gramática formal (ver Jerarquía de Chomsky)
- ✓ cadenas producidas por una expresión regular
- ✓ cadenas aceptadas por un autómata, tal como una máquina de Turing

Varias operaciones pueden ser utilizadas para producir nuevos lenguajes a partir de otros dados. Supóngase que L_1 y L_2 son lenguajes sobre un alfabeto común. Entonces:

- ✓ la **concatenación** L_1L_2 consiste de todas aquellas palabras de la forma vw donde v es una palabra de L_1 y w es una palabra de L_2
- ✓ la **intersección** $L_1 \cap L_2$ consiste en todas aquellas palabras que están contenidas tanto en L_1 como en L_2
- ✓ la **unión** $L_1 \cup L_2$ consiste en todas aquellas palabras que están contenidas ya sea en L_1 o en L_2
- ✓ el **complemento** $\sim L_1$ consiste en todas aquellas palabras producibles sobre el alfabeto de L_1 que no están ya contenidas en L_1
- ✓ el **cociente** L_1/L_2 consiste de todas aquellas palabras v para las cuales existe una palabra w en L_2 tales que vw se encuentra en L_1

- ✓ la *estrella* L_1^* consiste de todas aquellas palabras que pueden ser escritas de la forma $W_1 W_2 \dots W_n$ donde todo W_i se encuentra en L_1 y $n \geq 0$. (Nótese que esta definición incluye a & epsilon en cualquier L^*)
- ✓ la *intercalación* $L_1^* L_2$ consiste de todas aquellas palabras que pueden ser escritas de la forma $v_1 w_1 v_2 w_2 \dots v_n w_n$ son palabras tales que la concatenación $v_1 \dots v_n$ está en L_1 , y la concatenación $w_1 \dots w_n$ está en L_2

Una pregunta que se hace típicamente sobre un determinado lenguaje formal L es el qué tan difícil es decidir si incluye o no una determinada palabra v . Este tema es del dominio de la teoría de la computabilidad y la teoría de la complejidad computacional.

Por contraposición al lenguaje propio de los seres vivos y en especial el lenguaje humano, considerados lenguajes naturales, se denomina lenguaje formal a los lenguajes "artificiales" propios de las matemáticas o la informática, los lenguajes artificiales son llamados lenguajes formales (incluyendo lenguajes de programación). Sin embargo, el lenguaje humano tiene una característica que no se encuentra en los lenguajes de programación: la diversidad.

En 1956, Noam Chomsky creó la Jerarquía de Chomsky para organizar los distintos tipos de lenguaje formal.

• Gramáticas formales: definiciones, operaciones, tipos de lenguajes, ambigüedad, equivalencia, la jerarquización de Chomsky (R, E)

Gramática formal.

La gramática de un lenguaje es el conjunto de reglas capaces de generar todas las posibilidades combinatorias de ese lenguaje, ya sea éste un lenguaje formal o un lenguaje natural.

La expresión "gramática formal" tiene dos sentidos:

- (a) gramática de un *lenguaje formal*.
- (b) *descripción formal* de la gramática de un lenguaje natural.

En cuanto a (b), la descripción formal de la gramática de una lengua es la que **explicita las reglas combinatorias mentales a partir de fórmulas lógico-matemáticas**.

Estas reglas combinatorias que utiliza el lenguaje natural reciben el nombre de sintaxis, y son inconscientes.

En cuanto a (a), 'gramática formal' se utiliza en analogía con el concepto de gramática, usado tradicionalmente para los lenguajes naturales. Por tanto, una gramática formal es el conjunto de reglas combinatorias de un lenguaje formal.

Hay distintos tipos de gramáticas formales que generan lenguajes formales (la Jerarquía de Chomsky).

Imaginemos una gramática con estas dos reglas:

1. A $\rightarrow bAc$
2. A $\rightarrow de$

La idea es sustituir el símbolo inicial de la izquierda por otros símbolos aplicando las reglas. El lenguaje al cual representa esta gramática es el conjunto de cadenas de símbolos que pueden ser generados de esta manera: en este caso, por ejemplo:

A --> bAc --> bbAcc --> bbbAccc ---> bbbdeccc.

El elemento en mayúsculas es el símbolo inicial. Los elementos en minúsculas son símbolos terminales. Las cadenas de la lengua son aquellas que solo contienen elementos terminales, como por ejemplo: bbbdeccc, de, bdec, ... Estas serían tres posibles realizaciones del lenguaje cuya gramática hemos definido con dos reglas.

Jerarquía de Chomsky.

La jerarquía de Chomsky es una clasificación jerárquica de distintos tipos de gramáticas formales que generan lenguajes formales. Esta jerarquía fue descrita por Noam Chomsky en 1956.

Gramática	Lenguaje	Autómata	Normas de producción
Tipo-0	Recursivamente enumerable (LRE)	Máquina de Turing (MT)	Sin restricciones
Tipo-1	dependiente del contexto(LSC)	Autómatas Linealmente Acotados	$A \rightarrow$
Tipo-2	independiente del contexto (LLC)	Autómata a Pila	$A \rightarrow$
Tipo-3	regular (RL)	Autómata Finito	$A \rightarrow aB$ $A \rightarrow a$

Tabla de contenidos

- 1 Lenguajes Recursivamente Enumerables (de tipo 0)
- 2 Lenguajes Dependientes del Contexto (sensibles al contexto, de tipo 1)
- 3 Lenguajes Independientes del Contexto (de contexto libre, de tipo 2)
- 4 Lenguajes Regulares (de tipo 3)

II. Sistemas formales

1. Máquinas de Turing

• Concepto y límites de la computabilidad (R, E, RP)

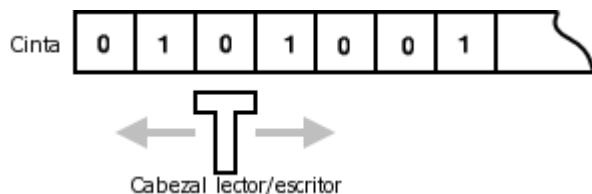
Prácticamente todas las áreas de las Ciencias Computacionales tratan, en mayor o menor grado con complejidad computacional. El tema es muy común entre expertos del área, sobre todo entre aquéllos relacionados con NP-completeness y entre quienes buscan algoritmos eficientes para diversos problemas de aplicación. La importancia de la complejidad computacional estriba en que se ha convertido en una forma de clasificar buenos y malos algoritmos y de clasificar problemas computacionales como fáciles y difíciles. Aquí presentaremos una introducción a este tema.

- **Máquinas de Turing: modelos de computabilidad, problemas indecidibles (*The Halting Problem*) (R, E, RP)**

La máquina de Turing es un modelo computacional creado por Alan Turing con el cual él afirmaba que se podía realizar cualquier cómputo.

La máquina de Turing, como modelo matemático, consta de un cabezal lector/escritor y una cinta infinita en la que el cabezal lee el contenido, borra el contenido anterior y escribe un nuevo valor. Las operaciones que se pueden realizar en esta máquina se limitan a:

- avanzar el cabezal lector/escritor para la derecha;
- avanzar el cabezal lector/escritor para la izquierda.



El cómputo es determinado a partir de una tabla de estados de la forma:
 (estado,valor) → (\nuevo estado, \nuevo valor, dirección)

Esta tabla toma como parámetros el estado actual de la máquina y el carácter leído de la cinta, dando la dirección para mover el cabezal, el nuevo estado de la máquina y el valor a ser escrito en la cinta.

Con este aparato extremadamente sencillo es posible realizar cualquier cómputo que un computador digital sea capaz de realizar.

Mediante este modelo teórico y el análisis de complejidad de algoritmos, fue posible la categorización de problemas computacionales de acuerdo a su comportamiento, apareciendo así, el conjunto de problemas denominados P y NP, cuyas soluciones en tiempo polinomial son encontradas según el determinismo y no determinismo respectivamente de la máquina de Turing.

De hecho, se puede probar matemáticamente que para cualquier programa de computadora es posible crear una máquina de Turing equivalente. Esta prueba resulta de la Tesis de Church-Turing, formulada por Alan Turing y Alonzo Church, de forma independiente a mediados del siglo XX.

III. Computabilidad

1. Complejidad

- **Problemas clásicos: satisfactibilidad, triangulación, agente viajero y mochila (R)**

Algoritmos y Complejidad

Cada vez nos sorprendemos más de la velocidad de las computadoras de sus logros. Han permitido la solución de problemas muy complejos que serían ciencia ficción hace apenas unos cuantos años: Realidad virtual, simulación de procesos gigantescos, etc.

Pese a lo que pueda pensarse, las computadoras no realizan ninguna actividad que pudiera calificarse como de una gran inteligencia. Las computadoras pueden únicamente llevar a cabo algoritmos; esto es, secuencias de instrucciones precisas y universalmente entendibles que solucionen cualquier instancia de problemas computacionales rigurosamente definidos.

A pesar de que se ha llegado a logros sorprendentes como derrotar a un campeón de ajedrez Gary Kasparov en 1997 por una computadora Deep Blue de IBM, es difícil atribuirle características de inteligencia, cuando se tuvieron miles de horas hombre programando estrategias que derrotaran a Kasparov. Nótese que la inteligencia de Kasparov requirió de cientos de cerebros pensando como derrotarlo. ¿Cuántas horas-hombre hubiera requerido cada uno de los que programaron a Deep Blue? ¿Cuál sería entonces la inteligencia relativa de estos hombres comparados con Kasparov?

Quienes programaron a Deep Blue pusieron una serie de pasos a realizar para cada posible ataque de Kasparov. Es decir en Deep Blue pusieron algoritmos.

¿Pero qué son los algoritmos?

El concepto intuitivo de algoritmo, lo tenemos prácticamente todos:

Un algoritmo es una serie finita de pasos para resolver un problema.

Hay que hacer énfasis en dos aspectos para que un algoritmo exista:

1. El número de pasos debe ser finito. De esta manera el algoritmo debe terminar en un tiempo finito con la solución del problema,
2. El algoritmo debe ser capaz de determinar la solución del problema.

Alan Turing, en 1936 desarrolló su Máquina de Turing (la cual se cubre en los cursos denominados Teoría de la Computación o Teoría de Automátas), estableciendo que cualquier algoritmo puede ser representado por ella.

Turing mostró también que existen problemas matemáticos bien definidos para los cuales no hay un algoritmo. Hay muchos ejemplos de problemas para los cuales no existe un algoritmo. Un problema de este tipo es el llamado problema de paro (halting problem):

Dado un programa de computadora con sus entradas, ¿parará este alguna vez?

Turing probó que no hay un algoritmo que pueda resolver correctamente todas las instancias de este problema.

Alguien podría pensar en encontrar algunos métodos para detectar patrones que permitan examinar el programa para cualquier entrada. Sin embargo, siempre habrá sutilezas que escapen al análisis correspondiente.

Alguna persona más suspicaz, podría proponer simplemente correr el programa y reportar éxito si se alcanza una declaración de fin. Desgraciadamente, este esquema no garantiza por sí mismo un paro y en consecuencia el problema no puede ser resuelto con los pasos propuestos. Como consecuencia de acuerdo con la definición anterior, ese último procedimiento no es un algoritmo, pues no se llega a una solución.

Muchas veces aunque exista un algoritmo que pueda solucionar correctamente cualquier instancia de un problema dado, no siempre dicho algoritmo es satisfactorio porque puede requerir de tiempos exageradamente excesivos para llegar a la solución.

Satisfactibilidad

El problema de Satisfactibilidad (SAT) consiste en encontrar alguna asignación de valores de verdad que logren satisfacer una colección de cláusulas en forma normal conjuntiva. En otras palabras, en el SAT cada restricción se expresa mediante una ecuación booleana (aquella compuesta por variables que pueden asumir únicamente los valores de V o F) llamada cláusula proposicional, y el conjunto de cláusulas establece una fórmula booleana. De esta forma, el problema consiste en determinar una asignación de valores de verdad para las variables que satisfaga la fórmula proposicional (que la haga verdadera), esto es, que satisfaga todas las restricciones. Esto ocasiona que el tiempo de ejecución del algoritmo crezca en forma exponencial. Este fue el primer problema computacional que se demostró como NP-Completo, aún cuando se restringían la cantidad de variables por cláusula a 3.

Métodos para resolver el problema SAT: Los métodos para resolver el problema SAT suelen clasificarse en dos grandes categorías: Métodos Completos y Métodos Incompletos.

- ✓ **Métodos Completos:** Los métodos completos tienen la ventaja que teóricamente son capaces de encontrar todas las soluciones de cualquier problema booleano, o en su defecto, probar que no existe una solución. Pero la naturaleza combinatoria del problema SAT convierte a estos métodos en imprácticos cuanto el tamaño del problema se incrementa.

Algunos métodos completos son: El Teorema de Herbrand, El método de Davis y Putnam, El Principio de Resolución, La Resolución Semántica, El algoritmo de dos fases para SAT y MAX-SAT, Programación lineal entera.

- ✓ **Métodos Incompletos:** Los métodos incompletos utilizan heurísticas eficientes para tratar de limitar el problema del aumento de complejidad lo más posible. Estos métodos no pueden demostrar insatisfactibilidad, pero son muchos más veloces que los métodos completos, ya que estos solo buscan en ciertas áreas del problema en busca de satisfactibilidad, y si no la encuentran, se van a otra área, y así sucesivamente

Algunos métodos incompletos son: GSAT, RWS-GSAT, Recorrido Simulado SAT, TSAT, GSAT de dos fases, MASK, NNSAT, TSAT, NNSAT

Agente viajero.

Dado un grafo de N nodos y A arcos, determinar la ruta más corta que parte de un nodo predeterminado N_i y visite todos los nodos solamente una vez y regrese al nodo de partida.

O su versión como un problema de decisión: Existe un recorrido más corto que un valor K que parte de un nodo inicial N_i , recorra visite todos los otros $N-1$ nodos una sola vez y regrese al nodo inicial N_i .

Este problema parece ser muy simple y puede ser que lo sea para ciertos tamaños de N y A .

Sin embargo existen instancias de este problema, con valores ni siquiera demasiado grandes de N y A , para los cuales encontrar la solución es terriblemente ardío. Tan solo basta pensar en

una red de computadoras de unos 500 nodos con cierta topología para que sea muy difícil determinar la ruta más corta de un mensaje enviado desde un nodo i determinado y que debe visitar todos los nodos solamente una vez, trayendo de regreso el mensaje enviado y la traza de los nodos visitados.

Para cualquier instancia del problema del agente viajero uno podría pensar en seleccionar todos y cada uno de los diferentes recorridos posibles y luego compararlos para encontrar el de menor longitud. Sin embargo, la ejecución de este algoritmo en una computadora requerirá de aproximadamente $n!$ pasos, lo que para instancias relativamente pequeñas lo vuelve inútil.

Para propósitos de comparación del rendimiento de diferentes algoritmos, se requiere de una medida de rendimiento de algoritmos; la medida más ampliamente aceptada es el tiempo que el algoritmo en cuestión consume antes de producir la solución final. Esta cantidad de tiempo puede variar de una computadora a otra, debido a diferencias en su velocidad de procesamiento y en la forma como el programa es traducido en instrucciones de máquina. En el análisis de algoritmos, el tiempo siempre es expresado en términos del número de pasos elementales (operaciones aritméticas, comparaciones, lecturas, escrituras, etc.) requeridos para la ejecución del algoritmo sobre una computadora hipotética. Dichos pasos elementales tienen una duración unitaria de tiempo. De esta forma, el tiempo requerido para la ejecución de un algoritmo es establecido como el número de pasos elementales o instrucciones que efectúa antes de producir la solución final.

El número de pasos requeridos por un algoritmo no es siempre el mismo para todas las entradas y frecuentemente es difícil obtener una fórmula precisa que establezca la función de tiempo. Es por ello que se consideran tres medidas del tiempo de ejecución del algoritmo, las cuales dependen del tamaño de la entrada n :

- a. El mínimo tiempo necesario (mejor caso),
- b. El máximo tiempo necesario (peor caso) y,
- c. El tiempo promedio necesario (caso promedio).

De esta forma la complejidad de un algoritmo es función del tamaño de la entrada (por ejemplo $10n^3$, 2^n , $n \log n$).

El interés principal en el comportamiento de un algoritmo es para entradas grandes, pues estas establecerán los límites de aplicabilidad de este. Por lo tanto, generalmente se tiene poco interés en el valor exacto del tiempo requerido para la ejecución de un algoritmo. Siendo el interés principal la razón de crecimiento del mejor caso.

ARQUITECTURA DE COMPUTADORAS

A. Física

I. Temas de mecánica, óptica y física moderna

1. Mecánica

Conjunto de reglas o principios que se aplican al análisis de todo tipo de movimiento. Estudio de las relaciones entre la fuerza, materia y movimiento.

• Dinámica (R)

Estudio de la relación entre el movimiento de un cuerpo y las causas dicho movimiento. Relación entre el movimiento y sus causas.

La dinámica investiga las leyes de los cuerpos en movimiento, lo mismo que sus causas. La dinámica estudia las causas del movimiento de los cuerpos, a diferencia de la cinemática que estudia los movimientos de los cuerpos sin preocuparse de lo que induce a este movimiento.

• Cinemática (R)

La cinemática estudia el movimiento de los cuerpos sin referencia a las fuerzas que actúan sobre ellos. La cinemática estudia la descripción matemática de la trayectoria de un objeto. Se preocupa del movimiento, pero sin atenerse a las causas que lo producen.

Parte de la mecánica que describe el movimiento. Estudia el movimiento de los cuerpos sin atender a las causas que lo producen.

- ✓ **Movimiento general:** Un cuerpo realiza dicho movimiento general cuando, durante dicho movimiento, todo segmento rectilíneo del mismo tanto se traslada como gira, de modo que todas las partículas de dicho movimiento describen trayectorias tridimensionales, no planas ni rectas
- ✓ **Movimientos de traslación:** Son casos particulares del movimiento general. Un cuerpo rígido realiza un movimiento de traslación cuando todo segmento rectilíneo del cuerpo permanece paralelo a su orientación original, durante su movimiento.
- ✓ **Movimientos de rotación:** Son un caso particular del movimiento general. Un cuerpo rígido realiza un movimiento de rotación cuando, durante su movimiento, al menos algunos segmentos rectilíneos del cuerpo van cambiando su orientación con respecto a la que tenían originalmente.

Se presentan los siguientes casos:

1. Rotación en torno a un eje fijo
2. Rotación en torno a un punto fijo.

- ✓ **Movimiento plano general:** Es un caso particular del movimiento general. Se presenta cuando las partículas del cuerpo describen trayectorias planas mientras este gira y se traslada. Los movimientos de traslación (ya sea rectilínea o curvilínea) y de rotación en torno a un eje fijo son casos particulares del movimiento plano general.
- ✓ **Movimiento de rodadura sin deslizar sobre una superficie plana:** Caso particular del movimiento general, realizado por cuerpos que se mueven sobre superficies planas, de modo que establecen un contacto continuo con las mismas, mientras ruedan y se trasladan sobre ellas.

• Leyes de Newton (R)

Las leyes de Newton son las leyes del movimiento.

Primera ley de Newton (ley de la inercia).

"Todo cuerpo tiende a mantener su estado de reposo o de movimiento mientras no exista una fuerza externa que modifique dicho estado"

Un cuerpo sobre el que no actúa una fuerza beta se mueve con velocidad constante (que puede ser cero) y cero aceleración.

Inercia: Tendencia de un cuerpo a seguir moviéndose una vez puesto en movimiento.

Segunda ley de Newton (ley del movimiento).

"Toda fuerza (F) aplicada a un cuerpo de masa (m), produce una aceleración (a) directamente proporcional a la fuerza e inversamente proporcional a la masa del cuerpo".

La tasa de cambio de momentum de una partícula con respecto al tiempo es igual a la fuerza que actúa sobre la partícula.

$$F=ma$$

Tercera ley de Newton (ley de acción y reacción).

"A toda Fuerza de acción le corresponde una de reacción de la misma intensidad, pero de sentido contrario".

Si el cuerpo A ejerce una fuerza sobre el cuerpo B (una "acción"), entonces B ejerce una fuerza sobre A (una "reacción"). Estas fuerzas tienen la misma magnitud pero dirección opuesta, y actúan sobre diferentes cuerpos.

• **Ley de la gravitación universal (R)**

La interacción gravitatoria entre dos cuerpos corresponde a una fuerza central atractiva proporcional a las masas de los cuerpos e inversamente proporcional al cuadrado de la distancia entre ellos.

Toda partícula de materia en el universo atrae a todas las demás partículas con una fuerza que es directamente proporcional a las masas de las partículas e inversamente proporcional al cuadrado de la distancia que las separa.

$$F_g = \frac{Gm_1 m_2}{r^2}$$

Donde F_g es la magnitud de la fuerza gravitatoria sobre cualquiera de las partículas, m_1 y m_2 son sus masas, r es la distancia entre ellas y G es una constante física fundamental llamada constante gravitacional (en unidades del SI $G=6.67 \times 10^{-11} \text{ N m}^2/\text{kg}^2$ o $\text{m}^3/(\text{kg s}^2)$).

2. Óptica

Rama de la física que se ocupa de la propagación y el comportamiento de la luz. Estudia los fenómenos luminosos y la visión, que incluyen el diseño de instrumentos ópticos.

• **Óptica geométrica: lentes, diafragmas, espejos, sistemas ópticos, fibra óptica (R)**

La óptica geométrica se fundamenta en la teoría de los rayos de luz, la cual considera que cualquier objeto visible emite rayos rectos de luz en cada punto de él y en todas direcciones a su alrededor. Cuando estos rayos inciden sobre otros cuerpos pueden ser absorbidos, reflejados o desviados, pero si penetran en el ojo estimularán el sentido de la vista.

La óptica geométrica estudia a la luz en forma de rayos y las propiedades de los lentes y espejos.

Lentes.

Una lente es un medio transparente limitado dos superficies curvas (por lo general esféricas o cilíndricas), aunque una puede ser plana. Por tanto, una onda incidente sufre dos refracciones al pasar por una lente.

- ✓ **Lente convexa.** Una lente convexa es más gruesa en el centro que en los extremos. La luz que atraviesa una lente convexa se desvía hacia dentro (converge). Esto hace que se forme una imagen del objeto en una pantalla situada al otro lado de la lente.
- ✓ **Lente cóncava.** Las lentes cóncavas están curvadas hacia dentro. La luz que atraviesa una lente cóncava se desvía hacia fuera (diverge). A diferencia de las lentes convexas, que producen imágenes reales, las cóncavas sólo producen imágenes virtuales, es decir, imágenes de las que parecen proceder los rayos de luz.

Fibra óptica.

Fibras transparentes con diámetro de unos cuantos micrómetros, recubiertas con un material de índice de refracción menor para mejorar la reflexión total en las paredes de la fibra. Estas se hacen de vidrio o cuarzo, aunque también se usan otros materiales, como el nailon. Un rayo que entra por un extremo se propaga por la fibra debido a reflexiones sucesivas, y sale por el otro, sin importar si la fibra es recta o curva.

Espejos.

- ✓ **Formación de imágenes en espejos planos:** conforme se deduce de las leyes de la reflexión, la imagen P' de un punto objeto P respecto de un espejo plano S' estará situada al otro lado de la superficie reflectora a igual distancia de ella que el punto objeto P . Además la línea que une el punto objeto P con su imagen P' es perpendicular al espejo. Es decir, P y P' son simétricos respecto de S ; si se repite este procedimiento de construcción para cualquier objeto punto por punto, se tiene la imagen simétrica del objeto respecto del plano del espejo.
- ✓ **Formación de imágenes en espejos esféricos:** Los espejos esféricos tienen la forma de la superficie que resulta cuando una esfera es cortada por un plano. Si la superficie reflectora está situada en la cara interior de la esfera se dice que el espejo es *cóncavo*. Si está situada en la cara exterior se denomina *convexo*.

• Reflexión óptica (R)

La reflexión luminosa es un fenómeno en virtud del cual la luz al incidir sobre la superficie de los cuerpos cambia de dirección, invirtiéndose el sentido de su propagación. De acuerdo con las características de la superficie reflectora, la reflexión luminosa puede ser regular o difusa.

- ✓ **La reflexión regular:** tiene lugar cuando la superficie es perfectamente lisa. Un espejo o una lámina metálica pulimentada reflejan ordenadamente un haz de rayos conservando la forma del haz.
- ✓ **La reflexión difusa:** se da sobre los cuerpos de superficies más o menos rugosas.

• Refracción óptica (R)

Se denomina refracción luminosa al cambio que experimenta la dirección de propagación de la luz cuando atraviesa obliquamente la superficie de separación de dos medios transparentes de distinta naturaleza.

Cuando la luz pasa de un medio con un bajo índice de refracción a un medio de mayor índice de refracción, ésta disminuye su velocidad y desvía su dirección.

• Interferencia óptica (R)

Cuando dos o más ondas coinciden en el espacio y el tiempo existe una interferencia.

Cuando dos disturbios de onda se combinan, en tal forma que los picos de una onda coinciden con los picos de la otra, las dos ondas se refuerzan para producir un disturbio mayor. Este proceso se conoce como interferencia constructiva. Por otro lado si los picos de una onda coinciden con los valles de la otra, entonces las ondas tendrán a cancelarse. Este proceso se conoce como interferencia destructiva.

• Difracción óptica (R)

La difracción se refiere al cambio de dirección que sufre la luz cuando encuentra alguna obstrucción, apertura o irregularidad en el medio. La resolución más fina que un sistema óptico puede obtener está limitada por la difracción.

• Radiación láser (R)

La característica más original del Láser es que emite luz coherente, o sea todas las porciones de la onda óptica vibran al unísono, mientras que una fuente ordinaria de luz emite luz incoherente de longitudes de onda variadas. Un haz de luz coherente puede ser enviado sobre largas distancias, con mucho menos divergencia que un haz incoherente.

II. Electricidad y magnetismo

1. Electricidad

• Corriente eléctrica (RP)

Cuando un electrón consigue liberarse de un átomo, se desplaza a través del espacio hasta que choca con un segundo átomo el cual lo acepta incorporándolo a su estructura y expulsando a uno de sus electrones originales. Este electrón a su vez, golpea a un tercer átomo, y así sucesivamente. Cada electrón individual no viaja muy lejos, pero la energía de los electrones en movimiento puede transmitirse a lo largo de la longitud del conductor.

Cuando este proceso ocurre con electrones en un conductor se le llama *electricidad o corriente eléctrica*. Otro punto de vista sería considerar a la corriente como el flujo efectivo de electrones (o simplemente *flujo eléctrico*). Si un Coulomb (6.25×10^{18} electrones) fluye pasando por un punto dado en un segundo, se dice que la corriente es de un *ampere*. El ampere es la unidad básica para la medición de la corriente eléctrica pero en, muchos casos prácticos, es una unidad demasiado grande para ser apropiada. En estos casos, es más simple utilizar *miliampères*, los cuales son milésimos de ampere, o bien *microampères*, que son millonésimos de ampere (o bien un milésimo de miliampere).

La corriente eléctrica se expresa en coulomb/segundo o $C\ s^{-1}$, unidad conocida como ampere. Un ampere es la intensidad de una corriente eléctrica cuando una carga de un coulomb pasa por una sección del material cada segundo.

$$I = \frac{Q}{t} \quad \text{donde: } I = \text{corriente en amperes (A)} \\ Q = \text{carga en coulombs (C)} \\ t = \text{tiempo en segundos (s)}$$

• Potencia eléctrica (RP)

Si se desea determinar la cantidad de potencia de trabajo que se está realizando en un circuito, se necesita considerar tanto al voltaje como a la corriente. La energía total consumida se llama potencia, y se mide en watts. Un watt de potencia se consume cuando un volt impulsa un ampere a través de un circuito.

P es la potencia, V es el voltaje e I es la corriente, de modo que la potencia en watts es igual al voltaje en Volts multiplicado por la corriente en amperes.

$$P = VI = I^2R = \frac{V^2}{R}$$

• Resistencia (RP)

En cualquier sistema eléctrico la presión es el voltaje aplicado, y el resultado es el flujo de la carga o corriente. La resistencia del sistema controla el nivel de la corriente resultante. Mientras mayor es la resistencia, menor es la corriente y viceversa

La resistencia es la capacidad de los materiales para impedir el flujo de la corriente, o flujo de carga eléctrica. La resistencia de una red, un sistema o un resistor se puede medir con un instrumento llamado ohmetro.

Resistores en serie.

Se dice que un número de resistores, ($R_1, R_2, R_3, R_n, \dots$) están conectados en serie si su conexión es consecutiva extremo con extremo, de tal suerte que la misma corriente (I), en amperes, fluya a través de cada una.

$$R = R_1 + R_2 + R_3$$

Resistores en paralelo.

Se dice que los resistores están en paralelo cuando son colocados uno al lado del otro y sus extremos permanecen unidos. La misma diferencia de potencial será entonces aplicada a cada uno, pero compartirán la corriente en el circuito.

$$\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}$$

• **Ley de Ohm (RP)**

Para un conductor a temperatura constante, la relación entre la diferencia de potencial que hay entre dos puntos del conductor y la corriente eléctrica en el conductor es constante.

“La intensidad de la corriente eléctrica (I) en un circuito es directamente proporcional al voltaje (E) aplicado e inversamente proporcional a la resistencia del circuito (R)”.

$$I = \frac{E}{R}$$

donde: I= amperes (A)
E= volts (V)
R= ohms (Ω)

• **Conductividad (E, RP)**

Recíproco de la resistencia. Habilidad de los electrones para moverse libremente por el sólido bajo la acción de un campo eléctrico externo. Se simboliza con la letra G, y se mide en siemens (S).

$$G = 1/R$$

Ciertas substancias podrán ceder un electrón (o aceptar un electrón extra) más fácilmente que otras. Tales substancias (típicamente los metales) se denominan conductores, porque pueden conducir electricidad. Es decir permiten el paso de una corriente eléctrica a través de ellos. Una sustancia que presenta una fuerte atracción sobre sus electrones y que por lo tanto es resistente a liberar o aceptar electrones, es un aislador. La corriente eléctrica puede pasar a través de un aislador, pero requiere de una fuerza mayor en magnitud que para un conductor. Cualquier átomo puede estar dispuesto para ceder un electrón. Los conductores son simplemente esas substancias que proporcionan electrones sin necesidad de una fuerza externa muy grande.

• **Circuitos eléctricos simples (RP)**

El circuito eléctrico simple constituye una fuente de poder que va a proporcionar una fuerza electromotriz estableciendo diferencias de potencial a través de los varios componentes del circuito e impulsando la corriente a través de ellos.

2. Magnetismo

La interacción entre polos magnéticos iguales es de repulsión y entre polos magnéticos distintos es de atracción.

• **Fuerzas sobre conductores portadores de corriente (R)**

Fuerza magnética ejercida sobre una corriente eléctrica.

Como una corriente eléctrica es un flujo de cargas eléctricas, cuando un conductor que lleva una corriente eléctrica se coloca en un campo magnético experimenta una fuerza, perpendicular a la corriente, que es la resultante de las fuerzas magnéticas ejercida sobre cada una de las cargas en movimiento.

• **Imanes (R)**

Están hechos de material ferromagnético como el hierro o el acero por que sus características permiten establecer las líneas de flujo a través de estos con muy poca presión interna. Una vez que se imanta, sus propiedades de retención son tales que exhiben propiedades magnéticas después de que se retira el agente magnetizante.

3. Circuitos eléctricos

Un circuito eléctrico es un modelo matemático que simula el comportamiento de un sistema eléctrico real. El término se utiliza comúnmente para referirse tanto a un sistema eléctrico real como al modelo que representa.

- **Medición de corriente y voltaje (R)**

Cuando se trata de medir la corriente, primero se debe “romper” el trayecto del flujo de carga (corriente) e insertar el medidor entre las dos terminales creadas en el circuito. El instrumento para medir la corriente se llama amperímetro. El medidor se conecta de modo que la corriente entre por la terminal positiva y salga por la negativa.

El voltímetro básico es muy similar al amperímetro en su apariencia fundamental, pero las técnicas de medición son muy diferentes. El voltímetro no “rompe” el circuito sino que se coloca en un extremo del elemento para el cual se va a determinar la diferencia de potencial.

Un amperímetro es un instrumento diseñado para medir corriente; se pone en serie con el elemento de circuito cuya corriente va a medir. Un voltímetro es un instrumento diseñado para medir voltaje; se conecta en paralelo con el elemento cuyo voltaje se va a medir.

- **Métodos de análisis en el dominio de la frecuencia y del tiempo de circuitos RC, RL y RLC (RP)**

- **Funciones de transferencia (E)**

Se define como la proporción en el dominio s entre la transformada de Laplace de la salida (respuesta) y la transformada de Laplace de la entrada (fuente). En el cálculo de la función de transferencia, se restringe a circuitos donde las condiciones iniciales son cero. Si un circuito tiene múltiples fuentes independientes, es posible encontrar la función de transferencia para cada fuente y utilizar la superposición con el fin de obtener la respuesta correspondiente a todas las fuentes.

La función de transferencia es: $H(s) = \frac{Y(s)}{X(s)}$

$X(s)$

Donde: $Y(s)$ es la transformada de Laplace de la señal de salida

$X(s)$ es la transformada de Laplace de la señal de entrada

III. Electrónica

1. Principios y elementos de la electrónica

- **Circuitos con diodos y transistores (E)**

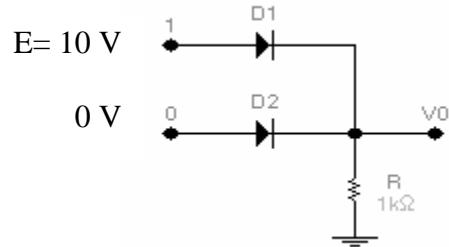
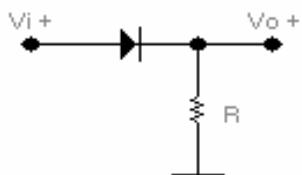
Diodo ideal.

Conducirá corriente en la dirección que define la flecha en el símbolo, y actuará como un circuito abierto en cualquier intento por establecer corriente en dirección opuesta.

El diodo es un dispositivo de dos terminales que, en una situación ideal, se comporta como un interruptor común con la condición especial de que solo puede conducir en una dirección. Tiene un estado encendido, en el que en teoría parece ser simplemente un circuito cerrado entre sus terminales, y en estado apagado, en el que sus características terminales son similares a las de un circuito abierto.

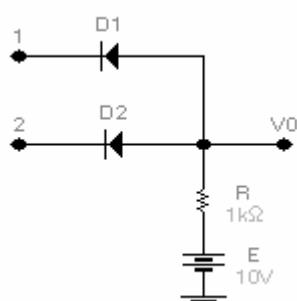


✓ Compuerta OR.



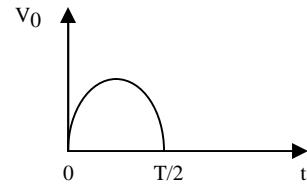
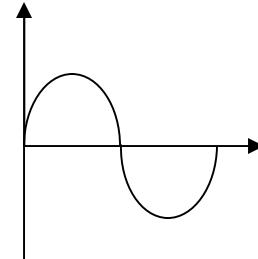
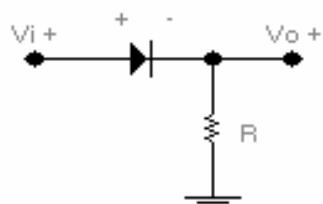
✓ Compuerta AND.

$$(1) E_1 = 10 \text{ V}$$

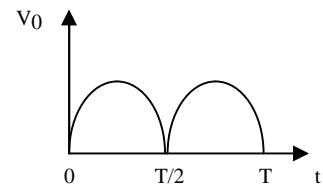
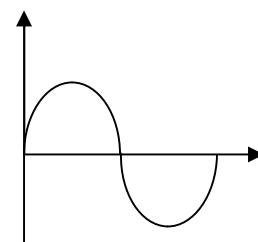
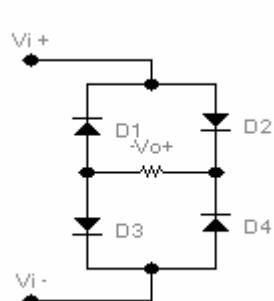


$$(0) E_2 = 0 \text{ V}$$

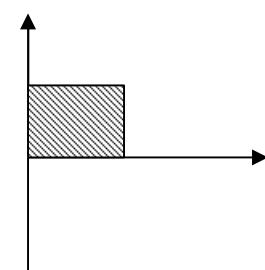
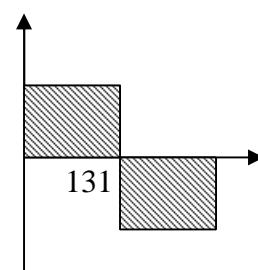
✓ Rectificador de media onda.



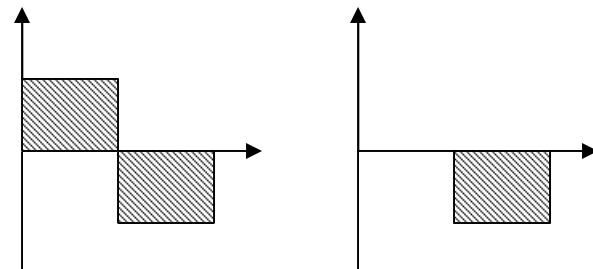
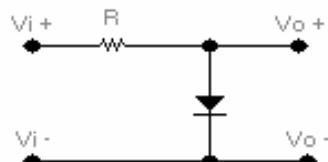
✓ Rectificador de onda completa (Puente de diodos)



✓ Recortadores.
1. En serie



2. En paralelo



Transistor.

Dispositivo semiconductor de tres capas que consiste de dos capas de material tipo *n* y una capa tipo *p* (npn) o viceversa (pnp).

Transistor pnp



Transistor npn



• Compuertas lógicas (R)

• Características e integración de las tecnologías: TTL, ECL, MOS, CMOS, I²L, MSI, LSI, VLSI (R)

TTL (Transistor Transistor Logic) *lógica transistor-transistor*. Circuito digital en el que la salida se obtiene a partir de dos transistores. Aunque la tecnología TTL constituye un método específico de diseño, el término suele aplicarse en forma genérica a las conexiones digitales, en contraste con las analógicas. Por ejemplo, una entrada TTL de un monitor requiere salida digital desde el panel de presentación no analógica.

ECL (Emitter-Coupled Logic) *Lógica acoplada de emisor*. Tipo de diseño de circuito microelectrónico que se distingue por sus extremadamente rápidas velocidades de conmutación. ECL es una variedad de transistor bipolar.

MOS (Metal Oxide Semiconductor) *semiconductor de óxido metálico*. Una de las dos categorías más importantes de diseño y fabricación de chips lógicos y chips de memoria (la otra es bipolar). La tecnología MOS deriva su nombre de su uso de capas de metal, óxido y semiconductores. Hay diversas variantes de tecnologías MOS, incluyendo PMOS, NMOS y CMOS.

CMOS (Complementary Metal Oxide Semiconductor) *semiconductor complementario de óxido metálico*. Un tipo de circuito integrado ampliamente empleado para procesadores y memorias. El CMOS utiliza transistores PMOS y NMOS en forma complementaria, de modo tal que necesitan menos energía para su funcionamiento.

I²L (Integrated Injection Logic) *lógica de Inyección Integrada*. Un tipo de diseño de transistor bipolar, conocido por su alta velocidad de conmutación.

SSI (Short Scale Integration) **Integración a pequeña escala. Hasta 100 compuertas.**

MSI (Medium Scale Integration) **Integración a mediana escala. Desde 100 a 3,000 compuertas**

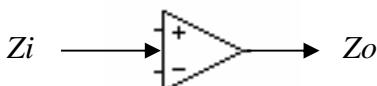
LSI (Large Scale Integration) Integración a gran escala.

Entre 3,000 a 100,000 transistores en un chip.

VLSI (Very Large Scale Integration) Integración a muy grande escala. Entre 100,000 a 1000,000 de componentes por chip.

- **Amplificadores operacionales (OPAMs) (E)**

Es un amplificador diferencial con una ganancia muy alta, con una elevada impedancia de entrada y una impedancia de salida baja. El amplificador operacional tiene la capacidad única de proporcionar una salida que esta en fase o fuera de fase con una señal de entrada aplicada. Los usos mas tipicos del amplificador operacional son proporcionar cambios de amplitud de voltaje, osciladores, circuitos de filtros y muchos otros tipos de circuitos de instrumentación. Un op-amp contiene varias etapas de amplificador diferencial para ganar una ganancia de voltaje muy alta.



Características del OPAM

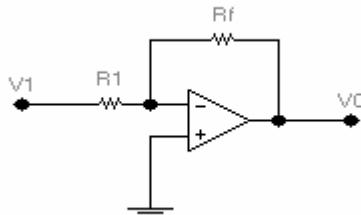
	Ideal	Valores de OPAM
A_v	∞	1000-20,000
Z_i	∞	1-100 M Ω
Z_o	0	10-100 Ω

2. Circuitos electrónicos

- **Circuitos básicos con OPAMs (osciladores, amplificadores, conmutadores, comparadores, inversores, sumadores, derivadores e integradores) (E)**

Op-amp básico.

Este circuito proporciona una conexión como un multiplicador de ganancia constante. Una señal de entrada V₁, se aplica a través de una resistencia R₁, a la terminal de entrada de signo menos. Luego, la salida se vuelve a conectar de regreso a la misma terminal de entrada de signo menos a través de una resistencia R_f; por su parte, la terminal de entrada de signo mas esta conectada a tierra. Debido a que la señal V₁ esta aplicada a la terminal de entrada del signo menos, la salida resultante esta opuesta en fase con la señal de entrada.



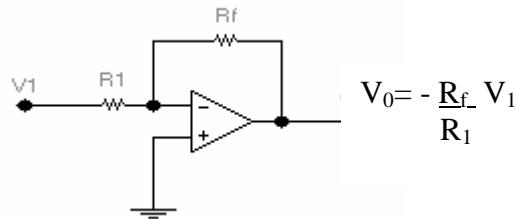
Amplificador de ganancia constante.

V₁ de entrada se aplica a través del resistor R₁ a la entrada inversora con la salida conectada de vuelta a la entrada inversora por conducto del resistor R₀. La entrada no inversora se utiliza como referencia, la que en este caso esta conectada a tierra.

$$\text{La ganancia de voltaje resultante es } A_v = \frac{V_0}{V_1} = -\frac{R_0}{R_1}$$

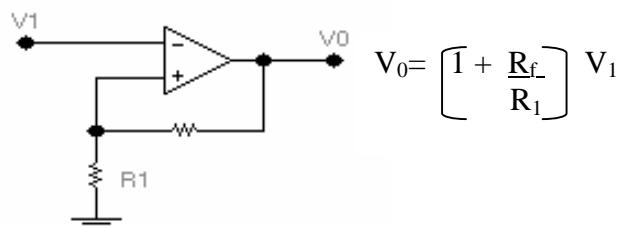
Amplificador inversor.

La salida se obtiene multiplicando la entrada por una ganancia física o constante, que determinan la resistencia de entrada (R₁) y la resistencia de retroalimentación (R_f), con la salida invertida respecto a la entrada.



Amplificador no inversor.

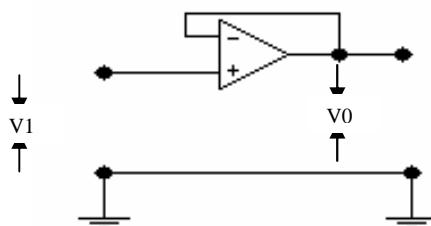
El voltaje a través de R_1 es V_{1-} debido a que $V_i \approx 0$ v. Este debe ser igual al voltaje de salida a través de un divisor de voltaje de R_1 y R_f .



Seguidor unitario.

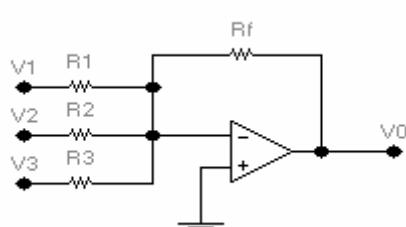
Proporciona una ganancia unitaria (1) sin inversión de polaridad o fase. A partir del circuito equivalente, esta claro que $V_0 = V_1$ y que la salida es de la misma polaridad y magnitud que la entrada. El circuito opera como un circuito emisor seguidor o seguidor de fuente, a excepción de que la ganancia es exactamente unitaria.

Este tipo de amplificador es muy útil como amplificador de acoplamiento, proporcionando una salida exactamente igual a la entrada, pero con una salida de baja impedancia capaz de manejar varias cargas.



Amplificador sumador.

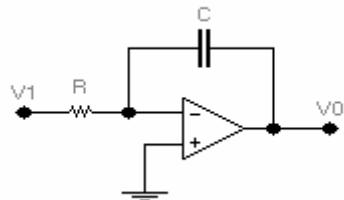
Proporciona un medio de sumar algebraicamente los voltajes, multiplicando cada uno por un factor de ganancia constante. Cada entrada suma un voltaje a la salida, multiplicado por su multiplicador de ganancias constante separado. Si se usan mas entradas, cada una añade un componente adicional a la salida.



$$V_0 = - \left(\frac{R_f}{R_1} V_1 + \frac{R_f}{R_2} V_2 + \frac{R_f}{R_3} V_3 \right)$$

Integrador.

El circuito equivalente de tierra virtual muestra que desde la entrada a la salida se puede derivar una expresión para el voltaje entre la entrada y la salida en términos de corriente I . La salida es matemáticamente la integral de la señal de salida. La tierra virtual significa que podemos considerar el voltaje entre la unión de R y X_c como si fuera tierra (debido a que $V_i \approx 0$ v), pero que ninguna corriente pasa a tierra en este punto.

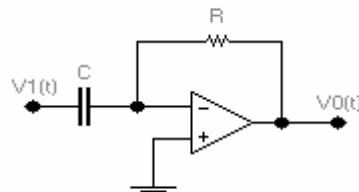


$$V_0 = -\frac{1}{C} \int V_1(t) dt$$

Diferenciador.

Proporciona una operación útil, siendo la relación resultante para el circuito.

$$V_0(t) = -RC \frac{dV_1(t)}{dt} \quad \text{donde el factor de escala es } -RC.$$



ARQUITECTURA DE COMPUTADORAS

C. Tipos y configuraciones de computadoras

I. Arquitecturas y formas de procesamiento

1. Historia y evolución

Generaciones.

CARACTERISTICAS	Primera	Segunda	Tercera	Cuarta	Quinta
DURACIÓN	1950-1960	1960-1970	1970-1980	1980-1990	1990-2000
TECNOLOGÍA	Válvulas electrónicas	Transistores	C.I. (SSI-MMI)	C.I. (LSI)	C.I. (VLSI)
FABRICANTE	IBM-UNIVAC	BURROUGHS NCR, CDC	DIGITAL	APPLE	DEC-INTEL
MAQUINA	IBM 701	CDC 6600	PDP-8 PDP-11	Fujitsu M382 Cray X-MP	Alpha 21164 P6
SIST. OPERATIVO	Muy rudimentario	Monitor de encadenamiento Muy primarios	Estructurados bajo multiprogramación y multiproceso	SO de tiempo compartido	Multiprocesamiento
TIPO DE MEMORIA	Tubos de Williams. Tambores y cintas magnéticas.	Núcleos de ferrita	Memorias en C.I. y memorias caché	Memorias virtuales	Caché a varios niveles
LENGUAJE	Maquina	FORTRAM, COBOL, ALGOL, PL1	BASIC PASCAL	Alto nivel. FORTRAM extendido	Lenguaje natural "C"
APORTACIONES HARDWARE	Registros indexados	Canales	Conexión delos CI en placas de circuito impreso	Microprocesador, memoria integrada	CI de alta escala de integración
APORTACIONES SOFTWARE	Mejoras en Ensambladores y Macroensamblador	Fuerte impulso del FORTRAN	Gran avance en los sistemas operativos	Extensión de lenguajes de alto nivel	Intento de sustituir por hardware
PRODUCTO	Computador	Computador comercial	Minicomputador	Microcomputador	Multiprocesador

Primera Generación (1951 a 1958)

Las computadoras de la primera Generación emplearon bulbos para procesar información. Los operadores ingresaban los datos y programas en código especial por medio de tarjetas perforadas. El almacenamiento interno se lograba con un tambor que giraba rápidamente, sobre el cual un dispositivo de lectura/escritura colocaba marcas magnéticas. Esas computadoras de bulbos eran mucho más grandes y generaban más calor que los modelos contemporáneos. Eckert y Mauchly contribuyeron al desarrollo de computadoras de la 1era Generación formando una compañía privada y construyendo UNIVAC I, que el Comité del censo utilizó para evaluar el censo de 1950. La IBM tenía el monopolio de los equipos de procesamiento de datos a base de tarjetas perforadas y estaba teniendo un gran auge en productos como rebanadores de carne, básculas para comestibles, relojes y otros artículos; sin embargo no había logrado el contrato para el Censo de 1950. Comenzó entonces a construir computadoras electrónicas y su primera entrada fue con la IBM 701 en 1953. Despues de un lento pero excitante comienzo la IBM 701 se convirtió en un producto comercialmente viable. Sin embargo en 1954 fue introducido el modelo IBM 650, el cual es la razón por la que IBM disfruta hoy de una gran parte del mercado de las computadoras. La administración de la IBM asumió un gran riesgo y estimó una venta de 50 computadoras. Este número era mayor que la cantidad de computadoras instaladas en esa época en E.U. De hecho la IBM instaló 1000 computadoras. El resto es historia. Aunque caras y de uso limitado las computadoras fueron aceptadas rápidamente por las Compañías privadas y de Gobierno. A la mitad de los años 50 IBM y Remington Rand se consolidaban como líderes en la fabricación de computadoras.

Segunda Generación (1959-1964) Transistor Compatibilidad Limitada

El invento del transistor hizo posible una nueva Generación de computadoras, más rápidas, más pequeñas y con menores necesidades de ventilación. Sin embargo el costo seguía siendo una porción significativa del presupuesto

de una Compañía. Las computadoras de la segunda generación también utilizaban redes de núcleos magnéticos en lugar de tambores giratorios para el almacenamiento primario. Estos núcleos contenían pequeños anillos de material magnético, enlazados entre sí, en los cuales podían almacenarse datos e instrucciones. Los programas de computadoras también mejoraron. El COBOL desarrollado durante la 1era generación estaba ya disponible comercialmente. Los programas escritos para una computadora podían transferirse a otra con un mínimo esfuerzo. El escribir un programa ya no requería entender plenamente el hardware de la computación. Las computadoras de la 2da Generación eran sustancialmente más pequeñas y rápidas que las de bulbos, y se usaban para nuevas aplicaciones, como en los sistemas para reservación en líneas aéreas, control de tráfico aéreo y simulaciones para uso general. Las empresas comenzaron a aplicar las computadoras a tareas de almacenamiento de registros, como manejo de inventarios, nómina y contabilidad. La marina de E.U. utilizó las computadoras de la Segunda Generación para crear el primer simulador de vuelo. (Whirlwind I). HoneyWell se colocó como el primer competidor durante la segunda generación de computadoras. Burroughs, Univac, NCR, CDC, HoneyWell, los más grandes competidores de IBM durante los 60s se conocieron como el grupo BUNCH.

Tercera Generación (1964-1971)

Circuitos Integrados, Compatibilidad con Equipo Mayor, Multiprogramación, Minicomputadora.

Las computadoras de la tercera generación emergieron con el desarrollo de los circuitos integrados (pastillas de silicio) en las cuales se colocan miles de componentes electrónicos, en una integración en miniatura. Las computadoras nuevamente se hicieron más pequeñas, más rápidas, desprendían menos calor y eran energéticamente más eficientes. Antes del advenimiento de los circuitos integrados, las computadoras estaban diseñadas para aplicaciones matemáticas o de negocios, pero no para las dos cosas. Los circuitos integrados permitieron a los fabricantes de computadoras incrementar la flexibilidad de los programas, y estandarizar sus modelos. La IBM 360 una de las primeras computadoras comerciales que usó circuitos integrados, podía realizar tanto análisis numéricos como administración ó procesamiento de archivos. Los clientes podían escalar sus sistemas 360 a modelos IBM de mayor tamaño y podían todavía correr sus programas actuales. Las computadoras trabajaban a tal velocidad que proporcionaban la capacidad de correr más de un programa de manera simultánea (multiprogramación). Por ejemplo la computadora podía estar calculando la nómina y aceptando pedidos al mismo tiempo. Minicomputadoras, Con la introducción del modelo 360 IBM acaparó el 70% del mercado, para evitar competir directamente con IBM la empresa Digital Equipment Corporation DEC redirigió sus esfuerzos hacia computadoras pequeñas. Mucho menos costosas de comprar y de operar que las computadoras grandes, las minicomputadoras se desarrollaron durante la segunda generación pero alcanzaron su mayor auge entre 1960 y 70.

Cuarta Generación (1971 a la fecha)

Microprocesador , Chips de memoria, Microminiaturización.

Dos mejoras en la tecnología de las computadoras marcan el inicio de la cuarta generación: el reemplazo de las memorias con núcleos magnéticos, por las de chips de silicio y la colocación de Muchos más componentes en un Chip: producto de la microminiaturización de los circuitos electrónicos. El tamaño reducido del microprocesador y de chips hizo posible la creación de las computadoras personales (PC). Hoy en día las tecnologías LSI (Integración a gran escala) y VLSI (integración a muy gran escala) permiten que cientos de miles de componentes electrónicos se almacenen en un chip. Usando VLSI, un fabricante puede hacer que una computadora pequeña rivalice con una computadora de la primera generación que ocupara un cuarto completo.

• Microcomputadoras (R)

Por el año 1972 surgen en el mercado una nueva familia de circuitos integrados de alta densidad, que reciben el nombre de microprocesadores. Las microcomputadoras que se diseñan con base en estos circuitos son extremadamente pequeñas y baratas, por lo que su uso se extiende al mercado de consumo industrial.

Un microprocesador combinado con memoria y capacidades entrada-salida se vuelve un microcomputador. El término microcomputador es utilizado para indicar un sistema de computador completo, consistente de tres unidades básicas: CPU, memoria y entrada-salida. Típicamente el microcomputador tiene un solo microprocesador. El RAM (memoria de acceso aleatorio) y el ROM (memoria de solo lectura) proporcionan la memoria para el computador. Las unidades de interfase comunican con los dispositivos entrada-salida a través de su barraje de dirección. Los datos son transferidos de y hacia la unidad seleccionada y el microprocesador vía los barrajes de datos. La información de control es transferida a través de líneas individuales en cada una especifica una función de control particular.

• **Minicomputadoras (R)**

Computadoras de tamaño mediano, que no son tan costosas como las grandes maquinas, pero que ya disponen de una gran capacidad de proceso. Surgen a mediados de la década de 1970.

• **Mainframes (R)**

Las grandes computadoras reciben el ingles el nombre de Mainframes, que significa, precisamente, gran sistema.

Una computadora grande. A mediados de los años sesenta, las épocas antiguas de las computadoras, todas las computadoras eran mainframes, ya que el termino se refería al gabinete que contenía la CPU. Aunque mainframe aún significa gabinete principal, usualmente se refiere a un sistema de computación y toda la experiencia asociada que va con el. Hay macrocomputadoras de escala pequeña, mediana y grande, manejando desde un manojo hasta varios miles de terminales en línea. Las macrocomputadoras de gran escala pueden tener centenares de megabytes de memoria principal y centenares de gigabytes de almacenamiento en disco. Las macrocomputadoras de media y gran escala usan computadoras más pequeñas como procesadores frontales que se conectan directamente a las redes de comunicaciones.

• **Máquinas de arreglos de procesadores (R)**

Esta arquitectura consiste en una malla cuadrangular de elementos procesador / memoria, que reciben las instrucciones transmitidas por la unidad de control; estas son ejecutadas por todos los procesadores, cada uno con datos de su propia memoria (cargados en la fase de inicio). El procesador de arreglos se adapta muy bien al calculo de matrices.

2. Arquitectura. Elementos básicos

• **Procesador (R, E)**

El procesador consta de dos componentes: la unidad de control, que apunta, recoge, interpreta y gobierna la ejecución de las instrucciones, el camino de datos, que recibe los operandos y efectúa la operación que implica la instrucción en curso, generando el resultado. Para que el procesador pueda realizar su cometido, hay que proporcionarle continuamente las instrucciones y los operandos así como registrar los resultados que produce. El bloque que guarda las instrucciones y los datos es la memoria,

El microprocesador, o simplemente el *micro*, es el cerebro del ordenador. Es un *chip*, un tipo de componente electrónico en cuyo interior existen miles (o millones) de elementos llamados transistores, cuya combinación permite realizar el trabajo que tenga encomendado el chip. Los micros, como los llamaremos en adelante, suelen tener forma de cuadrado o rectángulo negro, y van o bien sobre un elemento llamado *zócalo* (*socket* en inglés) o soldados en la placa o, en el caso del Pentium II, metidos dentro de una especie de cartucho que se conecta a la placa base (aunque el chip en sí está soldado en el interior de dicho cartucho). A veces al micro se le denomina "*la CPU*" (*Central Process Unit*, Unidad Central de Proceso), aunque este término tiene cierta ambigüedad, pues también puede referirse a toda la caja que contiene la placa base, el micro, las tarjetas y el resto de la circuitería principal del ordenador. La velocidad de un micro se mide en megahercios (MHz), aunque esto es sólo una medida de la fuerza bruta del micro.

Partes de un microprocesador:

- ✓ **El encapsulado:** es lo que rodea a la oblea de silicio en sí, para darle consistencia, impedir su deterioro (por ejemplo por oxidación con el aire) y permitir el enlace con los conectores externos que lo acoplarán a su zócalo o a la placa base.
- ✓ **La memoria caché:** una memoria ultrarrápida que sirve al micro para tener a mano ciertos datos que previsiblemente serán utilizados en las siguientes operaciones sin tener que acudir a la memoria RAM, reduciendo el tiempo de espera. Es lo que se conoce como *caché de primer nivel*; es decir, la que está más cerca del micro, tanto que está encapsulada junto a él. Todos los micros tipo Intel desde el 486 tienen esta memoria, también llamada *caché interna*.
- ✓ **El coprocesador matemático:** o, más correctamente, la FPU (*Floating Point Unit*, Unidad de coma Flotante). Parte del micro especializada en esa clase de cálculos matemáticos; también puede estar en el exterior del micro, en otro chip.

- **Puertos (R, E)**

Genéricamente se conoce por puerto a aquel punto donde se conectan dos dispositivos; en el caso de su computadora los puertos son aquellas interfaces por medio de las cuales interactúa con los periféricos (mouse, impresora, módem, etc.).

Existen dos tipos de puertos: seriales y paralelos y la diferencia estriba en la forma en que son transmitidos los bits: secuencialmente ("fila india") y consecutivamente (en grupos), respectivamente.

Las computadoras personales cuentan comúnmente con un puerto paralelo (conocido como LPT y dedicado a la impresora por lo general) y dos seriales (conocidos como COM1 y COM2, el primero casi siempre dedicado al mouse y el otro libre aunque esto puede ser modificado a voluntad). Otra forma de referirse al puerto serial es llamarlo RS-232C que no es otra cosa más que su nombre técnico.

3. Arquitecturas alternativas

- **Máquinas de pila, arreglo, vectorial y multiprocesador (R)**

La máquina vectorial.

La trayectoria de datos en lugar de tener una sola variable para cada entrada de la ALU, se tiene un vector con n entradas. La ALU en sí es una unidad vectorial capaz de realizar operaciones como la suma de vectores a partir de dos vectores de entrada y obteniendo como resultado un vector de salida.

Un procesador vectorial incluye operaciones de lector en su conjunto de instrucciones sumado al conjunto convencional de instrucción es que normalmente se encuentra en un CPU. Las instrucciones vectoriales típicas incluyen operaciones aritméticas con datos de punto flotante, operaciones lógicas, comparar, probar, operaciones en matriz tales como el producto de punto, e instrucciones de búsqueda para valores máximos o mínimos. Los procesadores vectoriales son especialmente diseñados para ejecutar instrucciones complejas rápida y eficientemente. Esto se hace por la organización interna que combina unidades funcionales múltiples operando en paralelo junto con procesamiento de oleducto dentro de las unidades funcionales.

El procesador arreglo.

Un procesador adecuado para procesamiento vectorial algunas veces se denomina procesador arreglo. Esto es debido a que el procesador es capaz de realizar cálculos paralelos en arreglos grandes de datos. El nombre arreglo se utiliza también para significar un tipo especial de organización paralela. Aquí el arreglo designa conexión de hardware de un número de unidades procesadoras idénticas conectadas junto con rutas directas entre cada unidad y sus vecinos.

Multiprocesador.

Diferentes CPU manejan distintos programas compartiendo a veces, una memoria común, a la cual tiene acceso a cada procesador a través de un bus. Para reducir la competencia por el acceso y mejorar el desempeño, los diseñadores de multiprocesadores elaboraron varios esquemas, en los cuales cada uno de los procesadores tiene cierta memoria local propia que no es accesible al resto. Esta puede ser usada para almacenar el código del programa así como para aquellos datos que no necesitan compartirse. El acceso a esta memoria privada no utiliza el bus principal, con lo que el tráfico se reduce en gran medida. Otros multiprocesadores utilizan más de un bus para reducir la carga, mientras que otros, usan la llamada memoria caché, una técnica para mantener en cada procesador las palabras de memoria usadas con mayor frecuencia.

- **Conceptos CISC y RISC (R)**

Maquinas CISC.

CISC es un acrónimo de las siglas en inglés para Computadora con Conjunto de Instrucciones Complejo. Las máquinas CISC tienen de doscientas a trescientas instrucciones, que están grabadas en micro código.

Maquinas RISC.

RISC es un acrónimo de las siglas en inglés para Computadora con Reducido Conjunto de Instrucciones. La arquitectura de la computadora RISC es un intento por obtener más potencia del CPU simplificando su conjunto de instrucciones. El diseño del hardware RISC trata de ejecutar instrucciones reales sencillas en una fracción del tiempo que suele requerirse para decodificar y realizar instrucciones más complejas. Las computadoras RISC poseen un pequeño número de instrucciones montadas en los circuitos de nivel inferior, que trabajan a máxima

velocidad. Aunque las máquinas RISC con sólo de un 15% a un 50% más veloces que sus contrapartidas CISC, los chips RISC son más baratos de producir.

• Arquitecturas secuenciales y arquitecturas paralelas (R)

Arquitectura en paralelo.

Aunque hay ciertas variaciones entre los procesadores y los diseños de memoria, el área en la que los sistemas en paralelo difieren mas es en como acomodar las piezas.

Algunas computadoras en paralelo se diseñan para correr de manera simultanea diversas tareas independientes, estas tareas son independientes entre sí y no se comunican (granulado del paralelismo, se refiere a los algoritmos y al software, pero tiene una gran correspondencia directa en el hardware). Hay computadoras en paralelo que se utilizan para correr un solo trabajo consistente de muchos procesos paralelos (paralelismo ordinario, evento de correr grandes porciones de software en paralelo, con poca o ninguna comunicación entre las partes). Se llega a maquinas en las que el paralelismo esta dado por un alto grado de procesamiento en serie o de muchas ALU operando en la misma corriente de instrucciones al mismo tiempo (paralelismo fino, procesamiento de vectores).

• Memoria compartida y distribuida (R)

• Clusters (R)

El cluster es un grupo de computadoras libremente acopladas que se configuran para que trabajen juntas como una unidad. A diferencia del sistema de multiprocesamiento, cada computadora del cluster es una unidad completa con su propio CPU, su memoria y su capacidad de entrada/salida. De hecho, puede ser a su vez sistema de multiprocesamiento. A las computadoras individuales se les llama nodo. A diferencia de la red, se busca que el usuario las considere una sola maquina. El cluster es transparente para él.

El propósito obvio del agrupamiento es aumentar la potencia disponible de computo al combinar la de cada computadora. Como pueden procesar los datos en forma independiente, el aumento es aproximadamente proporcional al numero de nodos del cluster.

Se obtienen otras ventajas. El cluster es escalable por naturaleza, tanto incrementalmente como absolutamente. Una instalación puede incorporar mas nodos a medida que se necesite mas potencia de computación. Además, es posible crear un cluster con gran cantidad de nodos. La formación de clusters aumenta la tolerancia y la disponibilidad. Como las computadoras que lo integran pueden funcionar de manera independiente, la falla en un nodo no interrumpirá el sistema entero.

Dos modelos principales se emplean en la formación de clusters:

- ✓ **El modelo de nada compartido (shared-nothing):** se asemeja a una conexión de red punto a punto entre dos computadoras. Cada computadora posee sus propios discos. La diferencia más importante es la presencia de un vínculo que transmite mensajes a alta velocidad entre los nodos, junto con el software que controla el comportamiento y la interacción entre ellos. La carga de trabajo se divide repartiendo los datos entre los nodos, de manera que las solicitudes que se les hacen sean más o menos iguales. Este método tiene la ventaja de que se requiere poca comunicación entre los nodos, porque son esencialmente independientes. El problema principal de la configuración radica en que no siempre es posible planear y predecir exactamente la partición.
- ✓ **El modelo de disco compartido (shared-disk):** Los datos pueden compartirse entre nodos del cluster porque los nodos pueden acceder a los discos presentes. El modelo ofrece la ventaja de un equilibrio fácil y dinámico de la carga de trabajo, además de que con un diseño riguroso se obtiene gran disponibilidad y tolerancia a las fallas. Aunque estas ventajas hacen atractivos los clusters, la complejidad del software tiene sus costos. El software que los controla debe ser capaz de mantener la coordinación y sincronización de los datos que procesan varios nodos, de evitar que se corrompan los datos compartidos y garantizar una recuperación fiel.

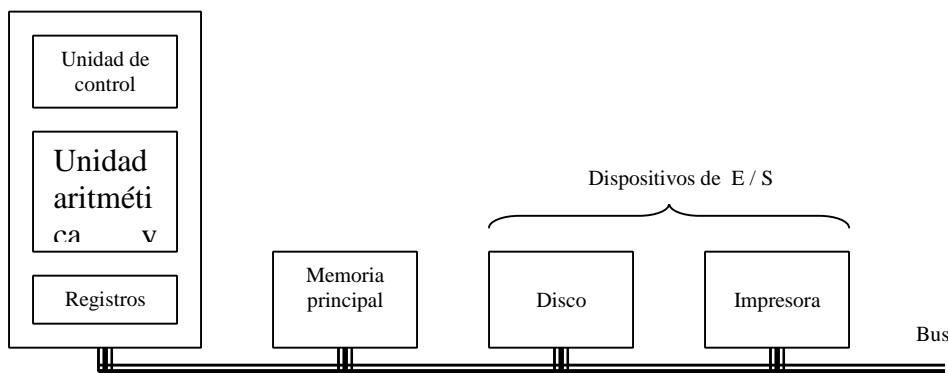
• Configuración en espejo (R)

- Comparación de arquitecturas, eficiencia, aplicaciones de arquitecturas alternas (R)

RISC	CISC
Instrucciones sencillas en un ciclo	Instrucciones complejas en varios ciclos
Solo LOAD/STORE hacen referencia a memoria	Cualquier instrucción puede hacer referencia a memoria
Procesamiento serie de varias etapas	Poco procesamiento en serie
Instrucciones ejecutadas por hardware	Instrucciones interpretadas por un microprograma
Instrucciones de formato fijo	Instrucciones de formato variable
Pocas instrucciones y modos	Muchas instrucciones y modos
La complejidad está en el compilador	La complejidad está en el microprograma
Varios conjuntos de registros	Un solo conjunto de registros

4. Selección y configuración de sistemas de cómputo

Unidad central de procesos (CPU)



Organización de una computadora sencilla, con un CPU y dos dispositivos de E / S

¿Qué equipo comprar?

Antes de comprar un equipo, tenemos que tener en cuenta ciertas premisas para evitar después desengaños o no saber rentabilizar el potencial total de nuestra máquina. Primero tendremos que valorar las siguientes cosas:

- 1.- Uso que le vamos a dar (Internet, juegos, profesional,)
- 2.- Presupuesto
- 3.- Posibilidades de ampliación.
- 4.- Cuantos usuarios lo van a utilizar.

- Unidad central de procesamiento y periféricos (R, E)

Unidad Central de Procesamiento (CPU).

Es el cerebro de la computadora. Su función es ejecutar programas almacenados en la memoria central tomando sus instrucciones, examinándolas y luego ejecutándolas una tras otra. La CPU se compone de varias partes. La unidad de control se encarga de traer las instrucciones a la memoria principal y de determinar su tipo. La unidad aritmética y lógica realiza operaciones para llevar a cabo las instrucciones. La CPU también contiene una pequeña memoria de alta velocidad utilizada para almacenar resultados intermedios y cierta información de control. Esta memoria consta de varios registros, cada uno de los cuales tiene cierta función. El registro más importante es el Contador de Programas (CP), que indica la próxima instrucción que debe ejecutarse. También es importante el Registro de Instrucción, que contiene la instrucción que está ejecutando. La mayoría de las computadoras tienen además otros registros.

Periféricos.

Los dispositivos entrada o salida adicionales al computador en línea (que están bajo control directo del procesador) o fuera de línea (operado independientemente del computador) se denominan periféricos.

• **Terminales: impresoras, monitores, teclados (R, E)**

Las terminales de computadora consisten de tres elementos: un teclado, un monitor y algunos componentes electrónicos que controlan a los dos anteriores.

Impresoras.

Se diseñaron para cubrir la necesidad de llevar un registro permanente en papel de los resultados.

- ✓ **Impresoras de impacto.** La más antigua clase de impresoras, trabaja como una maquina de escribir: una pieza de metal o plástico con letras en relieve golpea una cinta entintada contra una hoja de papel, dejando en este una imagen de una letra.
- ✓ **Impresoras de matriz.** Una cabeza impresora que contiene entre 7 y 24 agujas activadas en forma electromagnética, se desplaza a lo largo de cada línea de impresión. Cuando se aplica corriente a un electroimán, se hace que el pin correspondiente se dirija al papel. Se monta la cabeza de impresión de modo que se desplace a través del papel. Se mueve este por la impresora en un rodillo de platina. La cabeza imprime una línea cruzando el papel mientras se ejerce un control estricto sobre los pines para crear la imagen deseada. El pin que debe formar un punto golpea el papel mediante una cinta entintada. Cuando se imprime un renglón, la platina lo lleva al siguiente y la impresión prosigue.
- ✓ **Impresora láser.** El corazón de la impresora es un tambor rotativo de precisión. Al principio de cada ciclo de pagina se carga con alrededor de 1000 volts y se recubre de material fotosensitivo. La luz de un láser se desplaza a lo largo del tambor . El haz de luz se modula para producir un patrón de puntos claros y oscuros. Los puntos donde el haz toca, pierden su carga eléctrica. Después de pintar una línea, el tambor gira una fracción de grado para permitir que se pinte el siguiente. Al ir girando la primera línea llega donde se encuentra el tóner, que es un polvo negro electrosensitivo que se adhiere a aquellos puntos que aun conservan su carga, formando así una imagen visual de dicha línea. Mas adelante en la trayectoria, el tambor recubierto de tóner se presiona contra el papel transfiriendo a este el polvo negro. El papel pasa entonces a través de rodillos calientes para unir de manera permanente el tóner al papel y fijar la imagen. El tambor sigue rotando y se descarga y un raspador lo limpia de cualquier residuo de tóner, preparándose para una nueva carga recubrimiento, para la siguiente pagina.
- ✓ **Impresora de inyección de tinta.** Desde el punto de vista mecánico funciona e forma semejante a las de matriz de puntos. El cartucho de impresión se desplaza por la pagina para imprimir un renglón, y unos rodillos mecánicos bajan la pagina para imprimir los renglones sucesivos. El cartucho contiene un deposito de tinta y un renglón de diminutas boquillas más pequeñas que el ancho de un cabello humano. Se produce un punto calentando la tinta que esta detrás de la boquilla. Cuando se hiere esparce una pequeña gota hacia el papel. Algunas impresoras usan un piezo-cristal vibratorio en vez del calor para producir las gotas de tinta.
- ✓ **Impresoras de transferencia de cera térmica y se sublimación de tinta.** Se utilizan para obtener imágenes cromáticas de alta calidad. El mecanismo de ambos es similar. Se introduce papel en la impresora y se sujet a un tambor. Una cabeza de impresión ofrece un renglón de elementos térmicos del tamaño de un punto. Entre el papel y la cabeza de impresión la impresora introduce un rollo de película que se impregna con una cera o tinta de color. La película consta de secciones del tamaño de una pagina de magenta, azul verdusco y amarillo; algunas veces se incluye una sección de negro. Cada rotación del tambor expone el papel a un color distinto. El calor proveniente de la cabeza de impresión fusiona la cera o la tinta con el papel. LA cera térmica puede aplicarse al papel ordinario. El método de sublimación de tinta es un poco diferente, pues las tintas se esparcen en el papel, de modo que los puntos cromáticos se mezclen. Mas aun, es posible controlar la cantidad de tinta con solo ajustar la temperatura de los elementos de la cabeza de impresión.

Monitor.

Es en esencia una caja consistente de un tubo de rayos catódicos y sus fuentes de poder. El tubo consiste en un cañón que puede disparar un haz de electrones contra la pantalla fosforescente ubicada cerca de la parte frontal de tubo.

Teclados.

Los teclados se presentan en diversas variedades; en los más baratos, cada tecla es simplemente un interruptor que hace contacto cuando se presiona. Otros mas caros tienen un imán bajo cada tecla que pasa a través de una bobina cuando se presiona induciendo así una corriente eléctrica que puede detectarse. Se utilizan también algunos otros métodos tanto mecánicos como electromagnéticos en algunos teclados.

Cuando se oprime una tecla, se envía al controlador una señal llamada código de exploración. S emite otro código cuando se libera la tecla. Eso sucede en todas las teclas. Por medio de dos códigos las teclas pueden emplearse en combinación, pues el controlador sabe si una tecla se tiene oprimida mientras se pulsa otra. También puede

determinar cuando una tecla produce una acción repetida.

• Discos y cintas magnéticos y ópticos (R, E)

Cinta magnética.

Fue el primer tipo de memoria secundaria. La cinta es una faja de plástico revestida con un medio de registro magnético. Los bits son registrados como puntos magnéticos en la cinta a lo largo de varias pistas. Usualmente 7 o 9 bits son registrados simultáneamente para formar un carácter junto con un bit de paridad. Las cabezas lectura / escritura se montan una en cada una de las pistas de tal manera que los datos puedan ser grabados y leídos como una secuencia de caracteres.

Las cintas magnéticas son dispositivos de acceso secuencial. Si la cinta esta situada al principio, para leer el registro físico n primero es necesario leer los registros físicos 1 a $n-1$, de uno en uno.

Discos magnéticos.

Consta de uno o más platos planos circulares hechos de vidrio, de metal o plástico y revestidos con una sustancia magnética semejante a la que se usa en la cinta o casete (Englander, 273). Su diámetro varía entre 5 y 10 pulgadas. La información se graba en cierto numero de círculos concéntricos llamados pistas. El numero de pistas en un disco es de 40 a varios cientos en cada superficie. Para cada disco se tiene una cabeza que puede moverse acercándose o retirándose del centro del mismo. Esta cabeza es ancha como para leer o escribir información de exactamente una pista. Una unidad de disco a menudo tiene varios discos dispuestos verticalmente con una separación entre sí de aproximadamente una pulgada. Las pistas se dividen en sectores, normalmente entre 10 y 100 sectores por pista. Un sector consta de cierto numero de bytes, por lo general 512.

Discos flexibles.

Fue adoptado muy rápido como un medio conveniente para la distribución del software. Es llamado disquete o disco flexible, porque los primeros discos eran físicamente flexibles. Los disquetes de 3.5 pulgadas vienen en un empaque rígido para su protección, por lo que no son físicamente flexibles.

Discos ópticos.

Originalmente fueron desarrollados para grabar programas de televisión. Estos discos se denominan CD ROM (Memoria de Lectura Solamente en Disco Compacto). Estos discos se preparan usando un láser de alto poder para laserizar agujeros de una micra en un disco matriz; después se hace un molde para imprimir copias en discos plásticos. Luego se aplica en la superficie de una pequeña capa de aluminio, seguida de otra de plástico transparente para protección. Los discos CD ROM se leen por medio de un detector que mide la energía reflejada de la superficie al apuntar a esta un láser de bajo poder. Los agujeros, que se denominan huecos, y las áreas sin laserizar entre estos, la superficie plana, tienen diferente reflectividad, lo que hace posible distinguir entre ambos. La información en los CD ROM se graba en una sola espiral continua, a diferencia de los cilindros y pistas de los discos magnéticos.

• Otros periféricos: ratón, scanners, bocinas (R, E)

Ratón.

Un ratón es una pequeña caja de plástico que se apoya en una tableta cerca de la terminal. Cuando se mueve alrededor de la tableta, se mueve también en la pantalla un pequeño cursor, permitiendo al usuario seleccionar sus opciones. El ratón tiene de uno a tres botones en la parte superior para realizar la selección de los menús. Se usan tres clases de ratones: mecánicos, ópticos y optomecánicos.

- ✓ **Ratón mecánico:** tiene dos ruedas de hule que sobresalen de la base, con sus ejes perpendiculares entre sí. Cuando se mueve en forma perpendicular a su eje principal, la otra rueda es la que gira. Cada rueda maneja un resistor variable (potenciómetro).
- ✓ **El ratón óptico:** no tiene rueda ni esfera, en su lugar, tiene un diodo emisor de luz y un fotodetector en la base. El ratón óptico se utiliza sobre un tablero especial de plástico que tiene una cuadricula rectangular de líneas muy poco espaciadas entre sí. Al mover el ratón sobre la cuadricula, el fotodetector capta los cruces de las líneas observando los cambios en la cantidad de luz del diodo que refleja.
- ✓ **El ratón optomecánico:** tiene dos ruedas giratorias montadas a 90 grados una de la otra, cada rueda tiene un diodo emisor de luz en el centro, una serie de cortes espaciados de manera uniforme alrededor de la circunferencia de la rueda, y un detector justo afuera de la rueda. Al moverse esta, los impulsos de luz alcanzaban a los detectores a cada vez que hay un corte de la rueda entre ambos. El numero de impulsos detectados es proporcional a la cantidad de movimiento realizado.

Escáner.

Los escáneres (digitalizadores) son el medio más común con que se introducen imágenes de papel. Hay tres tipos fundamentales de escáneres: los de cama, los escáner con alimentación de hoja y los manuales; los tres funcionan de manera parecida y se distinguen solo en la forma de que el elemento de exploración se mueve respecto al papel. En los primeros se pone un papel en una ventana de vidrio, mientras el elemento digitalizado baja por ella como una máquina copiadora. En segundos se impulsa una página de papel mediante el mecanismo con rodillos. Por último, el usuario impulsa los escáneres manuales sobre la página. La operación fundamental es la misma sin importar qué medio se utilice. El mecanismo consta de una fuente luminosa y de una hilera de sensores luminosos. A medida que la luz se refleja contra los puntos de la página, los sensores la reciben y la traducen a señales digitales correspondientes a la brillantez de cada punto. Por medio de filtros de color se producen imágenes cromáticas, ya sea proporcionando muchos sensores o barriendo la imagen con un filtro de color en cada paso.

• Configuración de memorias centrales (R, E)

La memoria central en la mayoría de las computadoras se configura de manera automática esto es, después de ser instalada al encender la computadora el Bios realiza una prueba del sistema y detecta la cantidad de memoria que se encuentra instalada.

• Configuración de memoria virtual (R, E)

En sistemas operativos como Windows 95 y 98 existe una opción que permite al usuario configurar la cantidad de memoria virtual a utilizar y por default se administra de forma automática. En sistemas tipo Unix la definición de la memoria virtual se realiza en el momento de particionar el disco duro al crear una partición de tipo “swap” la cual se sugiere sea el doble de tamaño de la memoria física es decir si tenemos 32 Mbytes de memoria física se debe crear una partición de aproximadamente 64 Mbytes en el disco duro.

Memoria virtual.

Es un mecanismo que gestiona automáticamente la transferencia de información entre los dos niveles de jerarquía de memoria: memoria principal y memoria secundaria externa. El programador tiene la impresión de disponer de todo el tamaño del disco, aunque el procesador, en cada momento solo puede acceder a lo que tiene cargado en la memoria principal.

Cuando se genera una dirección virtual de un elemento al que se desea acceder se envía a la MMU (Unidad de manejo de memoria), la cual consultando las tablas que posee, comprueba si dicho elemento está en la memoria principal. En caso afirmativo, se procede a un acceso directo, para lo cual obtiene la dirección física que se obtiene de dichas tablas de traducción y se deposita en el bus de direcciones. En caso de que la MMU detecte la ausencia del objeto buscado en la memoria principal, se comunica el hecho al sistema operativo, que pone en marcha una rutina cuya misión es localizar el objeto en el disco y transferir el bloque donde se encuentra la memoria principal, actualizando las tablas de la MMU para poder efectuar el acceso directo.

Tipos de memoria virtual.

- ✓ **Memoria paginada.** En este modelo se organiza el espacio virtual y el físico en bloques del mismo tamaño, llamados páginas. Las páginas suelen tener un tamaño comprendido entre 512 y 8.192 bytes.
- ✓ **Memoria segmentada.** Este modelo explota la técnica de modularidad de los programas construidos de forma estructurada. Los módulos son bloques de información semejante, que al estar agrupada puede manejarse de forma independiente. A dichos módulos se les llama segmentos. La segmentación organiza el espacio virtual y el físico en bloques de tamaño variable, que se ajustan al tamaño que deben tener los segmentos para contener la información semejante.
- ✓ **Memoria con segmentos paginados.** Combina las ventajas de los dos modelos anteriores. Inicialmente, la memoria se descompone en segmentos de tamaño variable, pero cada uno de ellos se divide en páginas de tamaño fijo. Para acceder a un objeto, se comienza consultando la tabla de segmentos para localizar el segmento en que se encuentra y si está presente en la memoria principal. Posteriormente se consulta la tabla de páginas correspondiente al segmento seleccionado.

• Formas de procesamiento: monoprocesamiento, multitarea, por lotes, tiempo compartido, tiempo real (R, E)

Monoprocesamiento.

En este tipo de procesamiento un programa se carga en la memoria y se ejecuta a la vez. El sistema operativo de una sola tarea esta diseñado para procesar un programa a la vez. Este sistema operativo prácticamente esta obsoleto.

El sistema operativo ya no interviene una vez iniciada la ejecución del programa del usuario. Y este tiene prioridad en un sistema de un solo trabajo o tarea. A menos que requiera un servicio especial del sistema operativo, este no cumple ninguna función específica y permanece ocioso, de manera que el programa del usuario puede correr sin interrupción. El sistema operativo recobra el control de la computadora si la falla o error del sistema causa una interrupción en el hardware.

El sistema operativo con una sola tarea ofrece una administración mínima de la memoria, sin capacidad de planificar. No se necesita la planeación, pues no hay actividades que realizar durante la ejecución del programa; simplemente se suspende el sistema operativo mientras se corre el programa. Mas aun, como hay un solo programa que cargar, es fácil tomar la decisión de donde colocarlo.

Multitareas.

Casi todas las computadoras modernas ofrecen los medios y el soporte para manipular varios programas en un CPU, técnica conocida como multitareas o multiprogramación..

La técnica de multitareas se aplica a un solo usuario o se diseña para permitir que varios compartan los recursos de la computadora. Este sistema, denominado sistema multiusuarios, seguirá perteneciendo a la categoría de multitareas porque cada uno correrá al menos un programa. La mayoría de los sistemas les permiten, además, ejecutar varios programas concurrentemente. Aparte de los servicios al usuario, el sistema ha de asignar recursos a varias actividades y usuarios en forma equitativa y eficiente: memoria, dispositivos de entrada/salida y tiempo del CPU. El CPU procesa solo una instrucción a la vez, por lo cual con un solo procesador es evidentemente imposible ejecutar simultáneamente dos o más programas. Por eso, el sistema operativo actúa como controlador para llevar a cabo el procesamiento concurrente.

Procesamiento por lotes.

El proceso en lotes se organiza de la forma siguiente: los usuarios dejan sus trabajos en forma de paquetes de tarjetas, los cuales son leídos y grabados en cinta., cuando se tiene un grupo (lote) se pasa la cinta a la computadora la cual va leyendo los trabajos uno a uno, los procesa e imprime sus resultados en otra cinta, al terminar el lote, la cinta de salida se retira y se imprimen los, resultados. Estos sistemas son multiusuario porque en un cierto periodo de tiempo (algunas horas) atiende a un grupo de usuarios, todos los cuales reciben sus resultados al final del ciclo.

Los sistemas de lotes tienen limitaciones, al requerir de un operador que monte y desmonte cintas y también porque puede ser bloqueado por un usuario abusivo o descuidado, que envíe un trabajo excesivamente largo. Con la aparición de los discos magnéticos se hizo innecesario el uso de cintas como intermediarias, ya que para entonces existían controladores de dispositivo independientes del procesador. Con esto se obtiene un flujo continuo de trabajos, de la siguiente manera. Los trabajos se leen a disco y se anotan en una cola., de ahí se van tomando uno a uno, se procesan y los resultados se mandan a disco, formando otra cola. Estos sistemas mejoran el uso de la computadora y permiten la ejecución de trabajos en orden distinto al de llegada, evitando algunos de los defectos del proceso en lotes. De esta manera, se tienen los, sistemas de proceso continuo, en los cuales en cualquier momento pueden estarse cargando programas a disco, imprimiendo resultados de disco a papel o cinta, y procesando trabajos, todo en forma simultánea. Los sistemas anteriores, aún cuando aceptan varias tareas y varios usuarios, solo cargan una tarea a la vez a la memoria. Con el abaratamiento de memoria y uso de terminales en vez de tarjetas, se logró un nuevo tipo, al que se llama de multiacceso por el hecho de aceptar entradas y enviar salidas a diferentes puntos simultáneos. A la vez, con técnicas de manejo de memoria pueden tenerse varias tareas listas en memoria (a esto se le llama multiprogramación). De ésta manera, cuando una tarea deja el procesador para esperar un dato (que debe ser tecleado o transferido de algún dispositivo), de inmediato se inicia la ejecución de otra, sin

tener que esperar. De esta manera se atiende a muchos usuarios, aprovechando mejor el tiempo del procesador, ya que los dispositivos siguen siendo mucho más lentos que el procesador, y el uso interactivo es aún más lento, dejando mucho tiempo para ser aprovechado por otras tareas.

Tiempo compartido.

En un sistema tiempo compartido, muchos usuarios se comunican con el computador vía dispositivos de terminales remotos. Debido a que la respuesta humana es lenta comparada con las velocidades del computador, el computador puede responder a usuarios múltiples al mismo tiempo. Esto se logra haciendo que muchos programas residan en la memoria mientras el sistema asigna un pedazo de tiempo a cada programa para la ejecución en la CPU. El sistema de operación asigna a cada uno de los trabajos una porción de tiempo sobre una base prioritaria. Un *trabajo* es una unidad de trabajo específico como se aplica a la ejecución de las tareas de procesamiento de datos. Una *porción de tiempo* es una cantidad de tiempo asignada a un trabajo. En cualquier porción de tiempo, el sistema de operación hace que el computador procese un trabajo hasta que una de las cuatro condiciones siguientes ocurra: (1) el trabajo se complete; (2) se detecte un error; (3) se requiera una entrada o salida; o(4) la porción de tiempo se acabe. En cualquier caso el procesador es asignado al trabajo con la prioridad más alta siguiente. En los dos primeros pasos, el trabajo puede retirarse de la memoria. En los dos últimos el trabajo es solamente suspendido en forma temporal.

Tiempo real.

Los sistemas de control de procesos son los que con más propiedad pueden llamarse "de tiempo real". En ellos la ejecución de tareas está gobernada por la ocurrencia de eventos externos que requieren de atención inmediata, como pueden ser el aumento de temperatura en una caldera o el vaciado de un depósito. En estos sistemas las tareas se agrupan también por prioridades y tiempos límite a su ejecución. La diferencia entre los diferentes tipos está en el énfasis en la rapidez de respuesta y en las consecuencias de un retraso: desde un enojo en los primeros hasta una tragedia en los últimos. Existen sistemas de tiempo real para microcomputadoras, que vienen en un chip.

II. Sistemas de propósito especial

1. Configuraciones especializadas

• Dispositivos programables (R, E, RP)

Para facilitar a las empresas el desarrollo de productos basados en ROM, se inventó la memoria PROM (ROM programable). Esta memoria es como una ROM, excepto en que puede programarse (una sola vez).

El siguiente avance en esta línea fue la EPROM (PROM borrable), que no solo puede programarse sino también borrarse. Cuando la ventana de cuarzo de una EPROM se expone a luz ultravioleta potente durante unos 15 minutos, todos sus bits se ponen a 1.

Aún mejores que las EPROM son las EEPROM (EPROM eléctricamente borrables), que pueden borrarse mediante impulsos eléctricos, sin necesidad de que las introduzcan en un receptáculo especial para exponerlas a luz ultravioleta.

REDES

A. Transmisión y comunicación de datos

I. Conceptos básicos

1. Teoría de la información

Es la rama de la teoría matemática de la probabilidad y estadística que se relaciona con los conceptos de información y entropía de la información, sistemas de comunicación, transmisión de datos, así como la teoría de la distorsión de la transferencia, criptografía, relaciones señal-ruido, compresión de datos y temas relacionados.

Es de aceptación general que la disciplina de la teoría de la información comenzó con la publicación del artículo de Claude E. Shannon "La Teoría Matemática de la Comunicación" (The Mathematical Theory of Communication), en el Diario Técnico del Sistema Bell (Bell System Technical Journal) en Julio y Octubre de 1948. Este trabajo tuvo origen a partir de publicaciones anteriores de Harry Nyquist y Ralph Hartley. En el proceso de obtener una teoría de comunicaciones que pudiera ser aplicada por ingenieros eléctricos para diseñar mejores sistemas de telecomunicaciones.

Claude E. Shannon es conocido como el padre de la teoría de la información. Su teoría considera la transmisión de la información como un fenómeno estadístico y ofrece a los ingenieros en comunicaciones una forma de determinar la capacidad de un canal de comunicación en términos comunes de medida llamados bits. La parte de la teoría que hace referencia a la transmisión no está relacionada con el contenido de información o el mensaje en sí mismo, aún cuando el lado complementario de la teoría de la información se preocupa con el contenido a través de la compresión con pérdida de los mensajes sujetos a un criterio de fidelidad. Estas dos ramas de la teoría de la información están unidas y justificadas mutuamente por los teoremas de transmisión de información, o los teoremas de separación de canal de origen que justifican el uso de bits como el formato universal de información en diversos contextos

Esta teoría fue formulada en 1940 por el ingeniero Claude E. Shannon, y aparece publicada por primera vez en octubre de 1948. Posteriormente el sociólogo Warren Weaver redactó un ensayo destinado a enfatizar las bondades de esta propuesta, el cual fue publicado en 1949.



Componente de naturaleza humana o mecánica que determina el tipo de mensaje que se transmitirá y su grado de complejidad.	Recurso técnico que transforma el mensaje originado por la fuente de información en señales apropiadas.	Medio generalmente físico que transporta las señales en el espacio (cumple funciones de mediación y transporte).	Recurso técnico que transforma las señales recibidas	Componente terminal del proceso de comunicación, al cual está dirigido el mensaje.
---	---	--	--	--

RUIDO Expresión genérica utilizada para referirse a las distorsiones originadas en forma externa al proceso de comunicación

• Transmisión y comunicación de información (R, E, RP)

Para empezar, definimos a la comunicación como un proceso por medio del cual la información se transfiere de un punto llamado fuente, en espacio y tiempo, a otro punto que es el destino o usuario. Un sistema de comunicación es la totalidad de mecanismos que proporcionan el enlace para la información entre fuente y destino. Un sistema de

comunicación eléctrica es aquel que ejecuta esta función principal, pero no exclusivamente, por medio de dispositivos y fenómenos eléctricos.

Hay muchas clases de fuentes de información, incluso hombres y máquinas; por eso, los mensajes aparecen en muchas formas; una secuencia de símbolos o letras discretas; una magnitud sencilla variando con el tiempo, varias funciones del tiempo y otras variables. Pero, sea cual fuere el mensaje, el objeto de un sistema de comunicación, es proporcionar una réplica aceptable de él en su destino.

Como regla, establezcamos que el mensaje producido por una fuente no es eléctrico y, por lo tanto, es necesario un transductor de entrada. Este transductor convierte el mensaje en una señal, una magnitud eléctrica variable, tal como un voltaje o una corriente. Similarmente, otro transductor convierte la señal de salida a la forma apropiada del mensaje. En lo sucesivo, los términos señal y mensaje se usarán indistintamente.

Elementos funcionales: Omitiendo los transductores, hay tres partes esenciales en un sistema de comunicación eléctrica, el transmisor, el canal de transmisión y el receptor. Cada uno con su función característica.

Transmisor: El transmisor pasa el mensaje al canal en forma de señal. Para lograr una transmisión eficiente, se deben desarrollar varias operaciones de procesamiento de la señal. La más común e importante de estas operaciones es la modulación, un proceso que se distingue por el acoplamiento de la señal transmitida a las propiedades del canal, por medio de una onda portadora.

Canal de transmisión: El canal de transmisión o medio es el enlace eléctrico entre el transmisor y el receptor, siendo el puente de unión entre la fuente y el destino. Puede ser un par de alambres, un cable coaxial, una onda de radio o un rayo láser. Todos caracterizados por la atenuación, la disminución progresiva de la potencia de la señal conforme aumenta la distancia. La magnitud de la atenuación puede ser pequeña o muy grande, generalmente es grande.

Receptor: La función del receptor es extraer del canal la señal deseada y entregarla al transductor de salida, como las señales son frecuentemente débiles, resultado de la atenuación, el receptor debe tener varias etapas de amplificación.

Contaminaciones: Durante la transmisión de la señal ocurren ciertos efectos no deseados. Uno de ellos es la atenuación, la cual reduce la intensidad de la señal; sin embargo, son más serios la distorsión, la interferencia y el ruido, los cuales se manifiestan como alteraciones de la forma de la señal.

Estos efectos se clasifican de la manera siguiente:

Distorsión: es la alteración de la señal debida a la respuesta imperfecta del sistema a ella misma. A diferencia del ruido y la interferencia, la distorsión desaparece cuando la señal deja de aplicarse.

En la práctica debe permitirse cierta distorsión, aunque su magnitud debe estar dentro de límites tolerables.

Interferencia: es la contaminación, por señales extrañas, generalmente artificiales y de forma similar a las de la señal. La solución al problema de interferencia es obvia: eliminar en una u otra forma la señal interferente o su fuente.

Ruido; se le llama así a las señales aleatorias e impredecibles de tipo eléctrico originadas en forma natural dentro o fuera del sistema. Cuando estas variaciones se agregan a la señal portadora de la información, ésta puede quedar en gran parte oculta o eliminada totalmente..

• Medida de la información (entropía) (R)

El primer problema que nos deberíamos plantear es el de la medida de la información. Parece intuitivo decidir en que medida conocemos un sistema, pero necesitamos una formalización. La pregunta puede plantearse en unos términos bastante sencillos:

Supongamos que nos dan el valor de un cierto número, X. ¿Cuánta información obtenemos a partir de esto? Bien, esto depende de lo que supiésemos previamente sobre ese número. Por ejemplo, digamos que ya sabíamos el valor. En tal situación habremos aprendido exactamente nada. Por otra parte, pongamos que sabíamos que el valor X es obtenido al tirar un dado.

Una observación: una medida de la información es a su vez una medida de la ignorancia, puesto que la información que, dependiendo del contexto, contenga X, es precisamente la que ganaríamos al conocer su valor, y por lo tanto parte de la incertidumbre inicial.

Unidades de medida.

Un ordenador personal es un sistema electrónico digital. Debido a esto cualquier estructura de información, por más compleja que sea, se presenta y manipula como una colección de "0" y "1": muy sencillo aunque tremadamente voluminoso.

- ✓ **Bit:** Llamamos bit a la unidad mínima de información binaria. Un bit puede tomar el valor "1" o "0". Cualquier estructura de información puede ser representada por un conjunto de bits.
- ✓ **Byte:** Debido a que los primeros ordenadores personales de gran difusión fueron basados en microprocesadores de 8 bits, tomó cierta importancia la agrupación de datos en grupos de 8 bits. De ahí que, aunque los sistemas actuales manipulen unidades con más bits, se ha conservado este conjunto como agrupación básica.
Un byte es una unidad de información compuesta por 8 bits. Puede representar cualquier tipo de información: un valor numérico, un carácter alfanumérico, una dirección (posición) de memoria, etc. Los sistemas actuales manipulan agrupaciones de bits compuestas por un múltiplo de 8 (16, 32 bits,...).
- ✓ **Múltiplos (Kb, Mb y Gb):** Generalmente, los sistemas de almacenamiento de información son capaces de retener cantidades elevadas de bytes, por lo que se introducen una serie de unidades para evitar manejar números elevados.

Cuando se habla de sistemas de almacenamiento de información (memoria, discos flexibles, discos duros,...), se expresa la capacidad de almacenamiento en términos del número de bytes que es capaz de almacenar. Así, por ejemplo, si una memoria puede almacenar 300 bytes como máximo, diremos que su capacidad es de 300 bytes.

Unidad	Nº bytes	
	Exactos	Aproximados
Kilobyte (Kb)	1024 2^{10}	1.000 bytes
Megabyte (Mb)	1024×1024 2^{20} 1.048.576	1.000 Kb 1.000.000 bytes
Gigabyte (Gb)	$1024 \times 1024 \times 1024$ 2^{30} 1.073.741.824	1.000 Mb 1.000.000 Kb 1.000.000.000 bytes
Terabyte (Tb)	$1024 \times 1024 \times 1024 \times 1024$ 2^{40} 1.099.511.627.776	1.000 Gb 1.000.000 Mb 1.000.000.000 Kb 1.000.000.000.000 bytes

Las unidades *Kilo*, *Mega* y *Giga* son unidades de magnitud y pueden referirse a cualquier objeto, por ejemplo un Kilocaja, sería un Kilo de cajas. En el sistema decimal, *Kilo* representa al valor multiplicador 1.000. En el sistema binario, se suele aproximar de manera que: 1 Kb = 1000 bytes. Pero realmente 1 Kilobyte son 1024 bytes. ¿Por qué 1.024 y no 1.000? Porque 1024 es la potencia entera de 2 más próxima a 1000.

Con el *Mega* pasa algo parecido, un *Mega* no es 1.000.000 aunque tiene un valor parecido. Concretamente $1.024 \times 1.024 = 1.048.576$, por lo que un *Mega* de coches serían, según el sistema binario, 1.048.576 coches

Entropía de una Fuente de Información: H

Es un parámetro que nos permite determinar el contenido promedio de información de una fuente o un mensaje en particular. En un proceso de comunicación, transmitimos usualmente secuencias largas de símbolos, y estamos más interesados en el contenido promedio de información que la fuente produce, que en la información contenida en cada símbolo. Luego:

$$H(X) = E[I(x_i)] = \sum_{i=1}^m P(x_i)I(x_i) = -\sum_{i=1}^m P(x_i)\log_2 P(x_i) \text{bit / simbolo}$$

• Ruido (tipos y características) (R, E, RP)

Ruido.

Señales indeseables del canal de comunicación.

Es la aparición de una señal no deseada, de naturaleza auditiva en el medio de transmisión, la línea se puede ver afectada por cuatro tipos de ruidos:

- ✓ *Ruido Térmico*.- Es inevitable y es debido a las altas o bajas temperaturas a las que está sometida la línea de transmisión, de modo que aunque sea pequeño siempre va a existir.
- ✓ *Ruido Impulsivo*.- Es el ruido ocasionado por agentes ajenos a la línea de transmisión. Por ejemplo, un fenómeno meteorológico, el rayo. Es de corta duración y de gran amplitud.
- ✓ *Diáfonia*.- Aparece a partir del hecho físico de que al circular una corriente eléctrica por un conductor se genera un campo magnético en torno al mismo y si cerca de este conductor se encuentra un segundo conductor que también está transmitiendo, el campo magnético generado por el primero se superpone con el segundo. Ej. cuando hablamos por teléfono y se oye a otra persona. La forma de evitar este tipo de ruido es o bien, apantallando los cables o entrelazando unos cables con otros.
- ✓ *Intermodulación*.- Este ruido está relacionado con el comportamiento no lineal del medio de transmisión y consiste en la aparición de frecuencias que son suma o diferencias de la señal que se transmite.
- ✓ *Eco*.- Consiste en la aparición de una señal no deseada de las mismas características pero atenuada y retrasada en el tiempo respecto de esta

• Atenuación (R, E, RP)

Pérdida de energía de la señal de comunicación.

Es el debilitamiento de la señal, debido a la resistencia eléctrica que presentan tanto el canal como los demás elementos que intervienen en la transmisión, el debilitamiento se manifiesta en un descenso de la amplitud de la señal transmitida. La atenuación no afecta por igual a todas las frecuencias, cuanta mayor frecuencia lleva una señal, más posibilidades tiene de producirse la atenuación.

• Teorema de Shannon (R)

Establece el límite de velocidad de transferencia de datos en un sistema real de comunicaciones

El teorema de capacidad máxima de un canal

En 1928 Harry Nyquist, un investigador en el área de telegrafía, publicó una ecuación llamada la *Razón Nyquist* que media la razón de transmisión de la señal en bauds. La razón de Nyquist es igual a $2B$ símbolos (o señales) por segundo, donde B es el ancho de banda del canal de transmisión. Así, usando esta ecuación, el ancho de banda de un canal telefónico de 3,000 Hz puede transmitido hasta $2 \times 3,000$ o 6,000 bauds o Hz.

Claude Shannon después de la investigación de Nyquist estudió el cómo el ruido afecta a la transmisión de datos. Shannon tomó en cuenta la razón *señal-a-ruido* del canal de transmisión (medido en decibeles o dB) y derivó el teorema de Capacidad de Shannon. $C = B \log_2(1+S/N) \text{ bps}$

Un típico canal telefónico de voz tiene una razón de señal a ruido de 30 dB ($10^{(30/10)} = 1000$) y un ancho de banda de 3,000 Hz. Si sustituimos esos valores en el teorema de Shannon:

$$C = 3,000 \log_2(1+1000) = 30,000 \text{ bps}$$

Debido a que $\log_2(1001)$ es igual al logaritmo natural de $\ln(1001)/\ln(2)$ y es igual a 9.97, el teorema nos demuestra que la capacidad máxima* de un canal telefónico es aproximadamente a 30,000 bps.

Debido a que los canales de comunicación no son perfectos, ya que están delimitados por el ruido y el ancho de banda. El teorema de Shannon-Hartley nos dice que *es posible transmitir información libre de ruido siempre y cuando la tasa de información no exceda la Capacidad del Canal.*

Así, si el nivel de S/N es menor, o sea la calidad de la señal es más cercana al ruido, la capacidad del canal disminuirá.

Esta capacidad máxima es inalcanzable, ya que la fórmula de Shannon supone unas condiciones que en la práctica no se dan. No tiene en cuenta el ruido impulsivo, ni la atenuación la distorsión. Representa el límite tiempo máximo alcanzable.

¿Cuánto nivel de S/N requeriríamos para transmitir sobre la capacidad del canal telefónico, digamos a 56,000 bps?

De la fórmula de Shannon;

$$C = B \log_2(S/N + 1) = bps \text{ bps} = B \log_2(10^{(dB/10)} + 1)$$

$$\begin{aligned} &\text{despejando los dB} \\ bps/B &= \log_2(10^{(dB/10)} + 1) \\ 2^{(bps/B)} &= 10^{(dB/10)} + 1 \\ 10^{(dB/10)} &= 2^{(bps/B)} - 1 \\ dB/10 &= \log_{10}(2^{(bps/B)} - 1) \\ dB &= 10 * \log_{10}(2^{(bps/B)} - 1) \end{aligned}$$

$$\begin{aligned} &\text{sustituyendo} \\ B &= 3,000 \text{ y } bps = 56,000 \\ dB &= 10 * \log_{10}(2^{(56,000/3,000)} - 1) \\ dB &= S/N = 56.2 \text{ dB} \end{aligned}$$

Lo que significa que si queremos rebasar el límite de Shannon debemos de aumentar el nivel de S/N.

• Canal de datos (R)

Canal: Ruta o camino por la que se transmite información desde la CPU hacia un periférico determinado, un elemento de memoria, etcétera.

1. Una ruta de comunicación. Se pueden multiplexar múltiples canales en un solo cable en ciertos entornos.
2. En IBM, la ruta específica entre computadores de gran tamaño (como los mainframes) y dispositivos periféricos conectados.
3. Tipo de conducto con cubierta móvil montado en la pared utilizada para soportar el cableado horizontal. El canal es lo suficientemente grande como para contener varios cables.

Canal de datos: Canal utilizado para la transmisión de datos.

Canal RDSI, de 16 kbps (BRI) o 64 kbps (PRI), full dúplex. En SNA, dispositivo que conecta un procesador y el almacenamiento principal con periféricos.

Canal de datos hit (clear channel)

El canal de datos HIT es la interconexión de dos computadoras o redes de computadoras situados en diferentes países utilizando canales de 64 Kbps. En fibra óptica, reduciendo al máximo los retardos se logra el empleo óptimo de la comunicación. Para procesos en línea, la velocidad de respuesta permite el acceso a programas y archivos de forma remota con ejecuciones que no difiere del obtenido en una comunicación local.

El producto se puede trabajar en la modalidad de canal dedicado, en este caso la comunicación es permanente dentro de determinado horario. Esto para acceso remoto y permanente de aplicaciones y manejo de archivos. Otra modalidad es la conexión al momento de necesitar la transmisión, esta forma permite una mayor economía a cambio de efectuar el procedimiento de conexión más frecuentemente.

Características

- 1- No requiere instalación de Última milla pues utiliza canales asignados por Telecom.

- 2- El costo es proporcional a su utilización, si se decide esta opción.
- 3- La transmisión se hace por fibra óptica y en tiempo real
- 4- Los canales internacionales son proveídos por MCI WORLDCOM.
- 5- El sitio de destino no requiere última milla en la mayoría de casos.
- 6- La mayoría de trabajo de instalación hace referencia al Software sin incidir en el hardware.
- 7- Cada instalación requiere sin embargo de un estudio previo
- 8- Permite conexión a diferentes destinos
- 9- Puede ser utilizado para canales de voz y fax

Condiciones comerciales

- 1- No tiene costo de instalación en la mayoría de casos, los costos de nuevas líneas y gastos por motivo de requerimientos de software son por cuenta del usuario.
- 2- El usuario cancela el monto que facture la duración de la transmisión de datos que llagara relacionada mensualmente en una factura que discrimina el tiempo y el lugar a donde se hizo la transmisión.

2. Códigos

• Códigos de redundancia (*m* de *n*) (R)

Código es la correspondencia entre un conjunto de símbolos por números. Codificación es el proceso de hacer la correspondencia entre un símbolo y un número.

Propiedades:

- ✓ Uniformidad: un código es uniforme si a cada símbolo fuente le corresponde una palabra código. Los códigos que cumplen esta propiedad también se les denomina código bloque
- ✓ No singularidad: Un código uniforme es no singular si a cada símbolo fuente le corresponde palabras de código distintas

• Códigos binarios (R, E)

CÓDIGO DE REDUNDANCIA CÍCLICA (CRC)

Los códigos de redundancia cíclica (CRC) son muy utilizados en la práctica para la detección de errores en largas secuencias de datos. Se basan en representar las cadenas de datos como polinomios. El emisor realiza ciertas operaciones matemáticas antes de enviar los datos. El receptor realizará, a la llegada de la transmisión, una división entre un polinomio convenido (polinomio generador). Si el resto es cero, la transmisión ha sido correcta. Si el resto es distinto significará que se han producido errores y solicitará la retransmisión al emisor.

El tratamiento que tienen los mensajes están explicados mediante planteamientos matemáticos, mas directamente de manera polinomial, ya que estos mensajes son tratados como si fueran Polinomios, cuyos coeficientes son 1 y 0.

- ✓ El método consiste básicamente en que el emisor trata la información a transmitir como una cadena de bits, y la convierte en un polinomio binario $p(x)$ (donde los valores de los coeficientes son cero o uno). Supongamos que nuestro $p(x)=1100001(x^6 + x^5 + 1)$
- ✓ Este polinomio binario se multiplica por el grado del polinomio generador $g(x)$ que es conocido tanto por el emisor como por el receptor. Supongamos que nuestro $g(x)=1011(x^3 + x + 1)$.
- ✓ Posteriormente se realiza la multiplicación del $p(x)$ por el grado del polinomio generador $g(x)$. En nuestro ejemplo nos quedaría lo siguiente: $p(x)=1100001000$ resultado de: $(x^6 + x^5 + 1) * x^3$

Luego se obtiene el residuo de la división $p(x)/g(x)=101$, el residuo de esta operación es lo que conformará el CRC.

- ✓ Los coeficientes (ceros o unos) de este resto se añaden a $p(x)$. Y nos queda $p(x)=1100001101$
- ✓ El receptor tratará a toda la trama como un polinomio (información y código de detección de errores conjuntamente) y lo dividirá por el mismo polinomio generador $g(x)$ que uso el emisor. Si el resto de la división es cero, no se habrán detectado errores en la transmisión.

División de Comprobación:

Existen polinomios estándares usados por protocolos utilizados para generar **CRC**.

$$* \text{ CRC - 12 : } X^{10} + X^9 + X^5 + X^4 + X + 1$$

$$* \text{ CRC - 16 : } X^{12} + X^{11} + X^3 + X^2 + X + 1$$

$$* \text{CRC - 32 : } X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$$

El CRC - 12 es usado para transmisión de flujos de caracteres de 6-bits y genera una FCS de 12 bits.
El CRC - 16 es usado para caracteres de 8 bits y genera una FCS de 16 bits.

Éstos son usados ampliamente en EUA y Europa, dando una adecuada protección para la mayoría de las aplicaciones.

Éste es usado por el comité de estándares de red local (IEEE-802) y en algunas aplicaciones DOD.

CODIGO BINARIO

Un código Binario es un grupo de n bits que adopta hasta 2^n combinaciones distintas de números de 1 y 0 y donde cada combinación representa un elemento del conjunto del conjunto que se codifica. Por ejemplo, un conjunto de 4 elementos puede representarse mediante un código de 2 bits, donde a cada elemento se le asigna una de las siguientes combinaciones de bits 00, 01, 10 u 11. Un conjunto de 8 elementos requiere un código de 3 bits, un conjunto de 16 elementos requiere un código de 4 bits y así sucesivamente. Un código binario tendrá algunas combinaciones de bits nos asignadas si el numero de elementos en el conjunto no es una potencia múltiplo de 2. los 10 dígitos decimales constituyen un conjunto de esa naturaleza. Un código binario que distingue entre 10 elementos debe contener al menos 4 bits, pero 6 de sus combinaciones no se asignaran. Pueden obtenerse muchos códigos diferentes al arreglar 4 bits en 10 combinaciones distintas. La asignación de bits que se usa con mayor frecuencia para los dígitos decimales es la que se muestra a continuación en la siguiente tabla. Este código particular se llama decimal codificado en binario (binary-coded decimal) y por lo general se representa con su abreviatura BCD. En ocasiones se usan otros códigos decimales.

Numero Decimal	Números Decimales codificados en Binario (BCD)
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Es muy importante comprender la diferencia entre la conversión de números decimales binarios y la codificación binaria de números decimales. Por ejemplo cuando se convierte a un número binario, el numero decimal 99 se representa con la serie de bits 1100011, pero cuando se representa en BCD, se vuelve 1001 1001. La única diferencia entre un número decimal que se representa por los símbolos de dígitos con los que estamos familiarizados 0, 1, 2,..., 9 y los símbolos BCD 0001, 0010,..., 1001 esta en los símbolos que se utilizan para representar los dígitos; el numero por si mismo es igual.

Los códigos binarios se caracterizan por tener un alfabeto o código fijo, formado por 2 símbolos {1, 0}

Definiciones y Propiedades:

- ✓ *Ponderados*: a cada digito binario se le asigna un peso y a cada palabra código la suma de los pesos de los dígitos binarios con valor uno, siendo el resultado igual al numero decimal al que representa.
- ✓ *Distancia*: se define como la menor de las distancias entre 2 cualesquiera de sus palabras código; son adyacentes si su distancia es uno es decir sólo difieren en un bit.
- ✓ *Continuos*: palabras consecutivas son adyacentes
- ✓ *Cíclicos*: además de ser continuos, la primera y ultima palabra de código también son adyacentes.
- ✓ *Denso*: tenga una longitud de palabra de n bits esta formado por 2^n palabras de código.
- ✓ *Autocomplementarios*: palabra y su complementada suman a N. Los códigos con esta propiedad posibilitan efectuar más fácilmente las operaciones de resta mediante el complemento a N.

Definición de distintos tipos:

- ✓ Código natural
- ✓ Código no cílicos ni continuos
 - BCD Natural 8421
 - BCD Aiken 2421
 - BCD Aiken 5421
 - BCD 642-3
 - BCD exceso a 3
- ✓ Códigos cílicos y continuos
 - Código Gray
 - Código Johnson

En ocasiones se utilizan otros códigos binarios para números decimales y caracteres alfanuméricos. Las computadoras digitales también emplean otros códigos binarios para aplicaciones especiales.

CÓDIGO GRAY.

Los sistemas digitales pueden procesar datos solo de manera discreta. Muchos sistemas físicos proporcionan salida continua de datos. Los datos deben convertirse a su forma digital antes de que puedan ser utilizados por una computadora digital. La información continua o análoga se convierte a su forma binaria mediante un convertidor analógico/digital. El código binario reflejado o código Gray se utiliza en ocasiones para los datos convertidos a digitales. La ventaja del código Gray sobre los números binarios directos es que solo cambia un bit conforme avanza de número al siguiente. Una aplicación típica de este código se presenta cuando los datos análogos se representan mediante un cambio continuo de una posición relevante. La parte relevante se divide en segmentos y a cada segmento se le asigna un número. Si se hace que segmentos adyacentes correspondan a números en código Gray adyacentes, se reduce la ambigüedad cuando la posición relevante está en la línea que separa dos segmentos cualesquiera.

En ocasiones se utilizan contadores en código Gray para proporcionar las secuencias de temporización que controlan las operaciones en un sistema digital. Un contador en código Gray es aquel cuyos flip-flops recorren una secuencia de estado según se especifica en la siguiente tabla. Los contadores de código Gray eliminan la ambigüedad durante el cambio de un estado del contador al siguiente porque sólo cambia un bit durante la transición de estado.

Código Binario	Equivalente Decimal	Código Binario	Equivalente Decimal
0000	0	1100	8
0001	1	1101	9
0011	2	1111	10
0010	3	1110	11
0110	4	1010	12
0111	5	1011	13
0101	6	1001	14
0100	7	1000	15

Código Gray de 4 bits

La formación se realiza por reflexión del código n-1 bits, repitiendo simétricamente las combinaciones de éste y añadiendo a la izquierda un bit.

1 bit

1
0

2 bit

0	0
0	1
1	1
1	0

3 bit

0	0	0
0	0	1
0	1	1
0	1	0
1	1	0
1	1	1
1	0	1
1	0	0

Propiedades del código Gray

Propiedad	Código Gray
Ponderado	No
Distancia de Código	1
Continuo	Si
Cíclico	Si
Denso	Si
Autocomplementario	No

Por ser cíclico, el código Gray se utilizará para la síntesis Funciones lógicas en circuitos.

Conversión

- ✓ Conversión de código binario natural a Gray
 - $G(t)' = B(t) \text{ xor } B(t+1)$
- ✓ Conversión de código Gray a binario natural:
 - $B(t) = B(t+1) \text{ xor } G(t)$

Paridad

- ✓ La paridad de un código binario está relacionada con el número de unos que tiene éste.
 - Si el número de unos es par, se dice es de **paridad par**
 - Si es impar se dice es de **paridad impar**
- ✓ Si se añade un bit de paridad a un código denso, la distancia de código pasa a ser 2, y se podrán detectar errores de un solo bit.
 - Transversal: Se añade un bit a la izquierda de cada código
 - Longitudinal o de bloque: Al final de una sección de códigos se añade un código con la paridad por columnas de bits de todos los códigos de la secuencia
 - O-Exclusivo: Igual que a transversal
 - Operador suma: Hace que la suma de todos los códigos de la secuencia, incluido el código de paridad sea 0.

CÓDIGOS DE HAMMING

Son correctores de errores, cuya distancia mínima es 3, que permiten detectar errores de 2 bits y corregir errores de 1 bit.

Debe de cumplir la condición $2^p \geq n+p+1$

Otros Códigos Decimales: Los códigos binarios para dígitos decimales necesitan un mínimo de 4 bits. Pueden formularse muchos códigos diferentes al arreglar 4 o más bits en 10 posibles combinaciones diferentes. En la siguiente tabla se muestran unas cuantas posibilidades.

Digito Decimal	BCD 8421	2421	De exceso 3	Gray con Exceso 3
0	0000	0000	0011	0010
1	0001	0001	0100	0110
2	0010	0010	0101	0111
3	0011	0011	0110	0101
4	0100	0100	0111	0100
5	0101	1011	1000	1100
6	0110	1100	1001	1101

7	0111	1101	1010	1111
8	1000	1110	1011	1110
9	1001	1111	1100	1010
Combi-naciones De bit Sin usar	1010	0101	0000	0000
	1011	0110	0001	0001
	1100	0111	0010	0011
	1101	1000	1101	1000
	1110	1001	1110	1001
	1111	1010	1111	1011

Cuatro Códigos Binarios diferentes para el Dígito Decimal

Ya se ha visto lo que es un BCD (decimal codificado en binario). Utiliza una asignación directa del equivalente binario del código. Las 6 combinaciones listadas de bits que no se usan no tienen ningún valor cuando se utiliza BCD, igual que la letra H no representa nada cuando se anotan símbolos de dígitos decimales. Por ejemplo, decir que 1001 1110 es un número decimal en BCD es como decir que 9H es un número decimal en la denominación de signos convencional. Ambos casos tienen un signo inválido y por lo tanto representan un número sin significado.

Una desventaja de utilizar BCD es la dificultad que se encuentra cuando se va a capturar el complemento a 9 del número. Por otra parte, el complemento a 9 se obtiene fácilmente con los códigos 2421 y de 3 en exceso. Estos 2 códigos tienen una propiedad autocomplementaria que significa que todos los complementos a 9 de un número decimal, cuando se representan en uno de estos códigos, se obtienen con facilidad al cambiar los 1 a 0 y los 0 a 1. Esta propiedad es útil cuando se realizan operaciones aritméticas en representación del complemento con signo.

El 2421 es un ejemplo de un código ponderado. En el que los bits proporcionan el dígito decimal. Por ejemplo, una combinación de bits 1101, cuando se posiciona con los dígitos 2421 respectivos, proporciona el equivalente decimal de $2 \times 1 + 4 \times 1 + 2 \times 0 + 1 = 7$. Pueden asignarse al código BCD las ponderaciones 8421 y por esta razón en ocasiones se les llama código 8421.

El código de exceso 3 es un código decimal que se ha utilizado en computadoras anteriores. Este es un código sin ponderación. Su asignación de código binario se obtiene del número binario equivalente a BCD que le corresponde después de la adición del 3 binario (0011).

- **Códigos algebraicos (R, E)**

- **Caracteres alfanuméricos y de control en ASCII y EBCDIC (R, E)**

CODIGO ASCII

Después Del código Baudot, se hizo necesario aumentar el conjunto de caracteres, apareciendo códigos de 6 bits capaces de manejar 64 símbolos distintos (el código FIELDATA es un ejemplo de ellos). Más tarde se paso a códigos de 7 bits capaces de manejar hasta 128 caracteres, entre los cuales podemos citar el código ASCII (American Standard Code for Information Interchange), de 7 bits, también denominado código numero 5 de CCITT.

Los ordenadores de tipo PC utilizan el Código ASCII (american standart code for information interchange). El código ASCII representa con 7 bits las letras mayúsculas y minúsculas, los números, signos de puntuación y caracteres de control. Existe también un código ASCII extendido de 8 bits por carácter, el cual incorpora una gran variedad de símbolos gráficos. El código ASCII extendido es el utilizado habitualmente por el sistema operativo DOS del PC.

En la Actualidad se utiliza el código ASCII de 8 bits, en que aparecen los 128 caracteres del código anterior más otros 128 caracteres, donde cada fabricante puede hacer su propia ampliación del conjunto de caracteres a manejar. En las transmisiones entre equipos diversos no es recomendable la utilización de caracteres de estas ampliaciones por los posibles errores de interpretaciones incorrectas que puedan producirse.

Código ASCII de 7 bits

bits 3210	654 000	001	010	011	100	101	110	111
0000	NUL	DEL	SP	0	@	P		p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(8	H	X	h	x
1001	HT	EM)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

Caracteres de Control

NUL	Null (carácter nulo)	DC1	Device control 1 (control de dispositivo 1)
SOH	Start of heading (comienzo de cabecera)	DC2	Device control 2 (control de dispositivo 2)
STX	Start of text (comienzo de texto)	DC3	Device control 3 (control de dispositivo 3)
ETX	End of text (final de texto)	DC4	Device control 4 (control de dispositivo 4)
EOT	End of transmission (fin de transmisión)	NAK	Negative acknowledge (trans. negativa)
ENQ	Enquiry (petición de transmisión)	SYN	Synchronous idle (espera síncrona)
ACK	Acknowledge (reconocimiento de trans.)	ETB	End of transmission block (fin bloque de transmisión)
BEL	Bell (señal audible, timbre o alarma)	CAN	Cancel (cancelar)
BS	Backspace (retroceso)	EM	End of medium (fin del medio)
HT	Horizontal tabulation (tab. horizontal)	SUB	Substitute (sustitución)
LF	Line feed (avance de página)	ESC	Escape (escape)
VT	Vertical tabulation (tab. vertical)	FS	File separator (separador de archivos)
FF	Form feed (avance de página)	GS	Group separator (separador de grupos)
CR	Carriage return (retorno de carro)	RS	Record separator (separador de registros)
SO	Shift out (quitar desplazador de bits)	US	Unit separator (separador de unidades)
SI	Shift in (poner desplazador de bits)	DEL	Delete (borrar)

CODIGO EBCDIC

Otro código similar al código ASCII de 8 bits es el EBCDIC (Extended Binary Coded Decimal Interchange Code), utilizado por los ordenadores de IBM de los de la serie IBM PC (miniordenadores y mainframes), diferencia las letras mayúsculas de las minúsculas, contienen mas signos de puntuación que el código Baudot, además de todo un complejo juego de comandos de control. En este caso se utilizan 8 bits para representar cada símbolo.

En general, podemos decir que los terminales fabricados en los años treinta y cuarenta utilizaban el código BAUDOT de 5 bits por carácter, los terminales fabricados entre los años cincuenta y setenta utilizaban el código ASCII de 7 bits por carácter, y los terminales más modernos suelen utilizar el código ASCII extendido de 8 bits por carácter.

	0000 0	0001 1	0010 2	0011 3	0100 4	0101 5	0110 6	0111 7	1000 8	1001 9	1010 A	1011 B	1100 C	1101 D	1110 E	1111 F
0000 0	NUL DLE	DLE DS	DS SP	& 64	- 80	96	112	128	144	160	176	{ 192	}\br/>208	224	240 0	
0001 1	SOH 1	DCI 17	SOS 33	49	/ 65	81	97	113	a 129	j 145	k 161	A 177	J 193	1 209	225 241	
0010 2	STX 2	DC2 18	FS SYN	50	66	82	98	114	b 130	k 146	s 162	B 178	K 194	S 210	226 242	
0011 3	ETX 3	TM 19	RES 35	51	67	83	99	115	c 131	l 147	t 163	C 179	L 195	T 211	3 227	
0100 4	PF 4	RES 20	BYP 36	PN 52	68	84	100	116	d 132	m 148	u 164	D 180	M 196	U 212	4 228	
0101 5	HT 5	NL 21	LF 37	RS 53	69	85	101	117	e 133	n 149	v 165	E 181	N 197	V 213	5 229	
0110 6	LC 6	BS 22	ETB 38	UC 54	70	86	102	118	f 134	o 150	w 166	F 182	O 198	W 214	6 230	
0111 7	DEL 7	IL 23	ESC 39	EOT 55	71	87	103	119	g 135	p 151	x 167	G 183	P 199	X 215	7 231	
1000 8	CAN 8	24	40	56	72	88	104	120	h 136	q 152	y 168	H 184	Q 200	Y 216	8 232	
1001 9	RLF 9	EM 25	41	57	73	89	105	\br/>105	i 121	r 137	z 153	I 169	R 185	Z 201	9 233	
1010 A	SMM 10	CC 26	SM 42	cent 58	! 74	90	106	122	:	138	154	170	186	202	218	234
1011 B	VT 11	CU1 27	CU2 43	CU3 59	. 75	\$ 91	, 107	123	# 139	155	171	187	203	219	235	251
1100 C	FF 12	IFS 28	DC4 44	< 60	* 76	% 92	@ 108	124	,	140	156	172	188	204	220	236
1101 D	CR 13	IGS 29	ENQ 45	NAK (61) 77	- 93	109	125	' 141	157	173	189	205	221	237	253
1110 E	SO 14	IRS 30	ACK 46	+ 62	; 78	> 94	= 110	126	= 142	158	174	190	206	222	238	254
1111 F	SI 15	IUS 31	BEL 47	SUB 63	 79	- 95	? 111	127	" 143	159	175	191	207	223	239	255

Tabla del código EBCDIC

Caracteres de control EBCDIC:

NUL	Nulo	IGS	Separador para intercambio de grupos
SOH	Comienzo de cabeza	IRS	Separador para intercambio de registros
SOT	Comienzo de texto	IUS	Separador para intercambio de unidad
EOT	Final de texot	DS	Selección de dígito
PF	Perforadora desconectada	SOS	Comienzo de significado
HT	Tabulación horizontal	FS	Separador de campo
LC	Mínuscula	BYP	Desviar
DEL	Eliminar, borrar	LF	Alimentación de línea
RLF	Alimentación de linea invertida	ETB	Final de bloque de transmisión
SMM	Comienzo mensaje manual	ESC	Escape
VT	Tabulación vertical	SM	Fijar modo
FF	Alimentación de hoja	ENQ	Solicitud, petición
CR	Retorno de carro	ACK	Acuse de recibo
SO	Fuera de código	BEL	Pitido
SI	Dentro de código	SYN	Sincronización
DLE	Escape del enlace de datos	PN	Perforadora conectada
TM	Marca de cinta	RS	Detener lectora
RES	Restaurar	UC	Mayúsculas
NL	Pasar a línea siguiente	EOT	Fin de transmisión
BS	Retroceso de un espacio	NACK	Acuse de recibo negativo
IL	sin función	SUB	Sustituir
CAN	Cancelar	DC <i>i</i>	Control dispositivo <i>i</i>
EM	Final de soporte	CU <i>i</i>	Control usuario <i>i</i>
CC	Control del cursor		
SP	Espacio en blanco		
IFS	Separador para intercambio de archivos		

Cuando se utilizan caracteres alfanuméricos para el procesamiento interno de datos en una computadora (no para su transmisión) es conveniente utilizar un código de 6 bits para representar 64 caracteres. Un código de 6 bits puede

especificar las 26 letras mayúsculas del alfabeto, los números del 0 al 9 y hasta 28 caracteres especiales. Este conjunto de caracteres por lo general es suficiente para el procesamiento de datos. Utilizar menos bits para codificar caracteres tiene la ventaja de reducir el espacio de memoria que se necesita para almacenar grandes cantidades de datos alfanuméricos.

• Códigos de compresión (R)

Compresión orientada al carácter.

Se basa en el uso de un carácter especial que indica que se ha realizado la compresión. Estas técnicas pueden utilizarse de forma aislada o combinadas entre sí.

A continuación se expondrán los siguientes métodos: Eliminación de caracteres blancos, Bit mapping, Run length, Half-byte, Packing, Codificación dicotómica.

Eliminación de caracteres blancos: Se recorre la secuencia de datos para encontrar una secuencia repetida de caracteres blancos. Cuando se encuentra una secuencia de este tipo se sustituye por:

- ✓ Un carácter especial que indica que se está comprimiendo.
- ✓ Un número que indica la cantidad de blancos que se están comprimiendo.

Solo se aplica el método cuando el número de blancos consecutivos es superior a 2.

Ejemplo de compresión en el emisor

La cadena de entrada es:

kmqØØØØØØ bgpØØswØØØØj

Una vez realizada la compresión, la cadena resultante será:

kmqSc6bgpØØswSc4j

Donde Ø representa un carácter blanco y Sc es el carácter especial indicador de compresión.

En el *proceso de descompresión*, el receptor recorre la cadena de datos que llega a través del canal. Cuando encuentra un carácter especial que indique compresión sabrá que en esa posición se ha realizado una compresión y que el siguiente carácter indica el número de caracteres blancos que fueron comprimidos y de esta forma podrá reconstruir la cadena original.

Ejemplo de descompresión en el receptor

La cadena recibida a través del canal es:

mpØØgtSc6astSc4th

Una vez realizada la descompresión obtenemos que la cadena original era:

mpØØgtØØØØ ØØastØØØØth

Donde Ø representa un carácter blanco y Sc es el carácter especial indicador de compresión.

Esta técnica de compresión es útil para la compresión de documentos que contienen muchas indentaciones.

Bit mapping: Es una técnica efectiva cuando en la secuencia a comprimir aparecen en gran proporción datos de un determinado tipo: numéricos o caracteres blancos, por ejemplo.

Se utiliza un bit map para indicar:

- ✓ La presencia o ausencia de caracteres de datos y/o
- ✓ El hecho de que ciertos caracteres de datos se han encontrado anteriormente y deben ser tratados de nuevo para devolverles a su estado original.

El bit map tendrá tantos bits como el tamaño del carácter (normalmente serán 8 bits).

Ejemplo de compresión en el emisor

En este ejemplo se realizará la compresión de caracteres blancos.

La cadena de entrada es:

kmqØØØØØØbgp ØØswØØØØj

Dividimos la cadena de entrada en grupos de 8 caracteres (considerando que la representación interna de cada carácter se realiza con 8 bits).

kmqØØØØØ ØbgpØØsw ØØØØØj

Para cada grupo de 8 caracteres se calcula su bit map. En cada posición de un bit map se añade:

0 si el carácter correspondiente es un blanco

1 se el carácter correspondiente no es un blanco

Después de cada bit map se añaden los caracteres distintos del carácter blanco correspondientes a ese bit map.

De esta forma, la cadena resultante es:

11100000 kmq 01110011 bgpsw 00001000 j

Si el último grupo de caracteres es inferior a 8 caracteres se añaden blancos hasta completar 8 caracteres. En este ejemplo hemos reducido una cadena de 21 caracteres a una cadena de 12 caracteres.

En el *proceso de descompresión*, el receptor toma el primer carácter de la cadena que llega a través del canal y lo considera como un bit map. Al interpretar el bit map, por cada bit que es igual a 1 el receptor añade a la cadena resultante el siguiente carácter de la cadena recibida, y por cada bit que es igual a 0, el receptor le añade un blanco a la cadena resultante. Una vez finalizada la interpretación del bit map, el receptor toma el siguiente carácter de la cadena recibida interpretándolo como un bit map y repitiendo el proceso. De esta forma se obtiene la cadena original a partir de la cadena recibida a través del canal.

Ejemplo de descompresión en el receptor

La cadena recibida a través del canal es la siguiente:

11001100 mpgt 00001110 ast 00011000 th

Realizando la descompresión, la cadena original obtenida a partir de la cadena recibida es:

mpØØgtØØØØØØast ØØØØth

Run length: Esta técnica es una generalización del método de eliminación de blancos. Con esta técnica se puede reducir cualquier secuencia de caracteres cuando el nivel de ocurrencia es de tres o más caracteres iguales consecutivos.

Cuando se encuentran 3 o más caracteres iguales consecutivos lo que se hace es sustituir esta secuencia por:

- ✓ un carácter especial indicador de compresión
- ✓ el carácter que se comprime
- ✓ número que indica la cantidad de caracteres que se comprimen

Ejemplo de compresión en el emisor

La cadena de entrada es:

fghhhijerØØØØØ ØØØØtttrrrrrywjad

Una vez realizada la compresión, la cadena resultante será:

fgSch4jjerScØ9Sct3Scr5ywjad

Donde Ø representa un carácter blanco y Sc es el carácter especial indicador de compresión

En el *proceso de descompresión*, el receptor recorre la cadena de datos que llega a través del canal. Cuando encuentra un carácter especial que indique compresión sabrá que en esa posición se ha realizado una compresión y que el siguiente carácter indica el carácter que ha sido comprimido y a continuación aparece el número que indica cuantos caracteres fueron comprimidos y de esta forma podrá reconstruir la cadena original.

Ejemplo de descompresión en el receptor
La cadena recibida a través del canal es la siguiente:
gtScr5juliScØ4jklScp3nj

Una vez realizada la descompresión obtenemos que la cadena original era:
gtrrrrjuliØØØØ jklppphj

Half Byte Parking: Esta técnica de compresión puede ser considerada como una derivación de la técnica Bit mapping. Se aplica a símbolos codificados en ASCII o EBCDIC. Para su aplicación se escogen del conjunto de caracteres disponibles aquellos que tienen una parte igual (normalmente los 4 primeros bits iguales).

Esta técnica solo se aplica cuando aparecen cuatro o más caracteres consecutivos susceptibles de ser comprimidos, es decir, que pertenecen al conjunto de símbolos seleccionados.

Cuando se encuentran una secuencia de caracteres consecutivos susceptibles de ser comprimidos lo que se hace es sustituir esta secuencia por:

- ✓ Un carácter especial indicador de compresión
- ✓ Un contador que indica el número de caracteres que han sido comprimidos

De cada carácter comprimido se toman solo sus 4 bits menos significativos. El número de caracteres consecutivos que pueden ser comprimidos dependerá del número de bits elegido para representar el contador. Si el contador se representa mediante 4 bits, se podrán comprimir hasta 15 caracteres consecutivos. Si el contador se representa mediante 8 bits se podrán comprimir hasta un máximo de 255 caracteres consecutivos.

Con los grupos de 4 bits procedentes de los caracteres que son comprimidos se formarán nuevos caracteres de 8 bits que serán transmitidos a través del canal.

Ejemplo de compresión en el emisor

En este ejemplo se comprimirán caracteres numéricos pertenecientes al código EBCDIC.

La cadena de entrada es:

9001922

Que codificada en EBCDIC será:

1111 1001 1111 0000 1111 0000 1111 0001 1111 1001 1111 0010 1111 0010

Una vez realizada la compresión obtenemos:

Sc 0111 1001 0000 0000 0001 1001 0010 0010

Donde Sc es el carácter especial indicador de compresión y el primer grupo de 4 bits es el contador que indica el número de caracteres que se han comprimido (en este caso es un 7)

En el *proceso de descompresión*, el receptor recorre la cadena de datos que llega a través del canal. Cuando encuentra un carácter especial que indique compresión sabrá que en esa posición se ha realizado una compresión. A continuación lee el siguiente carácter de la cadena recibida, y dependiendo de la convención utilizada para el tamaño del contador se dará una de estas dos situaciones:

- ✓ Si el contador se representa con 4 bits, se divide el carácter en 2 grupos de 4 bits. El primer grupo se interpreta como el contador y el segundo grupo se interpreta como los 4 bits menos significativos del primer carácter comprimido.
- ✓ Si el contador se representa con 8 bits, el carácter se interpreta como el contador.

A continuación se irán leyendo los caracteres de la cadena recibida y cada uno se dividirá en dos grupos de 4 bits que se interpretarán como los 4 bits menos significativos de cada carácter comprimido, hasta haber obtenido un número de caracteres igual al indicado por el contador.

De esta forma se obtiene la cadena original a partir de la cadena recibida a través del canal.

Ejemplo de descompresión en el receptor

En este ejemplo se realizará la descompresión de una cadena que incluye caracteres numéricos comprimidos pertenecientes al código EBCDIC.

La cadena recibida a través del canal es la siguiente:

1001 0010 1010 0011 1001 0101 Sc 0111 1001 0000 0000 0001 1001
0010 0010 0110 1111

Consideramos que el contador está formado por 4 bits. En este caso el contador es 0111 lo que implica que se han comprimido 7 caracteres numéricos.

Una vez realizada la descompresión obtenemos que la cadena original era:

1001 0010 1010 0011 1001 0101 1111 1001 1111 0000 1111 0000 1111 0001
1111 1001 1111 0010 1111 0010 0110 1111

Que interpretado en código EBCDIC es:

ktn9001922?

Codificación dicotómica: Esta técnica de compresión consiste en sustituir cada par de caracteres por un carácter especial. De esta forma teóricamente se podría reducir el número de caracteres transmitidos a la mitad. Sin embargo el número de caracteres especiales que podemos utilizar es limitado, con lo que no será posible poder sustituir cualquier par de caracteres por un carácter especial.

Esto nos lleva a la conclusión de que para maximizar la compresión es necesario utilizar los caracteres especiales para sustituir aquellos pares de caracteres que tienen una mayor frecuencia de aparición.

Por ello, antes de aplicar esta técnica es necesario realizar un estudio previo del tipo de datos que se van a transmitir. En este estudio se obtendrán las frecuencias de aparición de cada posible par de caracteres.

Una vez obtenidas estas frecuencias se asignará un carácter de compresión a cada uno de los pares de caracteres que tengan una mayor frecuencia de aparición.

En el proceso de compresión, el emisor va comprobando los caracteres que forman la cadena que se quiere enviar, y cuando detecta un par de caracteres susceptibles de ser comprimidos, transmitirá el carácter especial asignado a ese par de caracteres.

Ejemplo de compresión en el emisor

La siguiente tabla muestra aquellos pares de caracteres que tendrán asociados un carácter especial.

par de caracteres	carácter especial
co	Sc
tr	Sx
ma	Sy
gs	Sz

La cadena de entrada es:

hmtrctrbjmajopgscowmgsf

Una vez realizada la conversión obtendremos:

hm Sx c Sx bj Sy jop Sz Sc wm Sz f

En el *proceso de descompresión*, el receptor va comprobando los caracteres de la cadena recibida a través del canal y cuando encuentra un carácter especial lo sustituye por el par de caracteres correspondiente a ese carácter especial. De esta manera se consigue obtener la cadena original a partir de la cadena que se recibe a través del canal.

Ejemplo de descompresión en el receptor

Para este ejemplo se utilizará la misma tabla que se utilizó en el ejemplo correspondiente a la compresión.

La cadena recibida a través del canal es la siguiente:

hjnScmSxpyaqSybSylk SzSzxx

Una vez realizada la descompresión obtenemos que la cadena original era:

hjncomtrpyaqmabmalkgsgsxx

3. Errores

• Naturaleza de los errores (R, E)

Se define como error el hecho de que un bit de un mensaje sufra una inversión durante su transmisión. Esto convierte un mensaje en ilegible y por tanto puede inutilizar todo el costoso proceso de transmisión. Los errores pueden ser debidos tanto al medio físico y sus imperfecciones, como a los elementos de conmutación y de conversión entre las diferentes formas del mensaje. Se llama tasa de error al cociente entre el número de bits erróneos recibidos y el de bits recibidos. Es un número que permite medir la fiabilidad de un medio de comunicación.

Hay dos problemas que resolver. Por un lado es útil poder corregir, basándose en el propio mensaje o en otra información, los errores producidos. Los métodos diseñados para este objetivo son llamados técnicas de corrección. Por otro lado es interesante, en caso de no poder corregir el error, darse cuenta al menos de que el mensaje es erróneo y tomar precauciones. A los métodos que avisan de los errores se les llama técnicas para la detección.

• Detección de errores (R, E)

Detección de errores: se introduce una redundancia en los bits de datos para detectar un error en caso de que este ocurra.

En ocasiones, los paquetes de datos al moverse de un nodo a otro pueden alterarse, generando errores en la información que transportan. Este tipo de errores deben evitarse para que la información que transporta la red sea confiable. Sin esta confiabilidad las personas (los usuarios) no utilizarían las redes.

La idea básica detrás de los esquemas de detección de errores en redes es adicionar información redundante al paquete, de tal forma que permita determinar si un error ha sido introducido mientras se llevaba de un nodo a otro. Un esquema de detección de errores que podemos imaginar es la transmisión completa de dos copias del mismo paquete. Si las dos copias son idénticas cuando lleguen al receptor, es muy probable que la información esté correcta. Si no son iguales, un error fue introducido en una de las copias (o en ambas) y deben ser descartadas. Este esquema de detección de errores es deficiente: se requieren n bits de corrección de errores para un mensaje de n bits y además, si por alguna razón, los errores ocurren en el mismo lugar en los dos paquetes, no serán detectados.

Afortunadamente se pueden hacer cosas mejores. En general, se puede proporcionar una capacidad de detección de errores más fuerte enviando sólo k bits redundantes en un mensaje de n -bits, donde $k \ll n$ (k mucho más pequeño que n). Por ejemplo, el frame de Ethernet puede llevar hasta 12000 bits (1500 bytes) y utiliza sólo 32 bits redundantes para detectar errores (usa un código de redundancia cíclica conocido como CRC-32).

Los bits adicionados al mensaje son redundantes pues no adicionan nueva información al mensaje sino que se calculan directamente del mensaje original utilizando un algoritmo bien definido. El nodo que transmite y el que recibe deben conocer exactamente el algoritmo utilizado. El transmisor del mensaje aplica el algoritmo al mensaje que desea enviar para generar los bits redundantes y envía el mensaje y los bits extras. Cuando el receptor aplica el mismo algoritmo al mensaje recibido, el resultado debe ser (en ausencia de errores) el mismo enviado por el transmisor. Al concordar los dos resultados, el receptor sabe, con una alta probabilidad, que no hubo errores durante el movimiento del paquete. En caso de no concordar, el receptor debe tomar la acción apropiada (descartar el paquete y esperar su retransmisión o, si le es posible, corregir los errores).

En general, los bits adicionados a un mensaje para detectar errores se llaman error-detecting codes. En ciertos casos específicos, cuando el algoritmo para crear el código se basa en la adición, reciben el nombre de checksums. Infortunadamente, la palabra checksum se usa de manera incorrecta, queriendo nombrar con ella cualquier código de detección de errores, incluyendo aquellos que no son sumas, como los CRCs. El checksum de Internet recibe el nombre correcto, pues es un chequeo de errores que utiliza un algoritmo de suma.

- **Algoritmos de detección de error: paridad, checksum, verificación de redundancia cíclica (R, E)**

Paridad simple.

Consiste en añadir a cada byte un bit adicional que indica si el número de unos que contiene el byte es par o impar.
La paridad puede ser par o impar.

Paridad par:

- ✓ si el numero de unos es par se añade un cero
- ✓ si el numero de unos (1's)es impar se añade un uno.

Paridad impar:- si el numero de unos es par se añade un uno

- ✓ si el numero de unos (1's) es impar se añade un cero.

Inconveniente: Si en una transmisión se altera un bit o un número impar de bits, se detectará el error, en cambio, si el número de errores es par no se detecta el error.

Paridad de bloque.

Consiste en organizar la información por bloque formando una tabla de m x n bits con los caracteres que componen el mensaje, después se calcula la paridad simple por filas y por columnas y se envía con el mismo bloque.

Ejemplo con paridad par.

1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0
1	0	1	0	1	0	1	0	0
1	0	1	0	1	0	1	1	1
0	1	1	1	0	1	0	1	x

Cuando el receptor recibe la información realiza el proceso de cálculo de paridad y compara con el resultado inicial y así detecta si hubo algún error localizando además donde ha sido. Solo funciona para un error por bloque.

Redundancia Cíclica.

Está basado en la propiedad de la división polinómica, en este caso a cada palabra se le asigna un polinomio.

$$110101 \rightarrow X^5 + X^4 + X^2 + 1$$

El método consiste en: al existir un polinomio conocido (polinomio generador) por el Emisor y el Receptor y que actúa como clave en la detección del error, llamamos M(x) al polinomio asociado a la palabra a transmitir, se multiplica por X^v , V es el grado del polinomio generador y al resultado de multiplicar estos factores se le divide entre el polinomio generador. Así: $X^v * M(x) G(x)$

La división da un cociente y un resto, el resto es la información redundante que se envía junto con la información necesaria para la detección de errores y se manda.

Cheksum (Suma de Verificación)

La suma de verificación es un pequeño entero utilizado para revisar si hay errores en la transmisión de datos de un maquina a otra. El software de protocolo, como en el caso del TCP, realiza una suma de verificación y le añade a un paquete cuando se transmite. En la recepción, el software de protocolo verifica el contenido de un paquete, al recalcular la suma de verificación y comparar el resultado con el valor enviado.

- **Códigos de corrección: Hamming y otros (R)**

Una vez el error se detecta en el receptor, se presenta una "conversación" entre el receptor y el transmisor, para que las tramas con errores sean nuevamente transmitidas.

Los códigos de corrección de errores tienen una eficiencia menor que los de detección para el mismo número de bits, y salvo que el medio de transmisión tenga muchos errores no salen rentables. Por este motivo los códigos de corrección de errores solo suelen utilizarse cuando el medio de transmisión es simplex, ya que en ese caso el receptor no puede solicitar retransmisión. En ocasiones los códigos de corrección de errores se denominan corrección de errores hacia adelante (forward error control) y los de detección se llaman códigos de corrección de errores hacia atrás o por realimentación (feedback o backward error control)

En esencia cualquier mecanismo de control de errores se basa pues en la inclusión de un cierto grado de redundancia, lo cual requiere un compromiso entre eficiencia y fiabilidad

El diseño de códigos correctores de errores tiene como objetivo construir códigos con redundancia razonable y rápida decodificación. Debido a sus múltiples aplicaciones en ingeniería e informática, los códigos más importantes tratados aquí son:

- ✓ Códigos de Hamming (códigos perfectos para errores simples) y sus variantes, descritos en el capítulo 2.
- ✓ Códigos de Golay (código perfecto para errores triples), descrito en el capítulo 3.
- ✓ Códigos BCH (correctores de errores múltiples) y una de sus variantes, los códigos Reed-Solomon (correctores de errores a ráfagas), descritos también en el capítulo 3.

Los errores se dan porque los medios no son perfectos, ruidos, interferencias, etc. produciéndose perdidas de información - para solucionarlo se emplean ciertas técnicas que detectan la existencia de errores y en ocasiones los corrigen.

Hay dos filosofías de corrección de errores:

Corrección hacia delante (en el destinatario)

Utilizando la información que el emisor introduce en el mensaje, el receptor localiza los bits erróneos y reconstruye el mensaje original. El inconveniente de este método es que el nº de bits necesarios para reconstruir los bits originales es muy grande y por lo tanto hay una perdida de eficacia

- ✓ códigos de hamming

Corrección hacia atrás o por retransmisión.

Esta es la mas utilizada el receptor no dispone de medios para corregir el error y pide al transmisor la retransmisión de los bloques erróneos. Su principal inconveniente es que se mantiene ocupada durante más tiempo la línea de comunicación.

Para este método es necesario que la transmisión de datos sea bidireccional. Existen dos tipos:

- ✓ **Envío y espera**: El emisor envía un bloque al receptor pero no envía el siguiente bloque hasta que recibe información de que el anterior le ha llegado bien. Es lento pero muy seguro
- ✓ **Envío continuo**: El emisor envía bloques numerados y cuando el receptor detecta un bloque erróneo da la información al emisor indicando el número de bloques que es erróneo y entonces el emisor vuelve a enviar este bloque o este bloque y los siguientes.

Distancia Hamming.

La "distancia Hamming" es el número de posiciones de bit en el que dos palabras de código difieren. Su significado consiste en que, si dos palabras clave se encuentran separadas por una distancia Hamming de valor d, se podrá transformar una palabra en otra si se tienen d errores de bit individuales. Para determinar en cuántos bits correlativos difieren, se realiza un XOR entre ambas palabras de código y se cuenta el número de bits con valor 1 obtenidos. Entiéndase una palabra de código como una unidad de $m+r$ bits que contiene m bits de datos y r bits de redundancia.

Existen 2^m mensajes de datos posibles y 2^n palabras de código utilizables, pero debido a la forma en que se calculan los bits de redundancia, no todas estas palabras son usadas. Sin embargo, se puede construir una lista completa de palabras de código legales, a partir del algoritmo de cálculo de bits de redundancia. Esto permite determinar las dos palabras de código cuya distancia sea mínima. Esta distancia es la distancia Hamming del código completo.

Por otra parte, las propiedades de detección y corrección de errores de un código, dependen de su distancia Hamming. Así, si se cuenta con un código de distancia $d+1$, se podrán detectar d errores, de otra forma no podrían cambiarse los d errores de bit por una palabra válida.

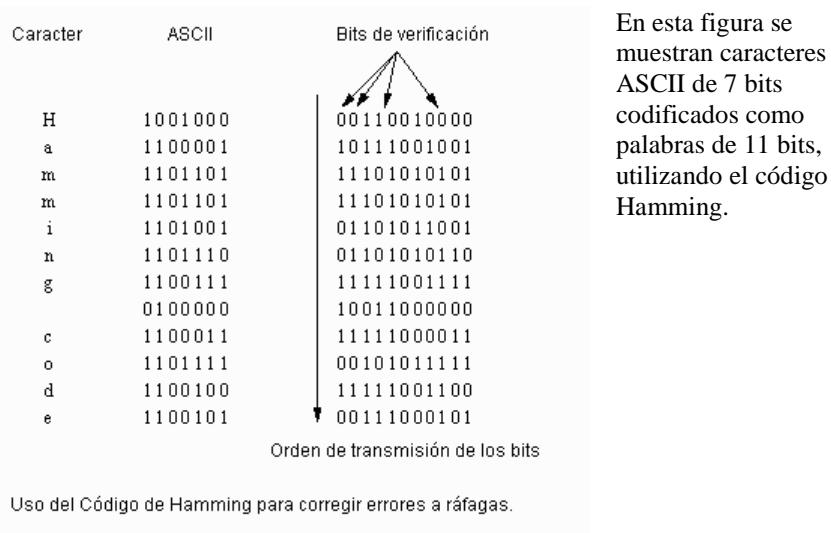
Usando este método, el receptor podrá detectar y corregir errores producidos en la transmisión. Cuando reciba una palabra de código inválida, detectará la existencia de un error. Cuando se quieran corregir d errores, se necesitará una distancia de $2d+1$, porque de ese modo, las palabras código legales estarán lo suficientemente separadas como

para que inclusive con d cambios, la palabra original esté más cercana que cualquier otra y podrá ser determinada en forma única.

Algoritmo del método desarrollado por Hamming (1950)

1. Se numeran los bits de la palabra código en forma consecutiva comenzando por el bit 1, localizado en el extremo izquierdo.
2. Los bits que son potencias de 2 (1, 2, 4, 8,...) son utilizados como bits de redundancia. El resto se llenan con los m bits de datos.
3. Cada bit de redundancia debe forzar la paridad de alguna colección de bits, incluyéndose a sí mismo para que sea par (o impar).
4. Se describe k como una suma de potencias de 2, para saber a qué bits de redundancia contribuye el bit de datos de la posición k. Por ejemplo $13 = 1 + 4 + 8$. Esta expansión es única.
5. El paso anterior se hace para comprobar el bit en base a aquellos que aparecen en su expansión. En el ejemplo anterior, 13 se comprobaría mediante el uso de los bits 1, 4 y 8.
6. El receptor inicia un contador en cero, cuando recibe una palabra código. Después, revisa cada bit de redundancia para ver si tiene la paridad correcta.
7. Si la paridad no es la esperada, el receptor suma el valor k al contador.
8. Si el contador es 0, después de haber revisado todos los bits de redundancia, la palabra se acepta como válida. Si no, el contador tiene el número del bit incorrecto.

Ilustración



Potencialidades del método

- ✓ Permite corregir errores individuales
- ✓ Puede ser modificado para corregir errores por ráfagas

II. Señales

1. Tipos y modulación

• Señales analógicas (R)

Las señales analógicas son las que varían en función del tiempo, adquiriendo valores dentro de un intervalo continuo.

Una señal analógica usa variaciones (modulaciones) en una señal, para enviar información. Es especialmente útil para datos en forma de ondas como las ondas del sonido. Las señales analógicas son las que usan normalmente su línea de teléfono y sus parlantes.

La velocidad de transmisión de una señal analógica es medida en baudios, donde 1 baudio es un cambio por segundo en la señal. Muchos usan bps y baudios como si fueran la misma cosa. Para velocidades de 2400 bps y menores, eso es verdad, pero no para las velocidades más elevadas donde por cada cambio de señal se transmite más de un bit

A la hora de transmitir una señal analógica aparecen diversos problemas:

- 1) La señal va siendo retransmitida por diversos amplificadores y otros transductores. La función de retransmisión de la señal se diseña de modo que resulte lo más lineal posible. Cualquier desviación de la linealidad que presente el sistema se traducirá en una distorsión de la señal. Toda señal analógica no es lineal en alguna medida.
- 2) El segundo problema es el ruido presente en el canal. En un canal telefónico puede oírse ese ruido, generado en este caso por las perturbaciones eléctricas de la atmósfera terrestre o por la radiación emitida por el sol y las estrellas.
- 3) Cuando se almacena una señal en algún medio de grabación, como puede ser un disco o una cinta, el propio medio constituye una fuente de ruido.
- 4) Toda señal se debilita o atenúa mediante su transmisión por un determinado medio. Esta atenuación puede debilitar tanto la señal que esta puede resultar ininteligible para el receptor. Con un cable de alta calidad y gran diámetro puede reducirse en cierta medida la atenuación, pero nunca se elimina por completo

• Señales digitales (R)

El desarrollo de la tecnología digital, en detrimento de la analógica, ha supuesto una revolución en el mundo de las telecomunicaciones. El tratamiento de la señal de este modo ha dado lugar a una serie de avances impensables unos años atrás. La comprensión y la codificación de la señal, son dos de las más importantes innovaciones de este sistema.

Se entiende por señales digitales aquellas que son discretas en el tiempo y en amplitud, lo que le da mayor eficiencia y efectividad. También llamadas señales discretas, se caracterizan porque su valor sólo puede cambiar en instantes específicos (o porque la señal sólo está definida en esos instantes).

La velocidad de la transmisión digital se mide en bits por segundo (bps). Son velocidades comunes de los módems: 28.8 Kbps, 33.6 Kbps, y 56 Kbps donde la K significa mil. Los dispositivos completamente digitales (discutidos más abajo) son mucho más rápidos. Cuanto más rápido, desde luego es mejor. Una velocidad de 2400 bps enviaría un texto de 20-páginas tipeado a un espacio, en 5 minutos.

Desde que Graham Bell descubrió el teléfono, las redes analógicas han dominado el panorama de las comunicaciones durante más de un siglo y han resultado adecuadas para las transmisiones de información (voz) a través de las redes telefónicas o de imágenes en movimiento mediante la difusión de las señales de TV. Con el tiempo este tipo de redes resultaron inapropiadas para transmitir datos cuando aparecieron los primeros ordenadores digitales, momento en el que nacieron los módems para realizar las transformaciones analógico-digital y poder utilizar las redes telefónicas existentes para conectar equipos digitales. Claro está la tecnología digital hoy en día no solo es utilizada para transmitir datos informáticos, también se adopta para la transmisión de voz y video, aprovechándose las posibilidades que ofrece y la calidad que se obtiene.

El último cuarto de siglo pasado marcó la progresiva digitalización de las redes de comunicaciones, que sucesivamente han ido sustituyendo a las redes analógicas

Comparación entre la transmisión digital y la analógica.

En la transmisión digital existen dos ventajas notables que hacen que tenga gran aceptación cuando se le compara con la transmisión analógica. En términos generales se puede decir que:

1. El ruido no se acumula en los repetidores y, por lo tanto, es una consideración secundaria en el diseño del sistema mientras que es la consideración principal en los sistemas analógicos.
2. El formato digital se adapta por sí mismo de manera ideal a la tecnología de estado sólido, particularmente a los circuitos integrados.

Con las señales digitales eliminamos el problema de la pérdida de calidad, ya que en lugar de amplificadores se emplean repetidores. Los repetidores no se limitan a aumentar la potencia de la señal, sino que decodifican los datos y los codifican de nuevo regenerando la señal en cada salto; idealmente el enlace podría tener longitud infinita.

Últimamente se utiliza mucho la transmisión digital debido a que:

- ✓ La tecnología digital se ha abaratado mucho.

- ✓ Al usar repetidores en vez de amplificadores, el ruido y otras distorsiones no es acumulativo.
- ✓ La utilización de banda ancha es más aprovechada por la tecnología digital.
- ✓ Los datos transportados se pueden encriptar y por tanto hay más seguridad en la información.
- ✓ Al tratar digitalmente todas las señales, se pueden integrar servicios de datos analógicos (voz , vídeo, etc..) con digitales como texto y otros

• **Digitalización de señales (R, E)**

Las señales digitales son discretas en el tiempo y en amplitud. Son utilizadas en los sistemas modernos de telecomunicaciones ya que son eficientes y efectivas.

Para obtener el sistema de muestreo, sólo basta observar los valores de la señal analógica o continua en el tiempo en instantes suficientemente cercanos entre sí, para poder obtener las variaciones más rápidas. A este proceso de observación H. Nyquist, lo denominó teorema del muestreo.

• **Adecuación de señales (filtros y regeneración) (R, E)**

• **Modulación en banda base (R, E)**

• **Modulación y demodulación: AM, FM, PM (R, E)**

La modulación transforma la señal digital binaria en analógica. La demodulación transforma la señal analógica en digital binaria.

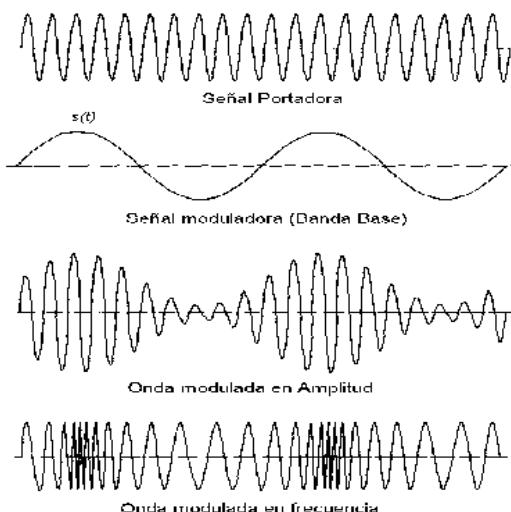
Muchas señales de entrada no pueden ser enviadas directamente hacia el canal, como vienen del transductor. Para eso se modifica una onda portadora, cuyas propiedades se adaptan mejor al medio de comunicación en cuestión, para representar el mensaje.

Definiciones:

"La modulación es la alteración sistemática de una onda portadora de acuerdo con el mensaje (señal modulada) y puede ser también una codificación"

"Las señales de banda base producidas por diferentes fuentes de información no son siempre adecuadas para la transmisión directa a través de un canal dado. Estas señales son en ocasiones fuertemente modificadas para facilitar su transmisión."

Una portadora es una senoide de alta frecuencia, y uno de sus parámetros (tal como la amplitud, la frecuencia o la fase) se varía en proporción a la señal de banda base $s(t)$. De acuerdo con esto, se obtiene la modulación en amplitud (AM), la modulación en frecuencia (FM), o la modulación en fase (PM). La siguiente figura muestra una señal de banda base $s(t)$ y las formas de onda de AM y FM correspondientes. En AM la amplitud de la portadora varía en proporción a $s(t)$, y en FM, la frecuencia de la portadora varía en proporción a $s(t)$.



Es interesante hacer hincapié en que muchas formas de comunicación no eléctricas también encierran un proceso de modulación, y la voz es un buen ejemplo. Cuando una persona habla, los movimientos de la boca ocurren de una manera mas bien lenta, del orden de los 10 Hz, que realmente no pueden producir ondas acústicas que se propaguen. La transmisión de la voz se hace por medio de la generación de tonos portadores, de alta frecuencia, en las cuerdas vocales, tonos que son modulados por los músculos y órganos de la cavidad oral. Lo que el oído capta como voz, es una onda acústica modulada, muy similar a una onda eléctrica modulada.

Tipos de modulación.

Existen básicamente dos tipos de modulación: la modulación ANALÓGICA, que se realiza a partir de señales analógicas de información, por ejemplo la voz humana, audio y video en su forma eléctrica y la modulación DIGITAL, que se lleva a cabo a partir de señales generadas por fuentes digitales, por ejemplo una computadora.

- ✓ Modulación Analógica: AM, FM, PM
- ✓ Modulación Digital: ASK, FSK, PSK, QAM

Transmisión de FM, AM, interferencia y ruido

La *señal* viene siendo la información deseada en una transmisión, y el *ruido* viene siendo como la información no-deseada. Generalmente las señales no deseadas son clasificadas como ruido. De aquí en adelante vamos a emplear el término ruido para señales no deseadas de fuentes naturales, y el término *interferencia* para señales no deseadas de fuentes *hechas por el hombre*. (Aunque existe interferencia también por fuentes naturales).

FM contra AM: Los receptores de FM tienen menor ruido que los receptores de AM. La razón es que existe mayor ruido e interferencia en la señal portadora modulada en amplitud, y los sistemas FM están diseñados para eliminar las señales no deseadas de la portadora en amplitud modulada.

Modulación en AM: Como se ve en la figura, la información de entrada varía la amplitud de la señal portadora. La frecuencia portadora es mantenida constante. Las señales transmitidas inducen un voltaje en la antena receptora, el receptor amplifica la señal y detecta las variaciones en amplitud en la señal, y reproduce la información transmitida en la salida del receptor. Note que cualquier señal de interferencia que varíe la amplitud de la portadora del receptor se convierte en una señal en la salida del receptor de AM. Es importante notar que en la transmisión de TV, las señales de video (imagen) son modulan en amplitud la portadora.

Modulación en FM: En la siguiente figura se muestra la transmisión y recepción en FM. La entrada de la información varía la frecuencia de una portadora transmitida. La frecuencia de la portadora se mantiene constante. Las señales transmitidas inducen un voltaje en la antena receptora, el receptor amplifica la señal, manda las señales a través de un limitador y discriminador, y reproduce la información transmitida en la salida del receptor. Como se muestra el limitador/discriminador corta las portadoras arriba y abajo para eliminar las variaciones en amplitud. Las señales no deseadas causan una variación en la portadora del receptor en amplitud de la antena receptora. Estas no aparecerán en la salida del receptor ya que están no varían la frecuencia de la portadora recibida. Esto es porque la transmisión en FM es esencialmente libre de interferencia y ruido, con respecto a la portadora modulada en amplitud.

Modulación en fase (PM): Variando la fase de una portadora con amplitud constante proporcional, a la amplitud de la señal modulante, con una velocidad igual a la frecuencia de la señal modulante. La PM se produce variando la fase instantánea de la portadora a una velocidad que es proporcional a la frecuencia de la modulación y por una cantidad que es proporcional a la amplitud instantánea de la modulante. La amplitud de la portadora permanece constante e inalterable.

• Modulación digital (ASK, FSK, PSK) (R)

La modulación digital es el proceso mediante el cual los símbolos digitales son transformados en forma de onda que son compatibles con la característica espectral de un canal paso banda. En el caso de la modulación paso banda, la señal de información deseada modula una senoide llamada onda portadora o simplemente portadora; para la transmisión de radio frecuencia (RF), la portadora es convertida en un campo electromagnético para su propagación al destino deseado. Se usa una portadora en la transmisión RF de señales base, porque para la transmisión de ondas electromagnéticas es necesario utilizar antenas. Para acoplar eficientemente la onda electromagnética al espacio, las dimensiones de la apertura de la antena deben ser al menos tan grandes como la longitud de la onda de la señal transmitida.

Para la transmisión de datos digitales, existen principalmente tres métodos de modulación que permiten alterar el ancho de banda sobre el cual será enviada la información. Estos tres métodos son muy empleados debido a su relativa sencillez y a que son ideales para la transmisión de datos digitales, ellos son, el ASK (Amplitude Shift Keying), FSK (Frequency Shift Keying) y PSK (Phase Shift Keying).

Los tipos básicos de modulación se listan en la tabla que a continuación se muestra. Cuando el receptor explora el conocimiento de la fase de la portadora para detectar las señales, al proceso se le conoce como detección coherente; cuando el receptor no utiliza tal información de referencia de fase, el proceso es llamado detección no coherente. En comunicaciones digitales, los términos demodulación y detección son usados en forma intercambiable, no obstante que la demodulación enfatiza en la eliminación de la portadora, y la detección involucra el proceso de decisión. En la detección coherente ideal en el receptor esta disponible el prototipo de cada señal que llega. Estas formas de onda prototipo intentan duplicar el conjunto de señales transmitidas en cualquier aspecto, inclusive en fase RF. Entonces se dice que el receptor esta “amarmando en fase” (phase locked) a la señal de la llegada. Durante la detección, el receptor multiplica e integra (correlacionada) la señal de llegada con cada una de sus réplicas prototipo.

Coherente	No coherente
Phase Shift Keying (PSK)	Diferencial Phase Shift Keying (PSK)
Frecuency Shift Keying (FSK)	Frecuency Shift Keying (FSK)
Amplitude Shift Keying (ASK)	Amplitude Shift Keying (ASK)
Continuos Phase Modulation (CPM)	Continous Phase Modulation (CPM)
Híbridos	Híbridos

Tipos de Modulación digital paso banda

La detección no coherente se refiere a los sistemas que emplean demoduladores que están diseñados para operar sin conocimiento del valor absoluto de la fase de la señal que llega; por lo tanto, no se requiere la estimación de fase. Entonces la ventaja de los sistemas no coherentes sobre los coherentes es la reducción en complejidad, y el precio pagado es el incremento en la probabilidad de error (PE).

Modulación por desplazamiento de frecuencia; La Modulación por desplazamiento de frecuencia o FSK, (Frequency Shift Keying) es la más simple de las modulaciones digitales y por lo tanto es de bajo desempeño. Es similar a la modulación de frecuencia (FM) , pero más sencillo, dado que la señal moduladora es un tren de pulsos binarios que solo varía entre dos valores de tensión discretos.

En la modulación digital, a la relación de cambio a la entrada del modulador se le llama bit-rate y tiene como unidad el bit por segundo (bps). A la relación de cambio a la salida del modulador se le llama baud-rate. En esencia el baud-rate es la velocidad de símbolos por segundo.

En FSK el bit rate = baud rate. Así por ejemplo un 0 binario se puede representar con una frecuencia f1, y el 1

binario se representa con una frecuencia distinta f_2 . El módem usa un VCO, que es un oscilador cuya frecuencia varía en función del voltaje aplicado.

Evita el problema del ruido. En este caso, al 0 y al 1 se les asignan unas frecuencias específicas y distintas. Es utilizada por módem de 300 a 4000 baudios.

Modulación en amplitud (ASK): Es el método más usado, por ejemplo, en la radio de A.M.

Las señales que maneja el ordenador son digitales, y se caracterizan por tener dos únicos valores: el uno llamado nivel alto emitiendo un impulso eléctrico de 5 voltios o el cero que es el nivel bajo sin emitirlo. El problema principal de esta modulación es que es muy sensible al ruido de la línea y un aumento en el nivel del ruido se puede interpretar fácilmente como un nivel alto cuando en realidad lo que se ha enviado es un nivel bajo.

Modulación en Fase (PSK): Cada cambio de fase es como si la porción de onda que sigue a dicho cambio, se adelantara (o atrasara) con relación a lo que debiera ser una forma senoidal continua, pura. Esta forma de cambiar la señal portadora para representar combinaciones binarias, se denomina modulación en fase (PSK=Phase-Shift-Keying=Codificación por cambio de fase). Resulta ser la más eficaz para transmitir datos binarios en líneas con ruido, siendo que requiere que el emisor y el receptor sean muy complejos.

Distintas diferencias de fase (ángulos), representan los valores de 0 y 1. Este sistema es utilizado por los módem de velocidad de 1200 bps. En este caso, la información es de dos en dos bits, asignándole a cada combinación (00, 01, 10, 11) una diferencia de fase determinada.

Además de estos modos individuales, se pueden combinar los mismos y aumentar de esta forma el número de bits transmitidos por segundo. En el caso de los módem de 2400 bps, se utiliza de forma combinada con modulación de amplitud y fase, siendo capaz de enviar grupos de 4 bits (0000, 1110, 1111, 0011, etc.) en cada estado de la señal modulada.

III. Transmisión de datos

1. Conceptos básicos

Tipos de líneas

- ✓ Líneas arrendadas/dedicadas
- ✓ Líneas comutadas

Modos de transmisión

- ✓ Modo simplex
- ✓ Modo half-duplex
- ✓ Modo full-duplex
- ✓ Técnicas de transmisión
- ✓ Transmisión asíncrona
- ✓ Transmisión síncrona

• Métodos de transmisión (serie y paralelo) (R)

Los dos tipos de transmisión que se pueden considerar son **serie** y **paralelo**. Para transmisión **serial** los bits que comprenden un carácter son transmitidos secuencialmente sobre una línea; mientras que en la transmisión en **paralelo** los bits que representan el carácter son transmitidos serialmente. Si un carácter consiste de ocho bits, entonces la transmisión en paralelo requerirá de un mínimo de ocho líneas. Aunque la transmisión en paralelo se usa extensamente en transmisiones de computadora a periféricos, no se usa aparte que en transmisiones dedicadas por el costo que implica el uso de circuitos adicionales.

La transmisión serial es más lenta que la paralela puesto que se envía un bit a la vez. Una ventaja significativa de la transmisión serial en relación a la paralela es un menor costo del cableado puesto que se necesita un solo cable se tiene un octavo del costo que se ocuparía para transmisión paralela. Este ahorro en costo se vuelve más significativo conforme sean mayores las distancias requeridas para la comunicación.

Otra ventaja importante de la transmisión serial es la habilidad de transmitir a través de líneas telefónicas convencionales a mucha distancia, mientras que la transmisión en paralelo esta limitada en distancia en un rango de metros.

- **Conecciones punto a punto y multipunto. *multicasting* y *broadcasting* (R)**

La distribución geográfica de dispositivos terminales y la distancia entre cada dispositivo y el dispositivo al que se transmite son parámetros importantes que deben ser considerados cuando se desarrolla la configuración de una red. Los dos tipos de conexiones utilizados en redes son punto a punto y multipunto.

Las líneas de conexión que solo conectan dos puntos son **punto a punto**. Cuando dos o más localidades terminales comparten porciones de una línea común, la línea es **multipunto**. Aunque no es posible que dos dispositivos en una de estas líneas transmita al mismo tiempo, dos o más dispositivos pueden recibir un mensaje al mismo tiempo. En algunos sistemas una dirección de difusión (broadcast) permite a todos los dispositivos conectados a la misma línea multipunto recibir un mensaje al mismo tiempo. Cuando se emplean líneas multipunto, se pueden reducir los costos globales puesto que porciones comunes de la línea son compartidos para uso de todos los dispositivos conectados a la línea. Para prevenir que los datos transmitidos de un dispositivo interfieran con los datos transmitidos por otro, se debe establecer una disciplina o control sobre el enlace.

Cuando se diseña una red local de datos se pueden mezclar tanto líneas punto a punto como multipunto, y la transmisión se puede efectuar en modo simplex, half-duplex o full-duplex.

Multicast

Multicast significa comunicación entre un solo emisor y múltiples receptores dentro de una red. Los usos típicos incluyen la actualización de personal móvil desde una oficina central y el manejo periódico del correo electrónico.

Multicast es soportado por redes de datos inalámbricas como parte de la tecnología de paquetes de datos Digitales Celulares (CDPD).

Multicast también es usado para la programación sobre Mbone, un sistema que permite a los usuarios con acceso a grandes anchos de banda sobre Internet el recibir en vivo imágenes de video y programas de sonido. Junto con el hecho de la existencia de una subred de Internet con un gran ancho de banda, Mbone multicast también utiliza un protocolo que permite encapsular señales como paquetes TCP/IP cuando pasan a través de lugares en la Internet que no pueden manejar el protocolo de multicast directamente

Broadcast

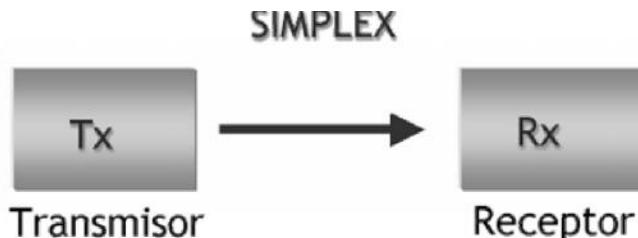
Tipo de comunicación basada en la difusión en que todo posible receptor es alcanzado por una sola transmisión.

- **Conecciones *half duplex* y *full duplex* (R)**

Un método de caracterizar líneas, dispositivos terminales, computadoras y módems es por su modo de transmisión o de comunicación. Las tres clases de modos de transmisión son simplex, half-duplex y full-duplex.

Transmisión simplex

La transmisión simplex (sx) o *unidireccional* es aquella que ocurre en una dirección solamente, deshabilitando al receptor de responder al transmisor. Normalmente la transmisión simplex no se utiliza donde se requiere interacción humano-máquina. Ejemplos de transmisión simplex son: La radiodifusión (broadcast) de TV y radio, el paging unidireccional, etc.



Transmisión half-duplex

La transmisión half-duplex (hdx) permite transmitir en ambas direcciones; sin embargo, la transmisión puede ocurrir solamente en una dirección a la vez. Tanto transmisor y receptor comparten una sola frecuencia. Un

ejemplo típico de half-duplex es el radio de banda civil (CB) donde el operador puede transmitir o recibir, pero no puede realizar ambas funciones simultáneamente por el mismo canal. Cuando el operador ha completado la transmisión, la otra parte debe ser avisada que puede empezar a transmitir (ej. diciendo "cambio").

HALF DUPLEX



Transmisión full-duplex

La transmisión full-duplex (fdx) permite transmitir en ambas dirección, pero simultáneamente por el mismo canal. Existen dos frecuencias una para transmitir y otra para recibir. Ejemplos de este tipo abundan en el terreno de las telecomunicaciones, el caso más típico es la telefonía, donde el transmisor y el receptor se comunican simultáneamente utilizando el mismo canal, pero usando dos frecuencias.

• Comutación de circuitos (R)

Cuando se hace una llamada telefónica, el equipo de comutación del sistema telefónico busca una trayectoria física de "cobre" (lo que incluye la fibra y la radio) que vaya desde su teléfono al del receptor. Esta técnica es llamada **comutación de circuitos**.

En los primeros días del teléfono, se establecía la conexión cuando el operador conectaba un cable punteador en los enchufes de entrada y de salida. Sin embargo, la idea básica es válida: una vez que se ha establecido una llamada, existe una trayectoria dedicada entre ambos extremos y continuará existiendo hasta que termine la llamada.

Una propiedad importante de la **comutación de circuitos** es la necesidad de establecer una trayectoria de un extremo a otro antes de que se pueda enviar cualquier dato. El tiempo que transcurre entre que se termina de marcar y que el timbre comienza a sonar puede ser fácilmente de 10 seg., y más en las llamadas de larga distancia o internacionales. Durante este intervalo de tiempo, el sistema telefónico está buscando una trayectoria de cobre. Observe que antes de que pueda comenzar la transmisión de datos, la señal de petición de llamada se debe propagar hasta el destino, y debe ser reconocida. En muchas aplicaciones de computadora (por ejemplo, la verificación de crédito en un punto de venta), son indeseables los tiempos de establecimiento largos.

Al existir una trayectoria de cobre entre las partes en comunicación, una vez completado el establecimiento el único retardo de los datos es el tiempo de propagación de la señal electromagnética, alrededor de 5ms por cada 1000 km. Otra ventaja de la trayectoria establecida es que no hay peligro de congestión; esto es, una vez que la llamada entró, no hay posibilidad de obtener una señal de ocupado, aunque podría obtener una antes de establecerse la conexión debido a la falta de capacidad de comutación o de troncal.

• Comutación de paquetes (R)

Una estrategia de comutación alterna es la **comutación de mensajes**. Cuando se usa esa forma de comutación, no se establece por adelantado una trayectoria de cobre física entre el emisor y el receptor. En cambio, cuando el emisor tiene un marco de datos para enviar, éste se almacena en la primera oficina de comutación (es decir, enrutador) y después se reenvía, un salto a la vez. Cada marco se recibe en su totalidad, se inspecciona en busca de errores, y después se retransmite. Una red que usa esa técnica se llama red de almacenar y reenviar.

Los primeros sistemas de telecomunicaciones electromecánicos usaban comutación de circuitos para enviar telegramas. El mensaje se perforaba en cinta de papel fuera de línea en la oficina emisora y después se leía y transmitía por una línea de comunicación a la siguiente oficina a lo largo del recorrido, donde se perforaba en cinta de papel. Allí una operadora arrancaba la cinta de papel y la insertaba en una de las muchas lectoras de cinta, una por cada troncal de salida. Tal oficina de comutación se llamaba **oficina de arrancado de cinta de papel**.

Con la comutación de mensajes, no hay límite para el tamaño de los marcos, lo que significa que los enruteadores (en un sistema moderno) deben tener discos para almacenar en forma temporal los marcos largos. También significa que un solo marco puede acaparar una línea de enruteador a enruteador durante minutos, lo que hace inútil

la conmutación de mensajes para el tráfico interactivo. Con el fin de resolver estos problemas se inventó la **comutación de paquetes**. Las redes de conmutación de paquetes establecen un límite superior al tamaño del marco, lo que permite almacenar los paquetes en la memoria principal del enrutador en lugar de hacerlo en disco. Al asegurarse de que ningún usuario pueda monopolizar una línea de transmisión durante mucho tiempo (milisegundos), las redes de conmutación e paquetes pueden manejar tráfico interactivo. Una ventaja adicional de la conmutación de paquetes sobre la conmutación de mensajes: el primer paquete de un mensaje de varios paquetes se puede reenviar antes de que el segundo haya llegado por completo, lo que reduce el retardo y mejora el rendimiento. Por estas razones, las redes de computadoras normalmente son de conmutación de paquetes, ocasionalmente de conmutación de circuitos y nunca de conmutación de mensajes.

La diferencia clave es que la conmutación de circuitos reserva de manera estática por adelantado el ancho de banda requerido, mientras que la conmutación de paquetes lo adquiere y lo libera según se necesita. Con la conmutación de circuitos, cualquier ancho de banda que no se use en un circuito asignado sencillamente se desperdicia. Con la conmutación de paquetes este ancho de banda se puede utilizar para transmitir otros paquetes de fuentes no relacionadas que van a destinos no relacionados porque los circuitos nunca son dedicados. Por otro lado, debido a que no hay circuitos dedicados, una crecida súbita en el tráfico de entrada puede saturar un enrutador, excediendo su capacidad de almacenamiento y provocando que pierda paquetes.

En contraste con la conmutación de circuitos, cuando se usa la conmutación de paquetes resulta sencillo para los enrutadores efectuar conversiones de velocidad y de código. También pueden realizar corrección de errores en cierto grado. Sin embargo, en algunas redes de conmutación de paquetes éstos se pueden entregar a su destino en el orden equivocado, cosa que nunca puede suceder con la conmutación de circuitos.

Otra diferencia es que la conmutación de circuitos es totalmente transparente. El emisor y el receptor pueden usar cualquier velocidad, formato o método de encuadrado de bits que quieran. La portadora no lo sabe ni le interesa. Con la conmutación de paquetes la portadora determina los parámetros básicos. Una analogía burda sería comparar un camino con una vía de tren. En el primero, el usuario determina el tamaño, la velocidad y la naturaleza del vehículo; en la vía del tren esto lo hace el prestador de servicios. Esta transparencia es la que hace posible que coexistan voz, datos y fax dentro del sistema telefónico.

Una diferencia final entre la conmutación de circuitos y la de paquetes es el algoritmo de cobro. Las portadoras de paquetes por lo general basan sus cargos tanto en la cantidad de bytes (o paquetes) transmitidos como en el tiempo de conexión. Además, la distancia de transmisión normalmente no importa, excepto quizás en lo internacional. Con la conmutación de circuitos, el cargo solo se basa en la distancia y el tiempo, no en el tráfico. Las diferencias se resumen en la siguiente tabla:

Elemento	Comutación de Circuitos	Comutación de paquetes
Trayectoria de “cobre” dedicada	Sí	No
Ancho de Banda Disponible	Fijo	Dinámico
Puede Desperdiciarse Ancho de Banda	Sí	No
Transmisión de Almacenamiento y Reenvío	No	Sí
Cada Paquete Sigue la misma Ruta	Sí	No
Establecimiento de Llamada	Requerido	No es Necesario
Cuando puede haber Congestión	Durante el establecimiento	En cada Paquete
Cargos	Por Minuto	Por Paquete

Comparación de las Redes de Comutación de Circuitos y Comutación de Paquetes

• Transmisión sincrónica (síncrona) y asincrónica (asíncrona) (R)

Transmisión asincrónica

La transmisión asincrónica es aquella que se transmite o se recibe un carácter, bit por bit añadiéndole *bits de inicio*, y bits que indican el término de un paquete de datos, para separar así los paquetes que se van enviando/recibiendo para sincronizar el receptor con el transmisor. El bit de inicio le indica al dispositivo receptor que sigue un carácter de datos; similarmente el bit de término indica que el carácter o paquete ha sido completado.

Transmisión Síncrona

Este tipo de transmisión el envío de un grupo de caracteres en un flujo continuo de bits. Para lograr la sincronización de ambos dispositivos (receptor y transmisor) ambos dispositivos proveen una señal de reloj que se usa para establecer la velocidad de transmisión de datos y para habilitar los dispositivos conectados a los módems para identificar los caracteres apropiados mientras estos son transmitidos o recibidos. Antes de iniciar la comunicación ambos dispositivos deben de establecer una sincronización entre ellos. Para esto, antes de enviar los datos se envían un grupo de caracteres especiales de sincronía. Una vez que se logra la sincronía, se pueden empezar a transmitir datos.

Por lo general los dispositivos que transmiten en forma síncrona son más caros que los asíncronos. Debido a que son más sofisticados en el hardware. A nivel mundial son más empleados los dispositivos asíncronos ya que facilitan mejor la comunicación.

• Formatos de mensajes (R)

• Multiplexaje por división de frecuencias (R, E)

Multiplexión

La economía de escala desempeña un papel importante en el sistema telefónico. Cuesta prácticamente lo mismo instalar y mantener un troncal de alto ancho de banda que uno de bajo ancho de banda entre 2 oficinas de conmutación (esto es, el gasto principal es la excavación de zanjas y no el alambre de cobre o la fibra óptica). En consecuencia, las compañías de teléfonos han desarrollado esquemas elaborados para multiplexar muchas conversaciones en un solo troncal físico. Estos esquemas de multiplexión se pueden dividir en 2 categorías: **FDM** (frequency division multiplexing, **multiplexión por división en frecuencia**) y **TDM** (time division multiplexing, **multiplexión por división en tiempo**). En FDM el espectro de frecuencia se divide entre los canales lógicos, con cada usuario en posesión exclusiva de alguna banda de frecuencia. En TDM los usuarios esperan su turno (por asignación cíclica), obteniendo cada uno en forma periódica la banda entera durante un breve lapso de tiempo.

La radiodifusión AM ilustra ambas clases de multiplexión. El espectro asignado es de alrededor de 1MHz, aproximadamente de 500 a 1500 kHz. A los diferentes canales lógicos (estaciones) se les asigna una frecuencia distinta, y cada uno funciona en una porción del espectro con una separación entre canales lo bastante grande como para prevenir la interferencia. Este sistema es un ejemplo de multiplexión por división en frecuencia. Adicionalmente, las estaciones individuales tienen 2 subcanales lógicos, música y publicidad, los cuales se alternan en el tiempo en la misma frecuencia, primero una ráfaga de música y después una ráfaga de publicidad, luego más música y así sucesivamente. Esta situación es multiplexión por división en tiempo.

Multiplexión por división de frecuencia

La multiplexión por división de frecuencia es el esquema de reparto de canal más viejo y probablemente el más utilizado aún. Un transpondedor típico de 36 Mbps puede dividirse estáticamente en unos 500 canales PCM de 64,000 bps, operando cada uno en su propia frecuencia para evitar la interferencia con los demás.

Aunque es simple, la FDM también tiene algunos inconvenientes. Primero, se requieren bandas de seguridad (laterales) entre los canales para mantener separadas las estaciones. Este requisito existe porque no es posible construir transmisores que envíen toda su energía por la banda principal y nada por las bandas laterales. La cantidad de ancho de banda desperdiciado en las bandas laterales puede ser una parte sustancial del total.

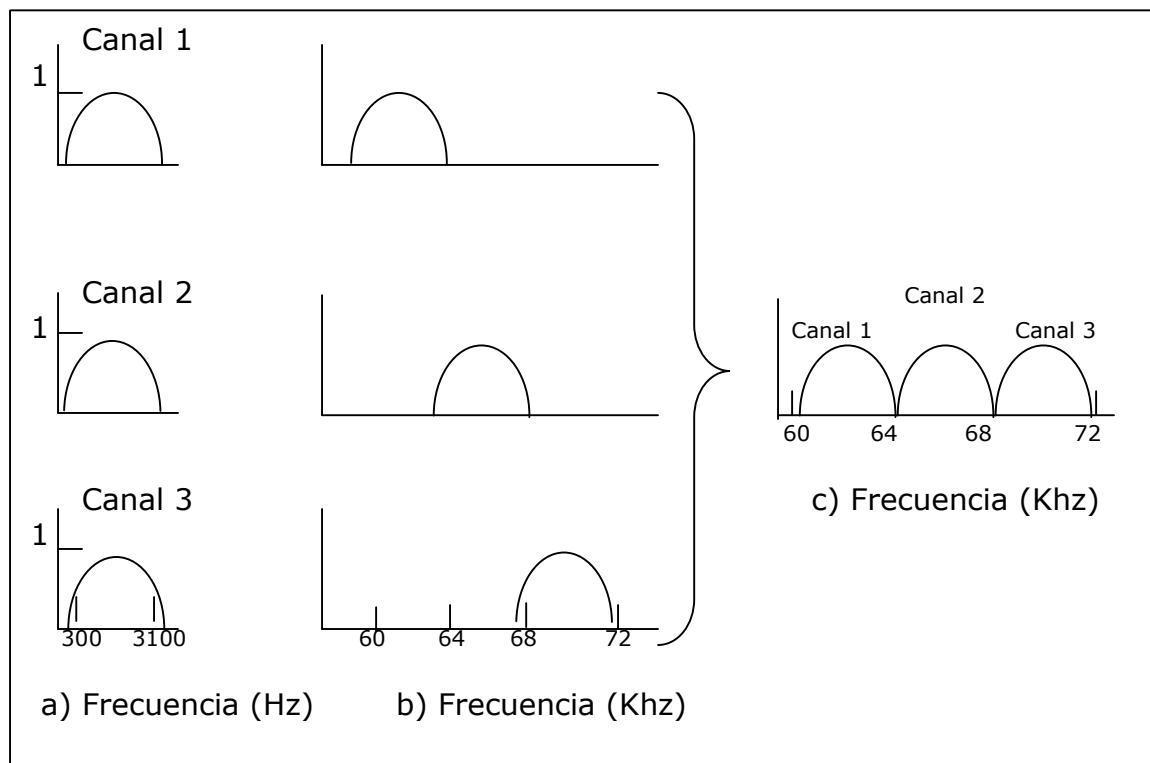
Segundo, debe controlarse cuidadosamente la potencia de las estaciones. Si una estación pone demasiada potencia en la banda principal, también pondrá automáticamente demasiada potencia en las bandas laterales, ocurriendo un derramamiento a los canales adyacentes que causa interferencia. Por último la FDM es una técnica completamente analógica que no se presta bien a una implementación en software.

Si el número de estaciones es pequeño y fijo, pueden repartirse por adelantado los canales de frecuencia. Sin embargo, si el número de estaciones, o la carga de cada una de ellas, puede fluctuar con rapidez, se necesita algún tipo de reparto dinámico de las bandas de frecuencias. Uno de tales mecanismos es el sistema SPADE empleado en algunos de los primeros satélites Intelsat. Cada repetidor SPADE se dividió en 794 canales de voz PCM simples (de 64 kbps). Junto con un canal de señalización común de 128 kbps. Los canales PCM se usaban en pares para proporcionar servicio dúplex integral. El ancho de banda de transpondedor total usado fue de 50 Mbps para la parte de enlace ascendente y otros 50 Mbps para el enlace descendente.

El canal de señalización común se dividió en unidades de 50mseg. Una unidad contenía 50 ranuras de 1 mseg (128 bits). Cada ranura “pertenece” a una de 50 estaciones de tierra (no más). Cuando una estación de tierra tenía datos por enviar, tomaba al azar un canal desocupado en ese momento y escribía el número del canal en su siguiente ranura de 128 bits. Si el canal seleccionado aún estaba desocupado cuando se veía la solicitud en el enlace descendente, el canal se consideraba como repartido y las demás estaciones se abstendían de intentar adquirirlo. Si dos o más estaciones trataban de reservar el mismo canal durante el mismo marco, ocurría una colisión y tenían que intentarlo de nuevo posteriormente. Al terminar de usar su canal, la estación enviaba un mensaje de liberación en su ranura de 1 canal común.

La siguiente figura muestra cómo multiplexar 3 canales telefónicos de grado de voz utilizando FDM. Los filtros limitan el ancho de banda utilizable a cerca de 3000 Hz por canal de grado de voz. Cuando se multiplexan muchos canales juntos, se asigna 4000 Hz a cada canal para mantenerlos bien separados. Primero se eleva la frecuencia de los canales de voz cada uno en una cantidad diferente, después de lo cual ya se pueden combinar, porque ahora no hay 2 canales que ocupen la misma porción del espectro. Observe que aunque existen separaciones entre los canales (bandas de protección), hay cierta superposición entre canales adyacentes porque los filtros no tienen bordes abruptos. Esta superposición significa que un pico fuerte en el borde de un canal se detectara en el adyacente como ruido no térmico.

Los esquemas de FDM que se emplean en el mundo están normalizados hasta cierto punto. Un estándar muy difundido es el de 12 canales de voz a 4000 Hz (3000 Hz para el usuario más dos bandas de protección de 500 Hz cada una) multiplexados dentro de la banda de 60 a 108 kHz. Esta unidad se llama grupo. La banda de 12 a 60 kHz a veces se usa para otro grupo. Muchas portadoras ofrecen un servicio de líneas alquiladas de 48 a 56 kbps que se basan en este grupo. Se pueden multiplexar 5 grupos (60 canales de voz) para formar un super grupo. La siguiente unidad es el grupo maestro, que se compone de 5 super grupos (en el estándar de CCITT) o de 10 super grupos (en el sistema Bell). Existen también otros estándares que llegan hasta 230,000 canales de voz.



Multiplexión por división en Frecuencia

- A) Los anchos de banda originales
- B) Los anchos de Banda elevados en Frecuencia
- C) El canal Multiplexado

- **Multiplexaje por división de tiempo (R, E)**

Al igual que la FDM, la TDM es bien entendida y se usa ampliamente en la práctica; requiere sincronización de tiempo para las ranuras, pero esto puede proporcionarse mediante una estación de referencia. De modo parecido a la FDM, para un número pequeño e invariable de estaciones, la asignación de ranuras puede establecerse por adelantado y no cambiarse nunca, pero con un número variable de estaciones, o un número fijo de estaciones con cargas variables en el tiempo, las ranuras de tiempo deben asignarse dinámicamente.

La asignación de ranuras puede hacerse de manera centralizada o descentralizada. Como ejemplo de asignación de ranuras centralizada, consideremos el ACTS (Advanced Communication Technology Satellite, satélite de tecnología de comunicaciones avanzada), que se diseñó para unas pocas docenas de estaciones. El ACTS se lanzó en 1992 y tiene 4 canales TDM independientes de 110 Mbps, dos de enlace ascendente y 2 de enlace descendente. Cada canal está organizado como secuencia de marcos de 1 mseg, conteniendo cada marco 1728 ranuras de tiempo. Cada ranura tiene una carga útil de 64 bits, permitiendo a cada una contener un canal de voz de 64 kbps.

Aunque todavía se usa la FDM en cables de cobre o canales de microondas, requiere circuitos analógicos y no es fácil que lo efectúe una computadora. En contraste, la TDM se puede controlar por completo con electrónica digital, de modo que se ha extendido en forma más amplia en años recientes. Desafortunadamente, esta técnica sólo se puede aplicar a datos digitales. Puesto que los lazos locales producen señales analógicas, se requiere una conversión de analógico a digital en la oficina final, donde se reúnen todos los lazos locales individuales para combinarse en troncales de salida.

Las señales analógicas se digitalizan en la oficina final con un dispositivo llamado codec (codificador decodificador), con lo que producen números de 7 u 8 bits. El codec toma 800 muestras por segundo (125 µseg/muestra) porque el teorema de Nyquist dice que esto es suficiente para capturar toda la información del ancho de banda de 4 kHz del canal telefónico. A una velocidad de muestreo menor, la información se perdería; a una mayor, no se ganaría información extra. Esta técnica se llama PCM (pulse code modulation, modulación de código de pulso).

2. Medios físicos de transmisión de datos

Medios de Transmisión

El propósito de la capa física es transportar una corriente de bits en bruto de una máquina a otra. Se puede usar varios medios físicos para la transmisión real; cada uno con su propio nicho en términos de ancho de banda, retardo, costo y facilidad de instalación y mantenimiento. A grandes rasgos, los medios se agrupan en medios guiados, como el cable de cobre y la fibra óptica, y medios no guiados, como la radio y los láseres a través del aire.

Medios Magnéticos

Una de las formas más comunes de transportar datos de una computadora a otra es escribirlos en cinta magnética o disquetes, transportar físicamente la cinta o los discos a la máquina de destino y leerlos de nuevo. Si bien este método no es tan avanzado como usar un satélite de comunicación geosíncrono, con frecuencia tiene mucha mayor eficacia de costo, en especial con aplicaciones en las que el ancho de banda o costo por bit transportado es un factor clave.

Respaldar diariamente los gigabytes de datos de un banco en una segunda máquina (de manera que el banco pueda continuar funcionando aun en caso de un desastre mayor o de un terremoto) es probable que ninguna otra tecnología de transmisión pueda siquiera acercarse a la cinta magnética en rendimiento.

- **Alámbricos (coaxial, par trenzado, fibra óptica) (R, E, RP)**

Par Trenzado

Aunque las características de ancho de banda de la cinta magnética son excelentes, las características de retardo son pobres. El tiempo de transmisión se mide en minutos o en horas y no en milisegundos. Para muchas aplicaciones se necesita conexión en línea. El medio de transmisión más viejo y todavía el más común es el par trenzado. Un par trenzado consiste en 2 alambres de cobre aislados, por lo regular de 1mm de grueso. Los alambres se trenzan en forma helicoidal, igual que una molécula de DNA. El propósito de torcer los alambres es reducir la interferencia eléctrica de pares similares cercanos. (Dos alambres paralelos constituyen una antena simple; un par trenzado no)

La aplicación más común del par trenzado es en el sistema de teléfonos. Casi todos los teléfonos se conectan a la central telefónica por un par trenzado. Se pueden tender varios kilómetros de par trenzado sin necesidad de amplificación, pero se necesitan repetidoras para distancias mayores. Cuando muchos ares entrelazados corren distancias sustanciales en paralelo, como los cables que van de un edificio de departamentos a la central telefónica, se atan en un haz y se forran con una funda que los protege. Los pares de estos haces interferirían unos con otros si no fuera por el entrelazamiento. En algunas partes del mundo donde las líneas telefónicas penden de postes sobre el terreno es común ver haces de varios centímetros de diámetro.

Los pares trenzados se pueden usar tanto para transmisión analógica como digital. El ancho de banda depende del grosor del cable y de la distancia, pero en muchos casos se pueden lograr varios megabits/seg durante algunos kilómetros. Los pares entrelazados se usan ampliamente debido a su rendimiento adecuado y a su bajo costo.

El cableado de par trenzado tiene algunas variaciones, dos de las cuales son importantes para las redes de computadoras. Los pares entrelazados de la categoría 3 consisten en 2 hilos aislados que se trenzan de manera delicada. Cuatro de estos pares se agrupan por lo regular en una funda de plástico para su protección y para mantener juntos los 8 hilos.

Alrededor de 1988 se comenzaron a introducir los pares trenzados más avanzados de la categoría 5; son similares a los de la categoría 3, pero con mas vueltas por centímetro y con aislamiento de Teflón, lo cual produce menor diafonía y una señal de mejor calidad a distancias más largas, lo que los hace más adecuados para la comunicación de computadoras a alta velocidad. Ambos tipos de cableado con frecuencia reciben el nombre de UTP (unshielded twisted pair, par trenzado sin blindaje).

Cable coaxial de banda base

Otro medio de transmisión es el cable coaxial. Este cable tiene mejor blindaje que el par trenzado, así que puede abarcar tramos más largos a velocidades mayores. Son 2 las clases de cable coaxial más utilizadas. Una clase, el cable de 50 ohms, se usa comúnmente para transmisión digital. La otra clase, el cable de 75 ohms, se usa comúnmente para la transmisión analógica.

Un cable coaxial consiste en un alambre de cobre rígido como núcleo, rodeado por un material aislante. El aislante está forrado con un conductor cilíndrico, que con frecuencia es una malla de tejido fuertemente trenzado. El conductor externo se cubre con una envoltura protectora de plástico.

La construcción y el blindaje del cable coaxial le confieren una buena combinación de elevado ancho de banda y excelente inmunidad al ruido. El ancho de banda posible depende de la longitud del cable. En cables de 1 Km es factible una velocidad de datos de 1 a 2 Gbps. También se pueden usar cables más largos, pero a velocidades de datos más bajas o con amplificadores periódicos. Los cables coaxiales solían usarse en el sistema telefónico, pero ahora se les ha reemplazado en gran medida por fibra óptica en rutas de largo recorrido. Sin embargo el cable coaxial todavía se utiliza para la televisión por cable y algunas redes de área local.

Cable coaxial de banda ancha

El otro tipo de sistema de cable coaxial transporta transmisión analógica en el cableado estándar de la televisión por cable; se le llama de banda ancha. En el mundo de las redes de computadoras “cable de banda ancha” significa cualquier red de cable que utilice transmisión analógica.

Puesto que las redes de banda ancha emplean tecnología estándar de la televisión por cable, los cables se pueden usar hasta 300 MHz (y con frecuencia hasta 450 MHz) y pueden extenderse distancias de cerca de 10 km gracias a la señalización analógica, que es mucho menos crítica que la digital. Para transmitir señales digitales por una red analógica, cada interfaz debe contener circuitos electrónicos para convertir la corriente de bits saliente en una señal analógica y la señal analógica entrante en una corriente de bits. Dependiendo del tipo de estos circuitos, 1 bps puede ocupar aproximadamente 1 Hz de ancho de banda. A frecuencias más altas pueden transmitirse muchos bits por Hz si se emplean técnicas de modulación avanzadas.

Los sistemas de banda ancha se dividen en múltiples canales, con frecuencia los canales de 6 Mhz que se usan para la difusión de televisión. Cada canal puede servir para televisión analógica, audio de calidad de disco compacto, o una corriente digital de bits independiente de las otras, digamos a 3 Mbps. La televisión y los datos se pueden mezclar en un cable.

Una diferencia clave entre la banda base y la banda ancha es que los sistemas de banda ancha normalmente cubren un área mayor y, por tanto, necesitan amplificadores analógicos para reforzar la señal en forma periódica. Estos amplificadores solamente pueden transmitir señales en una dirección, de modo que una computadora que esté enviando un paquete no será capaz de comunicarse con una computadora situada “corriente arriba” de ella si entre ellas hay un amplificador. Para superar este problema se han desarrollado dos tipos de sistemas de banda ancha: sistemas de cable dual y de cable sencillo.

Fibra Óptica

Los cables de fibra óptica son similares a los coaxiales, excepto por el trenzado. El núcleo de vidrio está al centro, y a través de él se propaga la luz. En las fibras multimodales el diámetro es de 50 micras, aproximadamente el grosor de un cabello humano. En las fibras de modo único el núcleo es de 8 a 10 micras.

El núcleo esta rodeado por un revestimiento de vidrio con un índice de refracción menor que el del núcleo, a fin de mantener toda la luz en el núcleo. A continuación viene una cubierta plástica delgada para proteger al revestimiento. Las fibras normalmente se agrupan en haces, protegidas por una funda exterior.

Los haces de fibras terrestres por lo regular se colocan un metro debajo de la superficie, donde ocasionalmente están sujetos a ataques de azadones o picos. Cerca de la costa, los haces de fibras transoceánicas se entierran en zanjas con una especie de arado marino.

Las fibras se pueden conectar de 3 formas diferentes. Primera, pueden terminar en conectores e insertarse en enchufes de fibra. Los conectores pierden casi el 10 o 20% de la luz, pero facilitan la reconfiguración de los sistemas. Segunda, se pueden empalmar de manera mecánica. Los empalmes mecánicos acomodan 2 extremos cortados con cuidado uno junto a otro en una manga especial y lo sujetan en su lugar. Se puede mejorar la alineación haciendo pasar luz por la unión y efectuando pequeños ajustes para maximizar la señal. Los empalmes mecánicos toman al personal entrenado cerca de 5 minutos y resultan en una perdida de luz del 10%.

Tercera, se pueden fusionar dos tramos de fibra PATRA formar una conexión sólida. Un empalme por fusión es casi tan bueno como una fibra de hilado único, pero aun aquí hay un poco de atenuación. Con los 3 tipos de empalme pueden ocurrir reflejos en el punto del empalme, y la energía reflejada puede interferir en la señal.

Las fibras ópticas se pueden utilizar con LAN, así como para transmisión de largo alcance, aunque derivar en ella es más complicado que conectarse a una ethernet.

• Inalámbricos (microondas, infrarrojos, satelital) (R, E, RP)

Nuestra era ha dado origen a adictos a la información; gente que necesita estar en línea todo el tiempo. Para estos usuarios móviles, los pares trenzados, los cables coaxiales y la fibra óptica no son útiles; necesitan obtener los datos para sus computadoras laptop, notebook, de bolsillo, de mano o de reloj de pulsera sin estar atados a la infraestructura de comunicaciones terrestres. La comunicación inalámbrica también tiene ventajas para los dispositivos fijos en ciertas circunstancias. Por ejemplo, si es difícil tender fibras hasta un edificio debido al terreno (montañas, selvas, pantanos, etc.), podría ser preferible un sistema inalámbrico. Es interesante que la comunicación digital inalámbrica moderna se inició en las islas de Hawái, donde largos tramos del Océano Pacífico separaban a los usuarios y el sistema telefónico era inadecuado.

El espectro magnético

Cuando los electrones se mueven crean ondas electromagnéticas que se pueden propagar por el espacio libre (aun en el vacío). La cantidad de oscilaciones por segundo de una onda electromagnética es su frecuencia, f , y se mide en Hz (en honor a Heinrich Hertz). La distancia entre 2 máximos (o mínimos) consecutivos se llama longitud de onda y se designa de forma universal con la letra griega (λ). Al conectar una antena del tamaño apropiado a un circuito eléctrico, las ondas electromagnéticas se pueden difundir de manera eficiente y captarse por un receptor a cierta distancia. Toda la comunicación inalámbrica se basa en este principio.

En el vacío, todas las ondas electromagnéticas viajan a la misma velocidad, no importa cuál sea su frecuencia. Esta velocidad, usualmente llamada velocidad de la luz, c , es de aproximadamente 3×10^8 m/seg, o de casi 1 pie (30 cm) por nanosegundo. En el cobre o la fibra, la velocidad baja a casi 2/3 de este valor y se vuelve ligeramente dependiente de la frecuencia. La velocidad de la luz es el límite máximo de velocidad. Ningún objeto o señal puede llegar a ser más rápido que la luz.

Radiotransmisión

Las ondas de radio son fáciles de generar, pueden viajar distancias largas y penetrar edificios sin problemas, de modo que se utilizan mucho en la comunicación, tanto en interiores como en exteriores. Las ondas de radio también son omnidireccionales, lo que significa que viajan en todas direcciones desde la fuente, por lo que el transmisor y el receptor no tienen que alinearse con cuidado físicamente.

Las propiedades de las ondas de radio dependen de la frecuencia. A bajas frecuencias, las ondas de radio cruzan bien los obstáculos, pero la potencia se reduce drásticamente con la distancia a la fuente, aproximadamente en proporción $1/r^3$ en el aire. A frecuencias altas, las ondas de radio tienden a viajar en línea recta y a rebotar en los obstáculos. También son absorbidas por la lluvia. En todas las frecuencias, las ondas de radio están sujetas a interferencia por los motores y otros equipos eléctricos.

Por la capacidad del radio de viajar distancias largas, la interferencia entre usuarios es un problema. Por esta razón, los gobiernos legislan estrictamente el uso de radiotransmisores.

En las bandas VLF, LF y MF, las ondas de radio siguen el terreno. Estas ondas se pueden detectar quizás a 1000 km en las frecuencias más bajas, y a menos en frecuencias más altas. La difusión de radio AM usa la banda MF, y es por ello que las estaciones de radio AM de Boston no se pueden oír con facilidad en Nueva York. El problema principal al usar estas bandas para comunicación de datos es el ancho de banda relativamente bajo que ofrecen. En las bandas HF y VHF, las ondas a nivel del suelo tienden a ser absorbidas por la tierra. Sin embargo, las ondas que alcanzan la ionosfera, se refractan y se envían de regreso a nuestro planeta. En ciertas condiciones atmosféricas, las señales pueden rebotar varias veces. El ejército se comunica también en las bandas HF y VHF.

Transmisión por Microondas

Por encima de los 100 MHz las ondas viajan en línea recta y, por tanto, se pueden enfocar en un haz estrecho. Concentrar toda la energía en un haz pequeño con una antena parabólica produce una señal mucho más alta en relación con el ruido, pero las antenas transmisora y receptora deben estar muy bien alineadas entre sí. Además esta direcciónabilidad permite a transmisores múltiples alineados en una fila comunicarse con receptores múltiples en fila, sin interferencia. Antes de la fibra óptica, estas microondas formaron durante décadas la base del sistema de transmisión telefónica de larga distancia.

Ya que las microondas viajan en línea recta, si las torres están muy separadas, partes de la Tierra estorbarán. En consecuencia, se necesitan repetidoras periódicas. Cuanto más altas sean las torres, más separadas pueden estar. La distancia entre las repetidoras se eleva en forma aproximada con la raíz cuadrada de la altura de las torres. Con torres de 100 m de altura, las repetidoras pueden estar espaciadas a 80 km de distancia.

A diferencia de las ondas de radio a frecuencias más bajas, las microondas no atraviesan bien los edificios. Además, aun cuando el haz puede estar bien enfocado el transmisor, hay cierta divergencia en el espacio. Algunas ondas pueden refractarse en las capas atmosféricas más bajas y tardar un poco más en llegar que las ondas directas. Las ondas diferidas pueden llegar fuera de fase con la onda directa y cancelar así la señal. Este efecto se llama desvanecimiento de trayectoria múltiple y con frecuencia es un problema serio que depende del clima y de la frecuencia. Algunos operadores mantienen el 10% de sus canales inactivos como repuesto para activarlos cuando el desvanecimiento de trayectoria múltiple cancela en forma temporal alguna banda de frecuencia.

La comunicación por microondas se utiliza tanto para la comunicación telefónica de larga distancia, los teléfonos celulares, la distribución de la televisión y otros usos, que el espectro se ha vuelto muy escaso. Esta tecnología tiene varias ventajas significativas respecto a la fibra. La principal es que no se necesita derecho de paso; basta comprar un terreno pequeño cada 50 km y construir en él una torre de microondas para saltarse el sistema telefónico y comunicarse en forma directa. Así es como MCI logró establecerse como una compañía nueva de teléfonos de larga distancia tan rápidamente.

Las microondas también son relativamente baratas. Erigir dos torres sencillas (quizá solo postes grandes con 4 cables de retén) y poner antenas en cada uno puede costar menos que enterrar 50 km de fibra a través de un área urbana congestionada o sobre una montaña, y también puede ser más económico que rentar la fibra de la compañía de teléfonos, en especial si la compañía de teléfonos aún no paga por completo el cobre que quitó cuando instaló la fibra.

Además de servir para transmisión de larga distancia, las microondas tienen otro uso importante, a saber, las bandas industriales, médicas y científicas. Estas bandas constituyen la excepción a la regla de las licencias: los transmisores que usan estas bandas no requieren la licencia del gobierno.

Ondas Infrarrojas y Milimétricas

Las ondas infrarrojas y milimétricas no guiadas se usan mucho para la comunicación de corto alcance. Todos los controles remotos de los televisores, grabadoras de video y estéreos utilizan comunicación infrarroja. Estos controles son relativamente direccionales, baratos y fáciles de construir, pero tienen un inconveniente importante: no atraviesan los objetos sólidos. En general, conforme pasamos de la radio de onda larga hacia la luz visible, las ondas se comportan cada vez más como la luz y cada vez menos como la radio.

Por otro lado, el hecho de que las ondas infrarrojas no atraviesen bien las paredes sólidas también es una ventaja. Esto significa que un sistema infrarrojo en un cuarto de un edificio no interferirá con un sistema similar en cuartos adyacentes. Además, la seguridad de los sistemas infrarrojos contra el espionaje es mejor que la de los sistemas de radio, precisamente por esta razón. Por lo mismo, no es necesario obtener licencia del gobierno para operar un sistema infrarrojo, en contraste con los sistemas de radio, que deben tener licencia.

Estas propiedades han hecho del infrarrojo un candidato interesante para las LAN inalámbricas en interiores. Por ejemplo, las computadoras y las oficinas de un edificio se pueden equipar con transmisores y receptores infrarrojos relativamente desenfocados. De esta manera, las computadoras portátiles capaces de utilizar infrarrojo pueden estar en la LAN local sin tener que conectarse a ella físicamente. Cuando varias personas se presentan en una reunión con sus máquinas portátiles, sólo tienen que sentarse en la sala de conferencias para estar conectados por completo, sin tener que enchufar. La comunicación con infrarrojo no se puede usar en exteriores porque el sol brilla con igual intensidad en el infrarrojo como en el espectro visible.

Transmisión por Ondas de Luz

La señalización óptica sin guías se ha usado durante siglos. Una aplicación moderna es conectar las LAN de dos edificios por medio de láseres montados en sus azoteas. La señalización coherente sin láseres es inherentemente unidireccional, de modo que cada edificio necesita su propio láser y su propio fotodetector. Este esquema ofrece un ancho de banda muy alto y un costo muy bajo. También es relativamente fácil de instalar y, a diferencia de las microondas, no requiere una licencia de la FCC.

La ventaja del láser, un haz muy estrecho, es aquí también una debilidad. Apuntar un rayo láser de 1 mm de anchura a un blanco de 1 mm a 500 metros de distancia requiere la puntería de una gran puntería. Por lo general, se añaden lentes al sistema para desenfocar ligeramente el rayo.

Una desventaja es que los rayos láser no pueden penetrar la lluvia ni la niebla densa, pero normalmente funcionan en días soleados.

• Ventajas y desventajas (R)

B. Modelos

I. Arquitecturas

1. El modelo OSI

- Conceptos básicos (origen, stacks, servicios, protocolos, interfaces, puntos de acceso a los servicios) (R, E)

El modelo de referencia OSI.

Este modelo se basa en una propuesta que desarrollo la Organización Internacional de Normas (ISO) como primer paso hacia una estandarización internacional de los protocolos que se usan en las diversas capas. El modelo se llama “modelo de referencia OSI (open systems interconnection, interconexión de sistemas abiertos) de la ISO” puesto que se ocupa de la conexión de sistemas abiertos, esto es, sistemas que están abiertos a la comunicación con otros sistemas.

El modelo OSI tiene siete capas. Los principios que se aplicaron para llegar a las 7 capas son:

- ✓ Se debe crear una capa siempre que se necesite un nivel diferente de abstracción.
- ✓ Cada capa debe realizar una función bien definida.
- ✓ La función de cada capa se debe elegir pensando en la definición de protocolos estandarizados internacionalmente.
- ✓ Los límites de las capas deben elegirse a modo de minimizar el flujo de información a través de las interfaces.
- ✓ La cantidad de capas debe ser suficiente para no tener que agrupar funciones distintas en la misma capa y bastante pequeña para que la arquitectura no se vuelva inmanejable.

El modelo OSI en si no es una arquitectura de red porque no especifica los servicios y protocolos exactos que se han de usar encada capa; solo dice lo que debe hacer cada capa.

OSI	
7	Aplicación
6	Presentación
5	Sesión
4	Transporte
3	Red
2	Enlace de datos
1	Física

- ✓ **Capa 1: Física.** Corresponde al hardware de red básico. Se encarga de la transmisión de cadenas de bits no estructurados sobre el medio físico; esta relacionado con las características mecánicas, eléctricas, funcionales y de procedimiento al medio físico.
- ✓ **Capa 2: Enlace de datos.** Especifica la manera de organizar los datos en cuadros y su transmisión por la red. Proporciona un servicio de transferencia de datos seguro a través del enlace físico; envía bloques de datos (tramas) llevando a cabo la sincronización, el control de errores y de flujo necesario.
- ✓ **Capa 3: Red.** Especifica la asignación de las direcciones y el reenvío de paquetes de un extremo a otro de la red. Proporciona independencia a los niveles superiores respecto a las técnicas de conmutación y de transmisión utilizadas para conectar los sistemas; es responsable del establecimiento, mantenimiento y cierre de las conexiones.
- ✓ **Capa 4: Transporte.** Especifican los detalles de la transferencia confiable. Proporciona seguridad, transferencia de datos entre los puntos finales; proporciona además procedimientos de recuperación de errores y control de flujo origen-destino.
- ✓ **Capa 5: Sesión.** Como establecer sesiones de comunicación con un sistema remoto, la especificación de detalles de seguridad, como la validación de identificación mediante contraseñas. Proporciona el control de la comunicación entre las aplicaciones; establece, gestiona y cierra las conexiones (sesiones) entre las aplicaciones cooperadoras.
- ✓ **Capa 6: Presentación.** Especifica la manera de presentar los datos. Tales protocolos son necesarios porque las diferentes marcas de computadoras usan diferentes representaciones internas para los enteros y caracteres.

Proporciona a los procesos de aplicación independencia respecto a las diferencias en la representación de los datos (sintaxis),

- ✓ **Capa 7: Aplicación.** Especifica como usa la red una aplicación particular, los detalles sobre como hace una solicitud el programa de aplicación de una máquina y como responde la otra máquina. Proporciona el acceso al entorno OSI para los usuarios y también proporciona servicios de información distribuida.

Servicios.

La verdadera función de las capas OSI consiste en proporcionar servicios a las capas superiores. Las capas pueden ofrecer dos tipos diferentes de servicios a las capas que se encuentran sobre ellas:

- ✓ **Orientados a conexión**

Se modela tomando en cuenta el sistema telefónico. Para poder hablarle a alguien se debe tomar el teléfono, marcar el numero, hablar y colgar. Similarmente, para utilizar una red de servicio orientado a conexión, el usuario del servicio establece primero una conexión, la utiliza y después termina la conexión. El aspecto fundamental de la conexiones que actúa en forma parecida a la de un tubo: el que envía, introduce objetos por un extremo, y el receptor los recoge, en el mismo orden, por le otro extremo.

Servicio	Ejemplo
Flujo de mensaje fiable	Secuencia de paginas
Flujo de octetos fiable	Conexión remota
Conexión no fiable	Voz digitalizada

- ✓ **Sin conexión**

Se modela con base en el sistema postal. Cada mensaje (carta) lleva consigo la dirección completa de destino y cada uno de ellos se encamina, en forma independiente, atreves del sistema.

Servicio	Ejemplo
Datagrama no fiable	Correo electrónico basura
Datagrama con asentimiento	Correo certificado
Pregunta-respuesta	Interrogación de base de datos

Definición de Servicio.

Esta formalmente especificado por un conjunto de primitivas (operaciones), a disposición de todos los usuarios o de otras entidades para acceder al servicio. Estas primitivas le indican al servicio que debe efectuar una acción o notifican la acción tomada por una entidad par. Las primitivas de servicio en el modelo OSI pueden dividirse en 4 clases:

Primitiva	Significado
Solicitud	Una entidad desea que el servicio realice un trabajo
Indicación	Una entidad es informada acerca de un evento
Respuesta	Una entidad desea responder a un evento
Confirmación	Una entidad va a ser informada acerca de su solicitud

Los servicios pueden ser confirmados o no confirmados. En un servicio confirmado hay una petición, una indicación, una respuesta y una confirmación. En un servicio sin confirmar, solamente hay una petición y una indicación.

Diferencia entre servicios y protocolos.

Los conceptos de servicio y protocolo tienen un significado diferente. Un servicio es un conjunto de primitivas (operaciones), que una capa proporciona a la superior. El servicio define las operaciones que la capa efectuara en beneficio de sus usuarios, pero no dice nada con respecto a como se realizan dichas operaciones. Un servicio se refiere a una interface entre dos capas, siendo la capa inferior la que provee el servicio y la capa superior la que utiliza el servicio.

Un protocolo, es un conjunto de reglas que gobiernan el formato y el significado de las tramas, paquetes o mensajes que son intercambiados por las entidades correpondientes dentro de una capa. Las entidades utilizan

protocolos para realizar sus definiciones de servicio, teniendo libertad para cambiar el protocolo, pero asegurándose de no modificar el servicio visible a los usuarios. De esta manera se observa con claridad como los conceptos de servicio y protocolo están completamente desacoplados.

- **Nivel 1: capa física (R, E)**

Tiene que ver con la transmisión de bits por un canal de comunicación. Las consideraciones de diseño tienen que ver con la acción de asegurarse de que cuando un lado envíe un bit 1, se reciba en el otro lado como bit 1, no como bit 0. Aquí las consideraciones de diseño tienen mucho que ver con las interfaces mecánica, eléctrica y de procedimientos y con el medio de transmisión físico que está bajo la capa física.

- **Nivel 2: capa de enlace de datos (R, E)**

La tarea principal de la capa de enlace de datos es tomar un medio de transmisión en bruto y transformarlo en una línea que parezca libre de errores de transmisión no detectados a la capa de red. Esta tarea la cumple al hacer que el emisor divida los datos de entrada en marcos de datos, que transmita los marcos en forma secuencial y procese los marcos de acuse de recibo que devuelve el receptor. Corresponde a la capa de enlace de datos crear y reconocer los límites de los marcos. Esto se puede lograr añadiendo patrones especiales de bits al principio y final del marco. Corresponde a esta capa resolver el problema provocado por los marcos dañados, perdidos y duplicados. Las redes de difusión tienen una consideración adicional en la capa de enlace de datos: como controlar el acceso al canal compartido. Una subcapa especial de la enlace de datos se encarga de este problema, la subcapa de acceso al medio.

- **Nivel 3: capa de red (R, E)**

Se ocupa de controlar el funcionamiento de la subred. Una consideración de diseño es determinar como se encaminan los paquetes de la fuente a su destino. Si en una subred se encuentran presentes demasiados paquetes a la vez, se estorbaran mutuamente, formando cuellos de botella. El control de tal congestión pertenece también a la capa de red. Cuando un paquete debe viajar de una red a otra para alcanzar su destino, pueden surgir muchos problemas. La capa de red debe resolver todos estos problemas para lograr que se interconecten redes heterogéneas.

- **Nivel 4: capa de transporte (R, E)**

La función básica de esta capa es aceptar datos de la capa de sesión, dividirlo en unidades más pequeñas si es necesario, pasarlo a la capa de red y asegurar que todos los pedazos lleguen correctamente al otro extremo. Además, todo esto se debe hacer de manera eficiente y en forma que aísle las capas superiores de los cambios inevitables en la tecnología del hardware. En condiciones normales, la capa de transporte crea una conexión de red distinta para cada conexión de transporte que requiera la capa de sesión. Sin embargo, si la conexión de transporte requiere un volumen de transmisión alto, la capa de transporte podría crear múltiples conexiones de red, dividiendo los datos entre las conexiones para aumentar el volumen. La capa de transporte debe lograr que la multiplexación sea transparente para la capa de sesión. La capa de transporte determina también que tipo de servicio proporcionara a la capa de sesión y, finalmente, a los usuarios de la red. Además de multiplexar varias corrientes de mensajes por un canal, la capa de transporte debe cuidar de establecer y liberar conexiones a través de la red.

- **Nivel 5: capa de sesión (R, E)**

La capa de sesión permite a los usuarios de máquinas diferentes establecer sesiones entre ellos. Una sesión permite el transporte ordinario de datos, como lo hace la capa de transporte, pero también proporciona servicios mejorados que son útiles en algunas aplicaciones. Uno de los servicios de la capa de sesión es manejar el control del dialogo. Las sesiones pueden permitir que el tráfico vaya en ambas direcciones al mismo tiempo, o solo en una dirección a la vez. Si el tráfico puede ir únicamente en un sentido a la vez, la capa de sesión puede ayudar a llevar el control de los turnos. Un servicio de sesión relacionado es el manejo de fichas. Para algunos protocolos es esencial que ambos lados no intenten la misma operación al mismo tiempo. A fin de controlar estas actividades, la capa de sesión proporciona fichas que se pueden intercalar. Solamente el lado que posea la ficha podrá efectuar la operación crítica. Otro servicio es la sincronización, la capa de sesión ofrece una forma de insertar puntos de verificación en la corriente de datos, de modo que después de cada

interrupción sólo se deban repetir los datos que se transfirieron después del último punto de verificación.

• **Nivel 6: capa de presentación (R, E)**

Realiza ciertas funciones que se piden con suficiente frecuencia para justificar la búsqueda de una solución general, en lugar de dejar que cada usuario resuelva los problemas. En particular, y a diferencia de todas las capas inferiores que se interesan solo en mover bits de manera confiable, de acá para allá, la capa de presentación se ocupa de la sintaxis y la semántica de la información que se transmite. Con el fin de hacer posible la comunicación entre computadoras con representaciones diferentes, las estructuras de datos por intercambiar se pueden definir en forma abstracta, junto con un código estándar que se use en el cable. La capa de presentación maneja estas estructuras abstractas y las convierte de la representación que se usa dentro de la computadora a la representación estándar de la red y viceversa.

• **Nivel 7: capa de aplicación (R, E)**

Contiene varios protocolos que se necesitan con frecuencia. Todo el software de terminal virtual esta en la capa de aplicación. Otra función de la capa de aplicación es la transferencia de archivos, lo mismo que el correo electrónico, la carga remota de trabajos, la búsqueda en directorios y otros recursos de uso general y especial.

• **Comparación con otras estructuras (R, E, RP)**

OSI y TCP/IP (DoD)

Los modelos de referencia OSI y TCP/IP tienen mucho en común. Ambos se basan en el concepto de un gran número de protocolos independientes. También la funcionalidad de las capas es muy similar. Por ejemplo, en ambos modelos las capas por encima de la de transporte, incluida esta, están ahí para prestar un servicio de transporte de extremo a extremo, independiente de la red, a los procesos que deseen comunicarse. Estas capas forman el proveedor de transporte. También en ambos modelos, las capas encima de la de transporte son usuarios del servicio de transporte orientados a aplicaciones.

A pesar de estas similitudes fundamentales, los dos modelos tienen también muchas diferencias.

En el modelo OSI tres conceptos son fundamentales: 1) servicios, 2) interfaces, y 3) protocolos. Es probable que la contribución más importante del modelo OSI sea hacer explícita la distinción entre estos tres conceptos. El modelo TCP/IP originalmente no distinguía en forma clara entre servicio, interfaz y protocolo, aunque se ha tratado de reajustarlo después a fin de hacerlo más parecido a OSI.

En el modelo OSI se oculta mejor los protocolos que en el modelo TCP/IP y se pueden reemplazar con relativa facilidad al cambiar la tecnología. La capacidad de efectuar tales cambios es uno de los principales propósitos de tener protocolos por capas en primer lugar.

El modelo de referencia OSI se desarrolló antes de que se inventaran los protocolos, este orden significa que el modelo no se orientó hacia un conjunto específico de protocolos, los cuales se convirtió en algo muy general. El lado malo de este orden es que los diseñadores no tenían mucha experiencia con el asunto y no supieron bien cuál funcionalidad poner en cada una. Lo contrario sucedió con TCP/IP: primero llegaron los protocolos, y el modelo fue en realidad sólo una descripción de los protocolos existentes. No hubo problema de ajustar los protocolos al modelo se ajustaban a la perfección. El único problema fue que el modelo no se ajustaba a ninguna otra pila de protocolos; en consecuencia, no fue de mucha utilidad para describir otras redes que no fueran del tipo TCP/IP.

Una diferencia obvia entre los dos modelos es la cantidad de capas: el modelo OSI tiene 7 capas y el TCP/IP 4. Ambos tienen capas de (inter)red, de transporte y aplicación, pero las demás capas son diferentes.

Otra diferencia se tiene en el área de la comunicación sin conexión frente a la orientada a conexión. El modelo OSI apoya la comunicación tanto sin conexión como la orientada a conexión en la capa de red, pero en la capa de transporte donde es más importante lo hace únicamente con la comunicación orientada a la conexión. El modelo TCP/IP solo tiene un modo en la capa de red (sin conexión) pero apoya ambos modelos en la capa de transporte, con lo que ofrece una alternativa a los usuarios. Esta elección es importante sobre todo para los protocolos simples

de petición y respuesta.

2. Evolución de las redes

• Clasificación de las redes por cobertura (R)

Las tecnologías de red se clasifican en tres categorías generales, de acuerdo con su tamaño:

- ✓ Red de área local (LAN), que puede abarcar un edificio o una instalación.
- ✓ Red de área metropolitana (MAN), que puede abarcar una ciudad.
- ✓ Red de área amplia (WAN), que abarcan instalaciones o sitios de varias ciudades, países y continentes.

• Sistemas propietarios (SNA, DNA) (R)

SNA (System Network Architecture, arquitectura de red de sistemas).

Protocolo de siete capas patentado por IBM.

SNA es una arquitectura de red que permite que los clientes de IBM construyan sus propias redes privadas, tomando en cuenta a los hostales y la subred. Por ej: un banco puede tener un a o más CPU en su departamento de proceso de datos, y numerosos terminales en cada una de sus terminales en cada una de sus sucursales. Con el uso de SNA todos estos componentes aislados pueden transformarse en un sistema coherente. Una red SNA esta constituida por una colección de maquinas denominadas nodos, de los cuales hay 4 tipos, que se caracterizan aproximadamente de la siguiente manera:

i p o 1	Son los terminales
Tipo 2	Son los controladores, es decir, son las maquinas que supervisan el funcionamiento de las terminales y los periféricos.
Tipo 4	Son los procesadores frontales, es decir aquellos dispositivos cuya función consiste en reducir la carga del CPU principal y realizar el manejo de interrupciones asociadas con la comunicación de datos.
Tipo 5	Son los hostales principales, aunque, con la aparición de los microprocesadores de bajo costo, algunos controladores han adquirido algunas propiedades de hostales.

*no hay nodos de tipo 3.

Cada uno de los nodos contiene uno o mas NAU (unidad direccionable de red, UDR), que son una pieza de software a través de la cual se permite que un proceso utilice la red; Puede considerarse como un SAP (punto de acceso a servicio), mas las entidades que proporcionan los servicios de las capas superiores. Para usar la red, el proceso debe conectarse directamente a una NAU y a partir de ese momento puede direccionarse y dirigir otras NAU. Las NAU son los puntos de entrada a la red para los procesos del usuario.

Hay 3 tipos diferentes de NAU:

Unidad Lógica(UL)	Es la variedad mas usual a la que se unen los procedimientos de usuario
Unidad física (UF)	Se proporciona una forma de direccionar en la red un dispositivo físico, sin tener en cuenta los procesos que la están utilizando. La red la utiliza para poner al nodo en línea, dejarlo fuera de línea, probarlo y ejecutar funciones parecidas al administrador de redes.
Punto de control en los servicios de sistemas (SSCP)	Tiene un conocimiento completo de, y a su vez control sobre, todos los procesadores frontales, controladores y terminales unidos o ligados al host. Normalmente hay uno por cada tipo de nodo 5 y ninguno de los otros.

Se conoce como dominio al conjunto de hardware y software manejados por un SSCP.

• Sistemas abiertos (DoD, IEEE) (R)

Sistema abierto.

Tecnología o sistema no patentado; cualquier productor puede utilizar las especificaciones de un sistema abierto para conformar productos y servicios.

El modelo de referencia TCP/IP (DoD)

La ARPANET era una red de investigación patrocinada por el DoD (Departamento de Defensa de los Estados Unidos). Al final conectó a cientos de universidades y instalaciones de gobierno usando líneas telefónicas rentadas. Cuando más tarde se añadieron redes de satélite y radio, los protocolos existentes tuvieron problemas para interactuar con ellas, de modo que se necesitó una arquitectura de referencia nueva. Esta arquitectura se popularizó después como el modelo de referencia TCP/IP, por las iniciales de sus dos protocolos primarios.

Debido a la preocupación del DoD porque algunos de sus costosos nodos, enruteadores o pasarelas de interredes pudiera ser objeto de un atentado en cualquier momento, otro de los objetivos principales fue que la red fuera capaz de sobrevivir a la pérdida de hardware de subred sin que las conversaciones existentes se interrumpieran. Es más, se necesitaba una arquitectura flexible, pues se tenía la visión de aplicaciones con requerimientos divergentes, abarcando desde la transferencia de archivos hasta la transmisión de discursos en tiempo real.

✓ La capa de interred.

Todos estos requerimientos condujeron a la elección de una capa de conmutación de paquetes basada en una capa de interred carente de conexiones. Esta capa, llamada de interred, es el eje que mantiene unida toda la arquitectura. La misión de esta capa es permitir que los nodos inyecten paquetes en cualquier red y los hagan viajar de forma independiente a su destino. Los paquetes pueden llegar incluso en un orden diferente a aquel en que se enviaron, en cuyo caso corresponde a las capas superiores reacomodarlos, si se desea la entrega ordenada. La capa de interred define un formato de paquete y protocolo oficial llamado IP. El trabajo de la capa de interred es entregar paquetes IP a donde se supone que deben ir. Aquí la consideración más importante es el ruteo de los paquetes, y también evitar la congestión.

✓ La capa de transporte.

Esta capa se diseña para permitir que las entidades pares en los nodos de origen y destino lleven a cabo una conversación. Aquí se definieron dos protocolos de extremo a extremo. El primero, TCP (transmission control protocol, protocolo de control de transmisión) es un protocolo confiable orientado a la conexión que permite que una corriente de bytes originada en una máquina se entregue sin errores en cualquier otra máquina de la interred. Este protocolo fragmenta la corriente entrante de bytes en mensajes discretos y pasa cada uno a la capa de interred. En el destino, el proceso TCP receptor reensambla los mensajes recibidos para formar la corriente de salida. El TCP también se encarga del control de flujo para asegurar que un emisor rápido no pueda abrumar a un receptor lento con más mensajes de los que pueda manejar. El segundo protocolo de esta capa, el UDP (user datagram protocol, protocolo de datagrama de usuario), es el protocolo sin conexión, no confiable, para aplicaciones que no necesitan la asignación de una secuencia ni el control de flujo del TCP y que desean utilizar los suyos propios. Este protocolo también se usa ampliamente para consultas de petición y respuesta de una sola ocasión, del tipo cliente-servidor, y en aplicaciones en las que la entrega pronta es más importante que la entrega precisa, como las transmisiones de voz o video.

✓ La capa de aplicación.

El modelo TCP/IP no tiene capas de sesión ni de presentación. La capa de aplicación contiene todos los protocolos de alto nivel. Entre los protocolos más antiguos están el de terminal virtual (TELNET), el de transferencia de archivos (FTP) y el de correo electrónico (SMTP). El protocolo de terminal virtual permite que un usuario en una máquina ingrese en una máquina distante y trabaje ahí. El protocolo de transferencia de archivos ofrece un mecanismo para mover datos de una máquina a otra en forma eficiente. El correo electrónico fue en sus orígenes sólo una clase de transferencia de archivos, pero más adelante se desarrolló para el protocolo especializado; con los años se le han añadido muchos otros protocolos, como el de servicio de nombres de dominio (DNS) para relacionar los nombres de los nodos con sus direcciones de la red; TNP, el protocolo que se usa para transferir artículos noticiosos; http, el protocolo que se usa para recuperar páginas en la World Wide Web y muchos otros.

✓ La capa del nodo a red.

El modelo de referencia TCP/IP realmente no dice mucho de lo que aquí sucede, fuera de indicar que el nodo se ha de conectar a la red haciendo uso de algún protocolo de modo que pueda enviar por ella paquetes de IP. Este protocolo no está definido y varía de un nodo a otro y de red a red.

TCP/IP	
4	Aplicación
3	Transporte
2	Interred
1	Host a red

IEEE

El IEEE (Institute of Electrical and Electronic Engineers, Instituto de Ingenieros Eléctricos y Electrónicos), tiene un grupo de estandarización que elabora estándares en las áreas de ingeniería eléctrica y computación. El estándar 802 del IEEE para redes de área local es el estándar clave para las LAN, y posteriormente fue adoptado por la ISO como base para el estándar ISO 8802.

- **Red de redes (R)**

Internet

La interred global que usa protocolos TCP/IP.

Un conjunto de redes interconectadas, desde el punto de vista del usuario, puede aparecer simplemente como una red más grande. Sin embargo, si cada una de las redes constituyentes retiene su identidad y se necesitan mecanismos especiales para la comunicación a través múltiples redes, entonces a la configuración entera se le conoce como conjunto de redes (o una Internet).

Cada red constituyente de una Internet permite la comunicación entre los dispositivos conectados a esa red; estos dispositivos se conocen como sistemas finales (ESs). Además, las redes se conectan por dispositivos denominados en los documentos ISO como sistemas intermedios (ISs). Los ISs proporcionan caminos de comunicación y realizan las funciones de retransmisión y encaminamiento necesarias para que los datos se puedan intercambiar entre los dispositivos conectados en las diferentes redes de la Internet.

3. Redes locales (LAN)

LAN (Local Área Network, red de área local).

Red que usa tecnología diseñada para abarcar un área geográfica pequeña.

Destinada a ser una alternativa a las costosas conexiones dedicadas punto a punto, los diseños varían fundamentalmente de las redes de larga distancia porque se basan en compartir la red. Cada LAN consta de un medio compartido, en general un cable al que se conectan muchas computadoras. Las computadoras se turnan el uso del medio para enviar paquetes.

La importancia de las LAN puede plantearse así: las tecnologías LAN se han convertido en la forma más común de red. Las LAN conectan ahora más computadoras que ningún otro tipo de red.

Una de las razones de que existan tantas LAN es de índole económica: las tecnologías LAN son baratas y están a la mano. Sin embargo, la razón principal de la gran demanda de las LAN puede atribuirse a n principio fundamental de la conectividad conocido como localidad de referencia, que establece que la comunicación entre un grupo de computadoras no es aleatoria, sino que sigue dos patrones. Primero, si un par de computadoras se comunican una vez, es probable que se comuniquen nuevamente en un futuro cercano y luego lo hagan periódicamente. El patrón se llama localidad temporal de referencia para hacer manifiesta la relación en el tiempo. Segundo, cada computadora tiene a comunicarse con mayor frecuencia con las computadoras cercanas. Este patrón se llama localidad física de referencia para subrayar la relación geográfica.

Una red de área local es una red de comunicaciones que interconecta varios dispositivos y proporciona un medio de intercambio de información entre ellos. Hay algunas diferencias entre las LAN y las WAN:

1. La cobertura LAN es pequeña, típicamente un edificio o como mucho un conjunto de edificios próximos.
2. Es común que la LAN sea propiedad de la misma entidad que es propietaria de los dispositivos conectados a la red. En WAN, esto no es tan corriente, o al menos una fracción significativa de recursos de la red son ajenos. Esto tiene dos implicaciones. La primera es que se debe cuidar mucho la elección de la LAN, ya que lleva acarreado una inversión substancial de capital tanto en la adquisición como en el mantenimiento. Segunda, la responsabilidad de la gestión de la LAN recae solamente en el usuario.

3. Las velocidades de transmisión internas en una LAN son mucho mayores.

Tradicionalmente en LAN se utiliza la difusión en lugar de utilizar técnicas de commutación. En una red de difusión no hay nodos intermedios. En cada estación hay un trasnmsisor/receptor que se comunica con las otras estaciones a través de un medio compartido. Una transmisión desde cualquier estación se recibirá por todas las otras estaciones. Los datos se transmiten en forma de paquetes. Debido a que el medio es compartido, una y solo una estación en cada instante de tiempo podrá transmitir el paquete. Más recientemente, la commutación también se esta utilizando en LAN, fundamentalmente en LAN tipo Ethernet.

Las redes de área local, llamadas LAN son redes de propiedad privada dentro de un solo edificio. Se usan ampliamente para conectar computadoras personales y estaciones de trabajo en oficinas de compañías y fabricas con objeto de compartir recursos e intercambiar información. Se distinguen de otro tipo de redes por tres características: 1) su tamaño, 2) su tecnología de transmisión, y 3) su topología.

Las LAN están restringidas en tamaño, lo cual significa que el tiempo de trasmisión del peor caso esta limitado. Conocer este limite hace posible usar ciertos tipos de diseños que de otra manera no serían prácticos, y también simplifica la administración de la red.

Las LAN usan una tecnología de transmisión que consiste en un cable sencillo al cual están conectadas todas las máquinas. Las LAN tradicionales operan a velocidades de 10 a 100 Mbps, tienen bajo retardo y experimentan muy pocos errores.

Las LAN de transmisión puede tener diversas topología. En una red de bus, en cualquier instante una computadora es la máquina maestra y puede transmitir; se pide a las otras máquinas que se abstengan de enviar mensajes. Es necesario un mecanismo de arbitraje para resolver conflictos cuando dos o más máquinas quieren trasmitir simultáneamente. El mecanismo de arbitraje puede ser centralizado o distribuido.

El segundo tipo de sistema de difusión es el anillo. En un anillo, cada bit se propaga por si mismo, sin esperar al resto del paquete al cual pertenece. Cada bit recorre el anillo entero en el tiempo que toma transmitir unos pocos bits, a veces antes de que el paquete completo se haya transmitido. Como en todos los sistemas de difusión, se necesitan reglas para arbitrar el acceso simultáneo al anillo.

Las redes de difusión se pueden dividir también en estáticas y dinámicas, dependiendo de cómo se asigna el canal. Una asignación estática típica divide el tiempo en intervalos discretos y ejecuta un algoritmo de asignación cíclica, permitiendo a cada máquina transmitir únicamente cuando le llaga su turno. La asignación estática desperdicia la capacidad del canal cuando una máquina no tiene nada que decir durante su segmento asignado, por lo que muchos sistemas intentan asignar el canal dinámicamente (es decir, por demanda).

• Mecanismos de acceso al medio (*Polling, CSMA/CD y Token Passing*) (R, E)

CSMA/CD

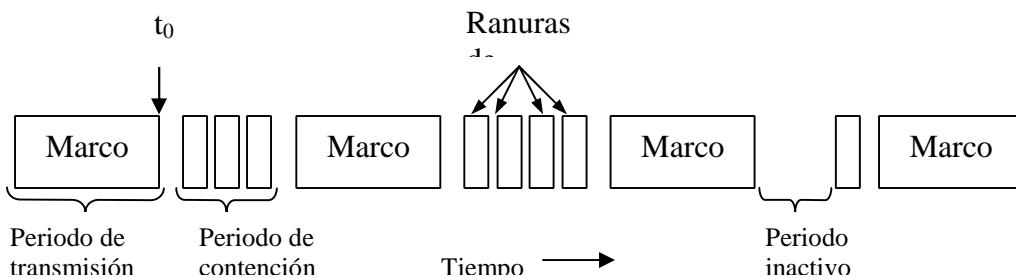
Red CSMA que tiene la capacidad de detectar los errores que resulten al transmitir simultáneamente varias estaciones.

El resultado de la interferencia entre dos señales se llama colisión. Aunque las colisiones no dañan a hardware, producen una transmisión alterada que impide que los dos cuadros sean recibidos correctamente. Para manejar dichas situaciones, la Ethernet solicita a los transmisores que vigilen las señales del cable para asegurar que ninguna otra computadora transmite al mismo tiempo. Al detectar una computadora transmisora una colisión, interrumpe la transmisión. La vigilancia del cable se conoce como acceso múltiple con detección de portadora y detección de colisión.

La CSMA/CD hace más que detectar colisiones: también se recupera de ellas. Tras una colisión, la computadora debe esperar que el cable quede otra vez en reposo antes de transmitir. Sin embargo, si las computadoras comienzan a transmitir tan pronto como se estabiliza el éter, sucederá otra colisión. Para evitar colisiones múltiples, la Ethernet equiere que las computadoras esperen tras una colisión antes de retransmitir.

CSMA/CD (carrier sense multiple access with collision detection, acceso múltiple con detección de portadora y detección de colisiones), se usa ampliamente en las LAN en la subcapa MAC.

CSMA/CD utiliza el modelo conceptual de la figura:



En el punto marcado t_0 , una estación ha terminado de transmitir su marco. Cualquier otra estación que tenga un marco por enviar puede intentar hacerlo. Si dos o más estaciones deciden transmitir simultáneamente, habrá una colisión. Las colisiones pueden detectarse observando la potencia o el ancho de pulso de la señal recibida y comparándola con la señal transmitida.

Una vez que una estación detecta una colisión, aborta la transmisión, espera un período de tiempo aleatorio e intenta de nuevo, suponiendo que ninguna otra estación ha comenzado a transmitir durante ese lapso. Por tanto, el modelo CSMA/CD consistirá en períodos alternantes de contención y transmisión, ocurriendo períodos muertos cuando todas las estaciones están calladas.

Es importante darse cuenta de que la detección de colisiones es un proceso analógico. El hardware de la estación debe escuchar el cable mientras transmite. Si lo que lee es distinto de lo que puso en él, sabe que está ocurriendo una colisión. La implicación es que la codificación de la señal debe permitir que se detecten colisiones. Por esta razón, comúnmente se usa una codificación especial. Es importante hacer notar que ningún protocolo de subcapa MAC garantiza la entrega confiable. Incluso en ausencia de colisiones, el receptor podría no haber copiado correctamente el marco por varias razones.

Token Passing (pase de fichas o paso de testigo)

Técnica usada en las redes de topologías de anillo para el control de la transmisión. La ficha consiste de un mensaje especial enviado por el anillo. Cuando una estación tiene paquetes por enviar, la estación espera que llegue la ficha, transmite un paquete y luego la ficha.

El protocolo MAC más usual en redes LAN con topología en anillo es el de paso de testigo. La técnica de anillo con paso de testigo se basa en el uso de una trama pequeña, denominada testigo (token), que circula cuando todas las estaciones están libres. Cuando una estación desea transmitir debe esperar a que le llegue el testigo. En este caso, toma el testigo, cambiando uno de sus bits, lo que lo convierte en la secuencia de comienzo en las tramas de datos. Posteriormente, la estación añade y transmite el resto de campos requeridos en la construcción de la trama.

Cuando una estación toma el testigo y comienza a transmitir, en el anillo deja de estar presente el testigo, de manera que el resto de estaciones que deseen transmitir deben esperar. La trama en el anillo realiza una vuelta completa y se absorbe o drena en la estación transmisora, que insertará un nuevo testigo en el anillo cuando se cumplan una de las dos condiciones siguientes:

- ✓ La estación haya terminado la transmisión de su trama.
- ✓ Los bits iniciales de la trama transmitida hayan vuelto a la estación (después de una vuelta completa al anillo).

Si la longitud del anillo es menor que la longitud de la trama, la primera condición implica la segunda. En caso contrario, una estación debería liberar el testigo después de que haya terminado de transmitir, pero antes de que comience a recibir su propia transmisión; la segunda condición no es estrictamente necesaria, relajándose en ciertas circunstancias. La ventaja que implica la suposición de la segunda condición es que no asegura que, en un instante de tiempo dado, solo pueda haber una trama de datos en el medio y solo puede estar transmitiendo una estación, simplificándose los procedimientos de recuperación de errores.

Una vez que se ha insertado un nuevo testigo en el anillo, la siguiente estación en la secuencia que disponga de datos a transmitir podrá tomar el testigo y llevar a cabo la transmisión.

En condiciones de baja carga, el anillo con pase de testigo presenta cierta ineficacia debido a que una estación debe esperar a recibir el testigo antes de transmitir. Sin embargo, en condiciones de alta carga, que es la situación más preocupante, el anillo funciona como un sistema de turno rotatorio, que es eficiente además de equitativo.

La principal ventaja del anillo con paso de testigo es el control de acceso flexible que ofrece. La principal desventaja esta en la necesidad de procedimientos para realizar el mantenimiento del anillo. La perdida de testigo impide posteriores utilizaciones del anillo. La duplicidad del testigo puede interrumpir también el funcionamiento del anillo. Se puede seleccionar una estación monitora para asegurar que solo hay un testigo y para reinsertar un testigo libre en caso necesario.

• Estándar IEEE 802.3: Ethernet (R, E)

El estándar IEEE 802.3 es para una LAN CSMA/CD persistente-1. Cuando una estación quiere transmitir, escucha el cable. Si el cable esta ocupado, la estación espera hasta que se desocupa; de otra manera, transmite de inmediato. Si dos o más estaciones comienzan a transmitir por un cable inactivo, habrá una colisión. Todas las estaciones en colisión terminan entonces su transmisión, esperan un tiempo aleatorio y repiten de nuevo todo el proceso.

Este sistema fue bautizado Ethernet por el éter luminífero, a través del cual alguna vez se pensó que se propagaba la radiación electromagnética. El estándar 802.3 publicado difiere de la especificación Ethernet en cuanto a que describe una familia completa de sistemas CSMA/CD persistente-1, operando a velocidades de 10 Mbps en varios medios.

Cableado del 802.3

Comúnmente se utilizan 5 tipos de cableado:

Nombre	Cable	Segmento máximo	Nodos/seg	Ventajas
10Base5	Coaxial grueso	500 m	100	Bueno para backbone
10Base2	Coaxial delgado	200 m	30	Sistema más barato
10Base-T	Parte trenzado	100 m	1024	Fácil mantenimiento
10Base-F	Fibra óptica	2000 m	1024	Mejor entre edificios

Históricamente, llegó primero el cable 10Base5, llamado comúnmente Ethernet grueso; semeja una manguera de jardín amarilla, con marcas cada 2.5 metros para indicar los puntos de las derivaciones. Las conexiones a él se hacen generalmente usando derivaciones vampiro, en las que se introduce cuidadosamente una punta hasta la mitad del núcleo del cable coaxial. La notación 10Base5 significa que opera a 10 Mbps, usa señalización banda base y puede manejar segmentos de hasta 500 metros.

El segundo tipo de cable fue el 10Base2 o Ethernet delgado. Las conexiones de hacen usando conectivos BNC estándar de la industria para formar uniones T, estas son más fáciles de usar y son más confiables. El Ethernet delgado es mucho más barato y fácil de instalar, pero solo puede extenderse 200 metros y puede manejar solo 30 máquinas por segmento de cable.

La detección de rupturas de cable, derivaciones malas y conectores sueltos puede ser un problema importante en ambos medios. Los problemas asociados con la localización de rupturas de cable han empujado a los sistemas a un tipo de patrón alambrado diferente, en el que todas las estaciones tienen alambres que conducen a un concentrador (hub) central. Generalmente estos alambres son pares trenzados telefónicos. Este esquema se llama 10Base-T.

Para 10Base5 se sujetó firmemente un transceptor (transceiver) alrededor del cable, de modo que su derivación haga contacto con el núcleo interno. El transceptor contiene la electrónica que maneja detección de portadora y detección de colisiones. Al detectarse una colisión, el transceptor también pone una señal no válida especial en el cable para asegurar que todos los demás transceptores también se den cuenta de que ha ocurrido una colisión.

Con 10Base5, un cable transceptor (transceiver cable) conecta el transceptor a una tarjeta de interfaz en la computadora. El cable de transceptor puede tener hasta 50 metros de longitud y contiene 5 pares trenzados aislados individualmente. Dos de los pares son para entrada y salida de datos, respectivamente, dos más son para entrada y salida de señales de control, el quinto par, permite que la computadora energice la electrónica del transceptor.

Algunos transceptores permiten la conexión de hasta 8 computadoras cercanas a ellos, a fin de reducir la cantidad de transceptores necesarios.

Con 10Base2, la conexión al cable es solo un conector BNC pasivo de unión T. La electrónica del transceptor esta en la tarjeta controladora, y cada estación tiene siempre su propio transceptor.

Con 10Base-T no hay cable en absoluto, solo el concentrador (hub). Agregar o remover estaciones es sencillo con esta configuración, y las rupturas de cable pueden detectarse con facilidad. La desventaja de 10Base-T es que la longitud máxima del cable es de solo 100 m, tal vez 150 m si se usa par trenzado de alta calidad (categoría 5).

Una cuarta opción de cableado para 802.3 es 10Base-F, que usa fibra óptica. Esta alternativa es cara debido al costo de los conectores y los terminadores, pero tiene excelente inmunidad contra el ruido y es el método a usar en conexiones entre edificios o entre concentradores muy separados.

Cada versión del 802.3 tiene una longitud máxima de cable por segmento. Para permitir redes mayores, se pueden conectar cables múltiples mediante repetidores. Un repetidor es un dispositivo de capa física que recibe, amplifica y retransmite señales en ambas direcciones. En lo que concierne al software, una serie de segmentos de cable conectados mediante repetidores no es diferente de un solo cable. Un sistema puede contener múltiples segmentos de cable y múltiples repetidores, pero ningún par de repetidores puede estar separado por más de 2.5 km y ninguna trayectoria entre dos transceptores puede atravesar más de 4 repetidores.

Codificación Manchester.

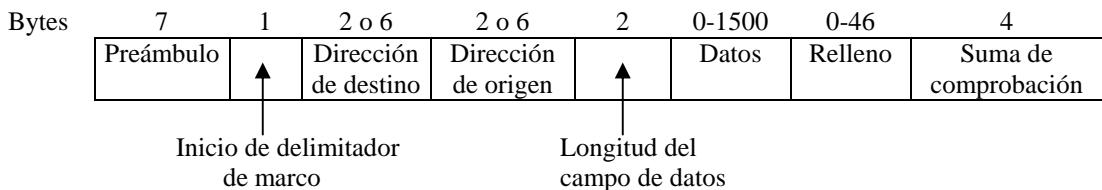
Ninguna de las versiones del 802.3 usa codificación binaria directa con 0 volts para un bit 0 y 5 volts para un bit 1, pues conduce a sus ambigüedades. Si una estación envía la cadena de bits 0001000, otros podrían interpretarla falsamente como 10000000 o 01000000, pues no pueden distinguir entre un transmisor inactivo (0 volts) y un bit 0 (0 volts).

Lo que se necesita es un mecanismo para que los receptores determinen sin ambigüedad el comienzo, el final o la mitad de cada bit sin referencia a un reloj externo. Dos de tales enfoques se llaman codificación Manchester y codificación Manchester diferencial. En la codificación Manchester, cada periodo de bit se divide en dos intervalos iguales. Un bit binario 1 se envía teniendo el voltaje alto durante el primer intervalo y bajo durante el segundo. Un 0 binario es justo lo inverso: primero bajo y después alto. Este esquema asegura que cada periodo de bits tiene una transición a la mitad, facilitando que el receptor se sincronice con el transmisor. Una desventaja de la codificación Manchester es que requiere el doble del ancho de banda que la codificación binaria directa, pues los pulsos son de la mitad de ancho.

La codificación Manchester diferencial, es una variación de la codificación Manchester básica. En ella, un bit 1 se indica mediante la ausencia de una transición al comienzo del intervalo. Un bit 0 se indica mediante la presencia de una transición al inicio del intervalo. En ambos casos, también hay una transición a la mitad. El esquema diferencial requiere equipo más complejo, pero ofrece mejor inmunidad al ruido. Todos los sistemas 802.3 de banda base usan codificación Manchester debido a su sencillez. La señal alta es de +0.85 volts, y la señal baja es de -0.85 volts, dando un valor de cc de 0 volts.

Protocolo de subcapa MAC de 802.3

La estructura de marco del 802.3 se muestra:



Cada marco comienza con un preámbulo de 7 bytes, cada uno de los cuales contienen un patrón de bits 10101010. A continuación viene un byte de inicio de marco que contiene 10101011 para indicar el inicio del marco mismo.

El marco contiene dos direcciones, una para el destino y otra para el origen. El estándar permite direcciones de 2 bytes y de 6 bytes, pero los parámetros definidos para el estándar de banda base de 10 Mbps usan solo direcciones

de 6 bytes. El bit de orden mayor de la dirección de destino es 0 para direcciones ordinarias y 1 para direcciones de grupo.

- Estándar IEEE 802.4: *Token Bus* (R, E)

El estándar 802.4 describe una LAN llamada token bus (bus de ficha). Físicamente, el token bus es un cable lineal o en forma de árbol al que se conectan las estaciones. Las estaciones están organizadas lógicamente en forma de anillo, donde cada estación conoce la dirección de la estación a su izquierda y a su derecha. Cuando se inicializa el anillo lógico, la estación de número más alto puede enviar el primer marco. Hecho esto, pasa el permiso a su vecino inmediato enviándole un marco de control especial llamado ficha (token). La ficha se propaga alrededor del anillo lógico, teniendo permiso de transmitir marcos solo quien tiene la ficha. Dado que solo una estación a la vez tiene la ficha, no ocurren colisiones.

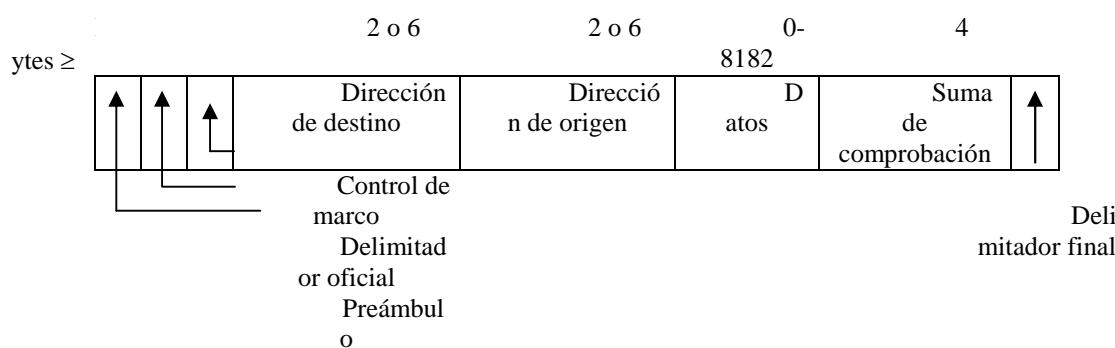
No es importante el orden físico en el que están conectadas las estaciones al cable. Dado que el cable inherentemente es un medio de difusión, todas las estaciones reciben todos los marcos. Descartando los que no estén dirigidos a ellas. Cuando una estación pasa la ficha, envía un marco de ficha dirigido específicamente a su vecino lógico en el anillo, sin importar la ubicación física de esa estación en el cable. También cabe la pena notar que, cuando inicialmente se prenden las estaciones, no están en el anillo, por lo que el protocolo de MAC contempla la adición y eliminación de estaciones del anillo.

Para la capa física, el token bus usa un cable coaxial de 75 ohms de banda ancha empleado para la televisión por cable. Se permiten tanto sistemas de cable sencillo como doble, con o sin terminadores. Se permiten tres esquemas básicos de modulación: codificación por desplazamiento de frecuencia de fase continua, codificación por desplazamiento de fase coherente y codificación por desplazamiento de fase de amplitud modulada multinivel duobinaria. Son posibles velocidades de 1, 5 y 10 Mbps. Los esquemas de modulación no solo proporcionan modos de representar 0, 1 e inactivo en el cable, sino también otros tres símbolos que sirven para el control de la red. En conjunto, la capa física es totalmente incompatible con el 802.3, y mucho más complicada.

Protocolo de subcapa MAC de token bus.

Cuando se inicializa el anillo, las estaciones se insertan en orden por dirección de estación, de mayor a menor. La entrega de la ficha también se hace por dirección, de mayor a menor. Cada vez que una estación adquiere una ficha, puede transmitir marcos durante cierta cantidad de tiempo; después debe pasar la ficha a otra estación. Si los marcos son lo bastante cortos, pueden enviarse varios marcos consecutivos. Si una estación no tiene datos, pasa la ficha inmediatamente después de recibirla.

El token bus define cuatro clases de prioridad, 0, 2, 4 y 6 para tráfico, siendo 0 la menor y 6 la mayor. Se muestra el formato de marco de token bus:



El preámbulo sirve para sincronizar el reloj del receptor. Los campos de delimitador inicial y delimitador final se usan para arcar los límites de los marcos. Estos dos datos contienen codificación analógica de símbolos distintos de ceros y unos, por lo que no pueden ocurrir accidentalmente en los datos de usuario. Como resultado no se necesita campo de longitud. El campo de control de marco se usa para distinguir los marcos de datos de los de control. En los marcos de datos este campo lleva la prioridad del marco; también puede llevar un indicador que pida a la estación destino reconocer la recepción correcta o incorrecta del marco. En los campos de control el campo de control de marco se usa para especificar el tipo de marco. Los tipos permitidos incluyen pase de ficha y varios marcos de mantenimiento del anillo, incluidos los mecanismos para permitir la entrada de estaciones nuevas en el anillo y la salida de estaciones del anillo, otros. Los campos de dirección de destino y dirección de origen son los mismos que en el 802.3. Como en el 802.3, una red dada debe usar

solo direcciones de 2 bytes o de 6 bytes, no una mezcla, en el mismo cable. El campo de datos puede tener hasta 8182 bytes de longitud cuando se usan direcciones de dos bytes y de hasta 8174 bytes cuando se usan direcciones de 6 bytes. El campo suma de comprobación sirve para detectar errores de transmisión. Se usa el mismo algoritmo y el mismo polinomio que en el 802.3.

• Estándar IEEE 802.5: Token Ring (R, E)

Un token ring opera como un solo medio compartido. Cuando una computadora necesita enviar datos, debe esperar permiso antes de acceder a la red. Una vez obtenido el permiso, el transmisor tiene el control completo del anillo. Cuando el trasmisor envía un cuadro, los bits pasan sucesivamente a las siguientes computadoras, hasta que dan la vuelta al anillo y regresan al transmisor.

Todas las estaciones reenvían los bits por el anillo. Por lo tanto, para comprobar que no han ocurrido errores de transmisión, el trasmisor puede comparar los datos recibidos con los enviados. Las demás estaciones vigilan todas las transmisiones. Si un cuadro está destinado a una computadora dada, esta hace una copia de cuadro al ir pasando los bits por el anillo.

A diferencia de una Ethernet, la transmisión token ring no depende del CSMA/CD. En cambio, el hardware de token ring coordina todas las computadoras conectadas para asegurar que el permiso pasa por turno a todas las computadoras. La coordinación usa un mensaje especial reservado, llamado token o ficha. La ficha es un patrón de bits que difiere de los cuadros normales de datos. Para asegurar que los datos normales no puedan interpretarse como fichas, algunas tecnologías de token ring utilizan relleno de bits para cambiar temporalmente la aparición de fichas en los datos de transmisión. Más importante, el hardware de token ring asegura que solo exista una ficha en la red de token ring.

En esencia, la ficha da permiso a las computadoras de transmitir cuadros. Por lo tanto, antes de enviar uno, cada computadora debe esperar la llegada de la ficha. Al llegar, la computadora retira temporalmente la ficha y usa el anillo para transmitir datos. Aunque puede tener as de un cuadro para transmisión, la computadora solo envía un cuadro y luego transmite la ficha. A diferencia de los cuadros de datos, que dan la vuelta completa al anillo, la ficha viaja de una computadora a la siguiente, que entonces puede usar la red para transmitir.

Si todas las computadoras de una red token ring tienen datos por enviar, el esquema de pase de ficha garantiza que se turnaran, que cada una transmitirá un cuadro antes de pasar la ficha. Si alguna no tiene datos por transmitir, simplemente pasa la ficha sin demora.

Las redes en anillo no son nada nuevo, pues se han utilizado desde hace mucho para redes tanto locales como de área amplia. Entre sus muchas características atractivas está el que un anillo no es realmente un medio de difusión, sino un conjunto de enlaces punto a punto individuales que, coincidentemente, forman un círculo. Los enlaces punto a punto implican una tecnología bien entendida y probada en el campo que puede operar en par trenzado, cable coaxial y fibra óptica. La ingeniería de anillos es casi completamente digital. Un anillo también es equitativo y tiene un límite superior conocido de acceso a canal. Por estas razones, IBM escogió el anillo como su LAN y el IEEE ha incluido el estándar token ring como el 802.5.

Un anillo en realidad consiste en un conjunto de interfaces de anillo conectadas por líneas punto a punto. Cada bit que llega a una interfaz se copia en un buffer de 1 bit y luego se copia en el anillo nuevamente. Mientras está en el buffer, el bit se puede inspeccionar y posiblemente modificar antes de enviarse. Este paso de copiado introduce un retardo de 1 bit en cada interfaz.

En un token ring (anillo con ficha) circula un patrón de bit especial, llamado ficha (token) alrededor del anillo cuando todas las estaciones están inactivas. Cuando una estación quiere transmitir un marco, debe tomar la ficha y retirarla del anillo antes de transmitir. Esta acción se lleva a cabo invirtiendo un solo bit de la ficha de 3 bytes, lo que instantáneamente la convierte en los tres primeros bytes de un marco de datos normal. Debido a que solo hay una ficha, solo una estación puede transmitir en un instante dado, resolviendo por lo tanto el acceso al canal de la misma manera en que 1 resuelve el token bus.

Una implicación del diseño token ring es que el anillo mismo debe tener un retardo suficiente para contener una ficha completa que circule cuando todas las estaciones están inactivas. El retardo tiene dos componentes: el retardo de 1 bit introducido por cada estación y el retardo de propagación de señal.

Las interfaces del anillo tienen dos modos operativos, escuchar y transmitir. En el modo de escuchar, los bits de entrada simplemente se copian en la salida, con un retardo de un tiempo de un bit. En el modo de transmitir, al que se entra una vez que se tiene la ficha, la interfaz rompe la conexión entre la entrada y la salida, introduciendo sus propios datos en el anillo. Para poder conmutar del modo de escuchar al de transmitir en un tiempo de 1 bit, que la interfaz generalmente necesita almacenar en su buffer uno o más marcos, el lugar de obtenerlos de la estación con tan poca anticipación.

A medida que los bits propagados alrededor del anillo regresan, son retirados por el transmisor. La estación transmisora puede guardarlos, para compararlos con los datos originales y verificar la confiabilidad del anillo, o descartarlos. Como el marco completo nunca aparece en el anillo en un mismo instante, esta arquitectura de anillo no pone límite al tamaño de los marcos. Una vez que una estación ha terminado de transmitir el último bit de su último marco, debe regenerar la ficha. Cuando ha regresado el último bit del marco, debe retirarse, y de inmediato la interfaz debe conmutarse nuevamente al modo de escuchar para evitar retirar las fichas que podría seguir si ninguna otra estación la ha retirado.

El manejo de acuses de recibo es directo en un token ring. El formato de marco sólo necesita incluir un campo de 1 bit para acuse de recibo, inicialmente 0. Cuando la estación de destino ha recibido un marco, establece el bit. Por supuesto, si el acuse significa que la suma de comprobación ha sido verificada, el bit debe seguir a la suma de comprobación y la interfaz con el anillo debe ser capaz de verificar la suma de comprobación tan pronto haya sido recibido el último bit. Cuando un marco se difunde a múltiples estaciones, debe usarse un mecanismo de acuse más complicado.

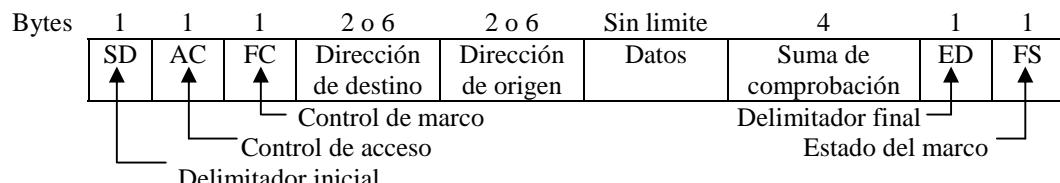
Cuando el tráfico es ligero, la ficha pasará la mayor parte del tiempo circulando inactivamente alrededor del anillo. Ocasionalmente, una estación la tomará, transmitirá un marco, y entonces emitirá una ficha nueva. Sin embargo, cuando el tráfico es pesado, de modo que hay una cola en cada estación, tan pronto una estación termina su transmisión y regenera la ficha la siguiente estación vera y retirara la ficha. De esta manera, el permiso para enviar gira continuamente, por turno circular. La eficiencia de la red puede acercarse al 100% en condiciones de carga pesada.

En la capa física, el 802.5 especifica par trenzado operando a 1 o 4 Mbps, aunque IBM posteriormente introdujo una versión de 16 Mbps. Las señales se codifican usando codificación Manchester diferencial, siendo alto (HI) y bajo (LO) señales positivas y negativas de magnitudes absolutas de 3.0 a 4.5 volts. Normalmente, la codificación Manchester diferencial usa HI-LO o LO-HI para cada bit, pero el 802.5 también usa HI-HI y LO-LO en ciertos bytes de control. Estas señales que no son de datos ocurren en pares consecutivos y no introducen una componente de CC en el voltaje del anillo.

Un problema con las redes de anillo es que, si se rompe el cable en alguna parte, el anillo se inhabilita. Este problema puede resolverse mediante el uso de un centro de alambrado. Si bien la red sigue siendo lógicamente un anillo, físicamente cada estación está conectada al centro de alambrado que contiene dos pares trenzados, uno de datos a la estación y otro de datos de la estación. Dentro del centro de alambrado hay relevadores de paso que se energizan mediante corriente de las estaciones. Si se rompe el anillo o se inactiva una estación, la perdida de la corriente de operación desactiva el relevador, poniendo en puente la estación. El anillo entonces puede continuar su operación tendiendo en puente la sección dañada.

Protocolo de subcapa MAC del token ring.

La operación básica del protocolo de MAC es sencilla. Cuando no hay tráfico en el anillo circula continuamente una ficha de 3 bytes, esperando que una estación la tome cambiando un bit 0 específico a bit 1, lo que convierte la ficha en una secuencia de inicio de marco. La estación entonces envía el resto de un marco de datos normal.



En condiciones normales, el primer bit del marco recorrerá el anillo y regresará al transmisor antes de que el marco completo haya sido transmitido. Solo un anillo muy grande será capaz de contener incluso un marco muy corto. En

consecuencia, la estación transmisora debe drenar el anillo mientras continua transmitiendo. Esto significa que los bits que han completado el viaje alrededor del anillo regresan al transmisor y ahí son removidos.

Los campos de delimitador inicial y delimitador final marcan el inicio y fin del marco. El byte de control de acceso contiene el bit de ficha y también el bit de revisión, los bits de prioridad y los bits de reservación. El byte de control de marco distingue los marcos de datos de los diferentes marcos de control posibles. Los campos de dirección destino y dirección origen son iguales que en el 802.3 y el 802.4. A estos marcos le siguen los datos, que pueden ser tan grandes como sea necesario, siempre y cuando el marco aun pueda transmitirse dentro del tiempo de retención de la ficha (10 mseg). El campo suma de comprobación también es como en 802.3 y 802.4. El delimitador final contiene un bit que se establece cuando cualquier interfaz detecta un error. También contiene un bit que puede servir para marcar el último marco de una secuencia lógica, algo parecido a un bit de fin de archivo.

El protocolo 802.5 tiene un elaborado esquema para manejar marcos con distintas prioridades. El marco de ficha de 3 bytes contiene un campo en el byte de en medio que indica la prioridad de la ficha. Cuando una estación quiere transmitir una estación de prioridad n, debe esperar hasta que pueda capturar la ficha cuya prioridad sea menor o igual que en. Además, al pasar un marco de datos, una estación puede tratar de reservar la siguiente ficha escribiendo la prioridad del marco que quiere enviar en los bits de reservación del marco. Sin embargo, si se ha reservado ahí una prioridad mayor, la estación podría no hacer la reservación. Cuando el marco actual ha terminado, se genera la siguiente ficha con la prioridad reservada.

- **Fast Ethernet (R, E)**

Versión de la tecnología Ethernet que opera a 100 Mbps.

Ethernet a alta velocidad (Fast Ethernet) es un conjunto de especificaciones desarrolladas por el comité IEEE 802.3 con el fin de proporcionar una red LAN de bajo coste compatible con Ethernet que funcione a 100 Mbps. La designación genérica para estos estándares es 10Base-T. El comité definió varias alternativas para diferentes medios de transmisión.

Todas las opciones 100Base-T usan el protocolo MAC y el formato de la trama IEEE 802.3. 100Base-X identifica al conjunto de opciones que usan las especificaciones del medio físico definidas originalmente para FDI. Todos los esquemas 100Base-X emplean dos enlaces físicos entre los nodos; uno para transmisión y otro para recepción. 100Base-X hace uso de pares trenzados apantallados (STP) o cables de pares trenzados no apantallados (UTP) de alta calidad (clase 5). 100Base-FX hace uso de fibra óptica.

Cualquiera de las opciones 100Base-X requiere la instalación de nuevo cableado. En estos casos, 100Base-T4 define una alternativa menos costosa que puede utilizar UTP de voz de clase 3 además de UTP de clase 5 de alta calidad. Para alcanzar los 100 Mbps en cables de baja calidad. 100Base-T4 especifica el uso de 4 líneas de par trenzado entre los nodos, de los cuales tres se usan simultáneamente para la transmisión de datos en una dirección.

La topología de todas las opciones 100Base-t es similar a la de 10Base-T, que corresponde a una estrella.

	100Base-TX	100Base-FX	100Base-T4
Medio de transmisión	2 pares clase 5	2 fibras ópticas	4 pares, clase 3, 4 o UTP 5
Técnica de señalización	MLT-3	4B5B, NRZI	8B6T, NRZ
Velocidad de transmisión	100 Mbps	100 Mbps	100 Mbps
Longitud máxima del segmento	100 m	100 m	100 m
Cobertura de la red	200 m	200 m	400 m
			200 m

El concepto principal en que se basa el Ethernet rápido (802.3u) es sencillo: mantener todos los formatos de paquete, interfaces y reglas de procedimientos anteriores, y simplemente reducir el tiempo de bit de 100 nseg a 10 nseg. Técnicamente, habrá sido posible copiar el 10Base-5 o el 10Base-2 y aun detectar colisiones a tiempo con solo reducir la longitud máxima del cable por un factor de 10. Sin embargo, las ventajas del alambrado 10Base-T fueron tan contundentes que el Ethernet rápido se basa completamente en este diseño. Por tanto, todos los sistemas Ethernet rápido usan concentradores; no se permiten cables de derivación múltiple con derivaciones vampiro ni conectores BNC.

• FDDI (R)

FDDI (Fiber Distributed Data Interconnect, interconexión de datos distribuidos por fibra). Tecnología de LAN que usa fibra óptica para interconectar estaciones en una topología de anillo.

La FDDI es una tecnología token ring que puede transmitir datos a razón de 100 millones de bits por segundo, ocho veces más rápido que las redes Token Ring de IBM y 10 veces más rápido que las Ethernet. Para dar razones de datos tan altas, la FDDI usa fibra óptica, en lugar de cables de cobre, para conectar las computadoras.

La FDDI usa redundancia para recuperarse de las fallas. Una red FDDI contiene dos anillos completos, uno que sirve para transmitir datos cuando todo está bien y otro que funciona cuando falla el primero. Se dice que los anillos de una red FDDI opera en contragiro, porque los datos fluyen por el segundo anillo en dirección opuesta a la dirección de flujo de datos del anillo principal.

Se dice que una red FDDI es autoreparable porque el hardware puede detectar fallas graves y recuperarse de ellas automáticamente. Para lograrlo, la FDDI usa un par de anillos en contragiro. Uno de los anillos sirve para la transmisión de datos. Al ocurrir una falla que rompe el anillo, las estaciones adyacentes se reconfiguran automáticamente y emplean el segundo anillo para atajar la falla.

La FDDI (interfaz de datos distribuidos por fibra) es una LAN token ring de fibra óptica de alto desempeño que opera a 100 Mbps y distancias de hasta 200 km con hasta 1000 estaciones conectadas. Esta red puede usarse de la misma manera que cualquiera de las LAN 802 pero, con su gran ancho de banda, otro uso común es como backbone para conectar varias LAN de cobre.

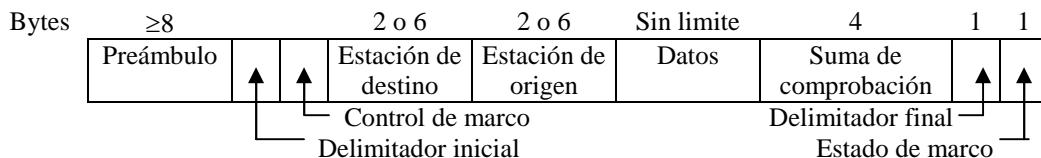
La FDDI usa fibras multimodo, pues el gasto adicional de las fibras monomodo no es necesario para redes que solo operan a 100 Mbps; también usa led en lugar de láser, no solo debido a su menor costo, sino también porque la FDDI a veces puede conectarse a estaciones de trabajo de usuario. La especificación de diseño de la FDDI pide no más de un error por cada 2.5×10^{10} bits.

El cableado de la FDDI consiste en dos anillos de fibra, uno que transmite en dirección de las manecillas del reloj y otro en dirección contraria. Si se rompe cualquiera de ellos, se puede usar el otro como respaldo. Si ambos se rompen en el mismo punto, pueden unirse los dos anillos en uno solo de aproximadamente el doble de longitud. Cada estación contiene relevadores que pueden servir para unir los dos anillos o para saltar la estación en caso de problemas con ella.

No usa codificación Manchester, en cambio se usa un esquema llamado 4 de 5. Cada grupo de 4 símbolos MAC se codifica en el medio como grupo de 5 bits. Diecisésis de las 32 combinaciones son para datos, 3 son para delimitadores, 2 son de control, 3 para señalamiento al hardware y 8 no se usan. La ventaja de este esquema es que ahorra ancho de banda, pero la desventaja es la perdida de la propiedad de autosincronización de la codificación Manchester. Para compensar esta perdida se usa un preámbulo grande para sincronizar el reloj del receptor con el reloj del transmisor.

Los protocolos FDDI básicos se usan en gran medida en los protocolos 802.5. Para transmitir datos, una estación debe primero capturar la ficha, luego transmite un marco y lo retira cuando regresa nuevamente. Una diferencia entre la FDDI y el 802.5 es que, en el 802.5 una estación no puede generar una ficha nueva hasta que su marco ha dado la vuelta completa y ha regresado. En la FDDI se decidió permitir que una estación ponga nuevamente una ficha en el anillo tan pronto ha terminado de transmitir sus datos. En un anillo grande puede haber varios marcos a la vez.

Formato de marco FDDI:



Los campos de delimitador inicial y delimitador final marcan los límites del marco. El campo de control del marco indica el tipo de marco de que se trata. El byte de estado de marco contiene bits de reconocimiento. Los otros campos son análogos al 802.5.

El protocolo MAC de la FDDI usa tres temporizadores. El temporizador de retención de la ficha determina el tiempo que una estación puede continuar transmitiendo una vez que ha adquirido la ficha. Este temporizador evita que una estación acapare permanentemente el anillo. El temporizador de rotación de la ficha se reinicia cada vez que se ve la ficha. Si termina el temporizador, significa que no se ha visto la ficha durante una ranura demasiado grande; probablemente se ha perdido, por lo que se inicia el procedimiento de recuperación de la ficha. Por último, el temporizador de transmisión valida se usa para terminar la temporización y recuperación de ciertos errores transitorios en el anillo.

La FDDI también tiene un algoritmo de prioridad parecido al de 802.4: determina las clases de prioridad que pueden transmitir durante un paso dado de la ficha. Si la ficha se adelanta a lo programado, todas las prioridades pueden transmitir, pero si está retrasada, solo pueden transmitir las mayores.

- **Comunicación en LAN's (LAN switching) (R, E)**

Se llama conmutada a una tecnología de red si el hardware incluye un dispositivo electrónico que se conecta a varias computadoras y permite que envíen y reciban datos. Una LAN conmutada consiste en un dispositivo electrónico que transfiere cuadros entre varias computadoras.

Físicamente, un conmutador se asemeja a un concentrador. La diferencia estriba en el modo en que operan los dispositivos: un concentrador simula un medio compartido; un conmutador simula una LAN en puente con una computadora por segmento.

La ventaja principal de las LAN conmutadas sobre los concentradores es la misma que la de las redes LAN en puente sobre los segmentos únicos: el paralelismo. Puesto que un concentrador simula un segmento compartido por todas las computadoras, cuando mucho dos computadoras pueden comunicarse por él al mismo tiempo. Sin embargo, en las LAN conmutadas cada computadora tiene un segmento de LAN simulado exclusivo para ella. Como resultado, hasta la mitad de las computadoras conectadas al conmutador puede transmitir simultáneamente.

Para la transmisión de datos a larga distancia, más allá de un entorno local, la comunicación se realiza generalmente mediante la transmisión de datos desde el origen hasta el destino a través de una red de nodos de conmutación intermedios. Este diseño de red conmutada se usa también a veces para implementar redes LAN. El contenido de los datos no es de interés de los nodos de conmutación, sino que el propósito de estos últimos es proporcionar un servicio de conmutación que posibilite el intercambio de datos entre nodos hasta que alcancen su destino.

Diversas consideraciones se pueden realizar acerca de las redes de comunicación conmutadas:

1. Algunos nodos se conectan con otros nodos, siendo su única tarea la conmutación interna (en la red) de los datos. Otros nodos tienen también conectadas una o más estaciones, de modo que además de sus funciones de conmutación estos nodos aceptan datos desde y hacia las estaciones conectadas a ellos.
2. Los enlaces entre nodos están normalmente multiplexados, utilizándose multiplexación por división en frecuencias (FDM) o por división en el tiempo (TDM).
3. Generalmente, la red no está completamente conectada; es decir, no existe un enlace directo entre cada posible pareja de nodos. Sin embargo, siempre resulta deseable tener más de un camino posible a través de la red para cada par de estaciones. Esto mejora la fiabilidad o seguridad de la red.

En las redes conmutadas de área amplia se emplean dos tecnologías diferentes: conmutación de circuitos y conmutación de paquetes. Estas dos tecnologías difieren en la forma en que los nodos conmutan la información entre enlaces en el camino desde el origen hasta el destino.

- **Características y servicios de los sistemas operativos para LAN's (R, E, RP)**

- 4. Redes metropolitanas (MAN)**

- **Protocolos para MAN (R, E)**

Una red de área metropolitana, o MAN es básicamente una versión más grande de una LAN y normalmente se basa en una tecnología similar.

• Características y servicios de las MAN's (R, E, RP)

Una MAN es básicamente una versión más grande de una LAN y normalmente se basa en una tecnología similar. Podría abarcar un grupo de oficinas corporativas cercanas o una ciudad y podría ser privada o pública. Una MAN puede manejar datos, voz, e incluso podría estar relacionada con la red de televisión por cable local. Una MAN solo tiene uno o dos cables y no contiene elementos de conmutación, los cuales desvían los paquetes por una de varias líneas de salida potenciales. Al no tener que conmutar, se simplifica el diseño.

La principal razón para distinguir las MAN como una categoría especial es que se ha adoptado un estándar para ellas: se llama DQDB (distributed queue dual bus, o bus dual de cola distribuida) o 802.6. El DQDB consiste en dos buses (cables) unidireccionales, a los cuales están conectadas todas las computadoras.

5. Redes amplias (WAN)

WAN (Wide Área Network, red de área amplia).

Red que usa tecnología diseñada para abarcar un área geográfica grande.

Generalmente, se considera como redes de área amplia a todas aquella que cubren una extensa área geográfica, requieren atravesar rutas de acceso público, y utilizan parcialmente circuitos proporcionados por una entidad proveedora de servicios de telecomunicaciones. Típicamente, una WAN consiste en una serie de dispositivos de conmutación interconectados. La transmisión generada por cualquier dispositivo se encaminara a través de estos nodos internos hasta alcanzar el destino. A estos nodos no les concierne el contenido de los datos, al contrario, su función es alcanzar su destino final.

Tradicionalmente, la WAN se ha implementado usando una de las dos tecnologías siguientes: conmutación de circuitos y conmutación de paquetes. Aunque últimamente, se está empleando como solución la técnica de retransmisión de tramas (frame relay), así como las redes ATM.

• Mecanismos de acceso al medio (R, E)

Hay un tipo de WAN que usa canales de acceso múltiple: las WAN basadas en comunicación satelital. Las WAN usan enlaces punto a punto, excepto en las redes satelitales.

Se emplean 5 clases de protocolos de acceso múltiple (acceso ascendente): sondeo, ALOHA, FDM, TDM y CDMA. El problema principal es con el canal de enlace ascendente, ya que el de enlace descendente solo tiene un transmisor (satélite) y por lo tanto no tiene problema con el reparto de canal.

- ✓ **Sondeo.** La forma tradicional de repartir un solo canal entre varios competidores es que alguien los sondee. Hacer que el satélite sondee por turno cada estación para ver si tiene un marco es prohibitivamente caro, dado el tiempo de 270 msec requerido para cada secuencia de sondeo / respuesta. Sin embargo, si todas las estaciones de tierra también están conectadas a una red de conmutación de paquetes, es concebible una variación menor este concepto. La idea es disponer todas las estaciones en un anillo lógico, de modo que cada estación conozca su sucesor. Por este anillo terrestre circula una ficha. El satélite nunca ve la ficha. Solo se permite a una estación transmitir por el enlace ascendente cuando ha capturado la ficha. Si el número de estaciones es pequeño y constante, el tiempo de transmisión de la ficha es corto y las ráfagas enviadas por el canal de enlace ascendente son mucho más grandes que el tiempo de rotación de la ficha, el esquema es moderadamente eficiente.
- ✓ **ALOHA.** El ALOHA puro es fácil de implementar: cada estación simplemente envía cuando quiere. El problema es que la eficiencia del canal es de solo 18%. El uso de ALOHA ranurado duplica la eficiencia, pero agrega el problema de cómo sincronizar todas las estaciones para que sepan en qué momento comienza su intervalo. Afortunadamente el satélite mismo inherentemente es un medio de difusión. Una estación de tierra, la estación de referencia transmite periódicamente una señal especial cuya retransmisión es usada por todas las estaciones de tierra como origen del tiempo. Puesto que los relojes operan con tasas ligeramente diferentes, es necesaria una resincronización periódica para mantener a todos en fase. Una complicación más es que el tiempo de propagación desde el satélite es diferente para cada estación de tierra, pero este efecto puede

compensarse. Para aumentar el aprovechamiento del canal de enlace ascendente, podemos pasar del canal de enlace ascendente individual al esquema de enlace ascendente doble. Una estación con un marco por transmitir escoge uno de los canales de enlace ascendente al azar y envía el marco a la siguiente ranura. Cada enlace ascendente opera entonces un canal ALOHA ranurado independiente. Si uno de los canales de enlace ascendente contiene un solo marco, simplemente se transmite después en la ranura de enlace descendente. Si ambos canales tienen éxito, el satélite puede poner en búfer uno de los marcos y transmitirlo durante una ranura inactiva posterior.

- ✓ **FDM (Multiplexión por división en frecuencia).** Un transpondedor típico de 36 Mbps puede dividirse estáticamente en unos 500 canales PCM de 64 kbps, operando cada uno en su propia frecuencia para evitar la interferencia con los demás. La FDM también tiene algunos inconvenientes. Primero, se requieren bandas de seguridad (laterales) entre los canales para mantener separadas las estaciones, porque no es posible construir transmisores que envíen toda su energía por la banda principal y nada por las bandas laterales. Segundo, debe controlarse cuidadosamente la potencia de las estaciones. Si una estación pone demasiada potencia en la banda principal, también pondrá automáticamente demasiada potencia en las bandas laterales, ocurriendo un derramamiento en los canales adyacentes que causa interferencia. Por último, la FDM es una técnica completamente analógica que no se presta bien a una implementación en software. Si el número de estaciones es pequeño y fijo, pueden repetirse por adelantado los canales de frecuencia. Sin embargo, si el número de estaciones, o la carga de cada una de ellas, puede fluctuar con rapidez, se necesita algún tipo de reparto dinámico de las bandas de frecuencia.
- ✓ **TDM (Multiplexión por división del tiempo).** Requiere sincronización de tiempo para las ranuras, pero esto puede proporcionarse mediante una estación de referencia. Para un número pequeño e invariable de estaciones, la asignación de ranuras puede establecerse por adelantado y no cambiarse nunca, pero con un número variable de estaciones, o un número fijo de estaciones con carga variable en el tiempo, las ranuras de tiempo deben asignarse dinámicamente. La asignación de ranuras puede hacerse de manera centralizada o descentralizada. También es posible el reparto dinámico de ranuras TDM. Existen tres esquemas, en cada uno de ellos se dividen los marcos TDM en ranuras de tiempo, dando a cada ranura un dueño (temporal). Solo el dueño puede usar una ranura de tiempo. El primer esquema supone que hay más ranuras que estaciones, por lo que puede asignarse una ranura inicial a cada estación. Si hay más ranuras que estaciones, las ranuras extra no se asignan a nadie. Si el dueño de una ranura no la quiere durante el grupo actual, se desperdicia. Una ranura vacía es una señal para todos los demás que el dueño no tiene tráfico. Durante el siguiente marco, la ranura queda disponible para quien la quiera, con base en contención. Si el dueño quiere recuperar su ranura especial, transmite un marco, obligando a una colisión (si hay otro tráfico). Tras la colisión, todos, excepto el dueño, deben desistir de usar la ranura durante el siguiente marco. Por tanto el dueño siempre puede comenzar a transmitir en un lapso de dos tiempos de marco. Es aplicable el segundo esquema aun cuando el número de estaciones es desconocido y variable. En este método, las ranuras no tienen dueños permanentes. En cambio, las estaciones compiten por ranuras usando ALOHA ranurado. Cuando tiene éxito una transmisión, la estación que hace la transmisión también tiene derecho a esa ranura durante el siguiente lapso. Así mientras una estación tenga datos por enviar, podrá continuar haciéndolo indefinidamente. Un tercer esquema requiere que las estaciones hagan solicitudes por adelantado antes de transmitir. Cada marco contiene una ranura especial, que se divide en V subranuras menores empleadas para hacer reservaciones. Cuando una estación quiere enviar datos, difunde un marco corto de solicitud en una subranura de reserva escogida al azar. Si la reserva tiene éxito (no hay una colisión), entonces se reserva la siguiente ranura normal. Todos deben conocer la longitud de la cola (cantidad de ranuras reservadas) en todo momento, para que al lograr una estación una reserva exitosa, sepa cuantas ranuras de datos deben pasar antes de transmitir. Cuando el tamaño de la cola cae en cero, todas las ranuras se revierten a subranuras de reserva a fin de acelerar el proceso de reserva. Aunque TDM se usa ampliamente, tanto con esquemas de reserva como sin ellos, tiene algunos inconvenientes. Por una parte, requiere que todas las estaciones se sincronicen en el tiempo. También requiere que cada estación de tierra pueda alcanzar velocidades de ráfaga extremadamente altas.
- ✓ **CDMA (Code Division Multiple Access, acceso múltiple por división de código).** El CDMA evita el problema de sincronización de tiempo y también el problema de reparto de canal; es completamente descentralizado y totalmente dinámico. Sin embargo, tiene tres desventajas principales. Primero la capacidad de un canal CDMA ante la presencia de ruido y de estaciones descoordinadas es típicamente menor de lo que se puede lograr con TDM. Segundo, con 128 chips/bit, aunque la tasa de bits no sea alta, la tasa de chips sí lo es, necesitándose un transmisor rápido. Tercero, pocos ingenieros entienden realmente el CDMA, lo que no aumenta la probabilidad de que lo usen.

- **Protocolos para WAN's (R, E)**

- **Commutación y ruteo (R, E)**

En lugar de usar un medio compartido o un conmutador electrónico para mover los paquetes de una computadora a otra, las WAN se construyen con muchos conmutadores. El tamaño inicial de la WAN se determina por la cantidad de sitios y de computadoras conectadas.

El conmutador electrónico básico de las WAN se llama conmutador de paquetes porque mueve paquetes completos de una conexión a otra. Cada conmutador de paquetes es una pequeña computadora con procesador, memoria y dispositivos de E/S que se usa para enviar y recibir paquetes.

Los sistemas de área amplia por conmutación de paquetes usan la técnica de almacenamiento y reenvío en la que los paquetes que llegan por un conmutador se colocan en una cola hasta que el conmutador puede reenviarlos a su destino. La técnica que permite que un conmutador de paquetes maneje en búfer descargas cortas de paquetes que llegan simultáneamente.

Para manejar unas cuantas computadoras adicionales, puede aumentarse la capacidad de un conmutador agregando hardware de interfaz de E/S o una CPU más rápida. El concepto fundamental que hace posible la construcción de una WAN de gran capacidad es la capacidad de conmutación que puede aumentarse sin agregar computadoras nuevas. En particular, pueden agregarse conmutadores de paquetes al interior de una red para manejar la carga; tales conmutadores no necesitan tener computadoras conectadas. Los llamaremos conmutadores interiores, y a los que se conectan computadoras directamente, conmutadores exteriores.

Para que una LAN opere correctamente, tanto los conmutadores interiores como exteriores deben tener una tabla de enrutamiento y deben reenviar paquetes. Además, los valores de la tabla de enrutamiento deben garantizar lo siguiente.

- ✓ **Enrutamiento universal.** La tabla de enrutamiento del conmutador debe contener una ruta de siguiente salto a cada destino posible.
- ✓ **Rutas optimas.** En el conmutador, la cifra de siguiente salto de la tabla de enrutamiento para un destino dado debe apuntar a la trayectoria más corta al destino.

Puede pensarse en el enrutamiento en una WAN como una grafica que modela la red. Cada nodo de la grafica responde a un conmutador de paquetes de la red. Si la red contiene una conexión directa entre un par de conmutadores de paquetes, la grafica contiene una arista o enlace entre los nodos correspondientes.

Casi todos los sistemas WAN incluyen un mecanismo para eliminar el enrutamiento duplicado. Llamado ruta predeterminada o entrada predeterminada de tabla de enrutamiento, el mecanismo permite que una sola entrada de la tabla de enrutamiento reemplace una lista grande de entradas con la misma cifra de siguiente salto. Solo se permite una entrada predeterminada en una tabla de enrutamiento, y la entrada tiene menor prioridad que las demás. Si el mecanismo de reenvío no encuentra una entrada explícita para un destino dado, usa la entrada predeterminada.

- **Características y servicios (R, E, RP)**

Las WAN deben crecer según se necesite para conectar muchos sitios distribuidos en grandes distancias geográficas, con muchas computadoras en cada uno. Una tecnología no se clasifica como WAN a menos que pueda ofrecer un rendimiento razonable en una red grande. Esto es, las WAN no conectan simplemente muchas computadoras de muchos sitios, sino que deben tener la capacidad suficiente para permitir que las computadoras se comuniquen simultáneamente.

Para formar una WAN, se interconecta un grupo de conmutadores de paquete. En general, los conmutadores tienen varios conectores de E/S, lo que permite establecer diferentes topologías. El conmutador de paquetes es el bloque de construcción básico de las WAN. Se forma una WAN con la conexión de un grupo de conmutadores de paquetes a los que luego se conectan las computadoras. Se pueden agregar conmutadores e interconexiones para aumentar la capacidad de la WAN.

Se han desarrollado muchas tecnologías Wan, incluyendo ARPANET, X.25, ISDN, relé de trama, SMDS y ATM. Actualmente, el relé de trama y SMDS ofrecen servicios de WAN de alta velocidad. El ATM podría ser más

importante en el futuro.

Una red de área amplia o WAN se extiende sobre un área geográfica extensa, a veces un país o un continente; contiene una colección de máquinas dedicadas a ejecutar programas de usuario.

En muchas redes de área amplia, la subred tiene dos componentes distintos: las líneas de transmisión y los elementos de conmutación. Las líneas de transmisión (también llamadas circuitos, canales o troncales) mueven bits de una máquina a otra.

Los elementos de conmutación son computadoras especializadas que conectan dos o más líneas de transmisión. Cuando los datos llegan por una línea de entrada, el elemento de conmutación debe escoger una línea de salida para reenviarlos.

En casi todas las WAN, la red contiene numerosos cables o líneas telefónicas, cada una conectada a un par de enrutadores. Si dos enrutadores que no comparten un cable desean comunicarse, deberán hacerlo indirectamente, por medio de otros enrutadores. Cuando se envía un paquete de un enrutador a otro a través de uno o más enrutadores intermedios, el paquete se recibe completo en cada enrutador intermedio, se almacena hasta que la línea de salida requerida esté libre, a continuación se reenvía. Una subred basada en este principio se llama, punto a punto, de almacenar y reenviar, o de paquete conmutado. Casi todas las redes de área amplia (excepto aquellas que usan satélites) tienen subredes de almacenar y reenviar. Cuando los paquetes son pequeños y el tamaño de todos es el mismo, suele llamarse celdas.

Cuando se usa una subred punto a punto, una consideración de diseño importante es la topología de interconexión del enrutador. Las redes locales que fueron diseñadas como tales usualmente tienen una topología simétrica. En contraste, las redes de área amplia típicamente tienen topologías irregulares.

Una segunda posibilidad para una WAN es un sistema de satélite o de radio en tierra. Cada enrutador tiene una antena por medio de la cual puede enviar y recibir. Todos los enrutadores pueden oír las salidas enviadas desde el satélite y en algunos casos pueden también oír la transmisión ascendente de los otros enrutadores hacia el satélite. Algunas veces los enrutadores están conectados a una subred punto a punto de gran tamaño y únicamente algunos de ellos tienen una antena de satélite. Por su naturaleza, las redes de satélite son de difusión y son más útiles cuando la propiedad es importante.

II. Protocolos e interfaces de comunicación de datos

Protocolo.

Diseño que especifica los detalles sobre la manera en que se relacionan las computadoras, incluyendo el formato de los mensajes que intercambian y el manejo de los errores.

1. Protocolos e interfaces de bajo nivel

• Interfaces en la capa física: RS-232, V.35, X.21, RS-449 (R, E)

RS-232

Una norma de la EIA se ha convertido en el modo más difundido de transferencia de caracteres por alambres de cobre entre una computadora y un dispositivo, como un modem, teclado o terminal. La norma EIA RS 232-C, comúnmente abreviada RS-232, especifica los detalles de la conexión física y los detalles eléctricos. Debido a que la RS-232 está diseñada para usarse con dispositivos como módems y terminales, especifica la transmisión de caracteres. En general, cada carácter consta de siete bits de datos.

La RS-232 define una comunicación serial asíncrona. Se llama serial porque los bits viajan por el alambre uno detrás otro. La RS-232 permite al transmisor el envío de un carácter en cualquier momento y un retardo arbitrariamente grande antes del envío de otro.

X.21

El protocolo de la capa física X.21, especifica la interfaz física, eléctrica y de procedimientos entre el host y la red. En realidad, muy pocas redes públicas manejan este estándar, pues requiere señalamiento digital en lugar de analógico en las líneas telefónicas.

- **Protocolos orientados a bytes y a caracteres (R, E)**
- **Protocolos orientados a bits (R, E)**
- **Control de errores en los protocolos de bajo nivel (R, E)**
- **Control de flujo en los protocolos de bajo nivel (R, E)**

Las computadoras no operan a la misma velocidad. Los rebasamientos de datos ocurren cuando una computadora envía datos por una red más rápido de lo que el destino puede absorberlos. En consecuencia hay perdida de información.

Hay varias técnicas para manejar los rebasamientos de datos. En conjunto, las técnicas se conocen como mecanismos de control de flujo. La manera más sencilla de control de flujo es el sistema de parada y continuación, en el que el transmisor espera tras la transmisión de cada paquete. Cuando está listo el receptor para otro paquete, envía un mensaje de control.

Aunque los protocolos de parada y continuación evitan el rebasamiento, también pueden hacer un uso muy ineficiente de ancho de banda de la red. Para obtener mayores rendimientos, los protocolos usan una técnica de control de flujo conocida como ventana deslizante. El transmisor y el receptor se programan para usar un tamaño de ventana fijo, que es la cantidad máxima de datos que pueden enviar antes de llegar a un acuse de recibo. El transmisor y el receptor pueden acordar un tamaño de ventana. El transmisor comienza por los datos que va a enviar, extrae los suficientes para llenar la primera ventana y los transmite. Si se requiere confiabilidad, retiene copias en caso de que se necesite hacer una retransmisión. El receptor debe tener espacio de búfer listo para recibir la ventana completa. Al llegar en secuencia un paquete, el receptor lo pasa a la aplicación receptora y transmite un acuse de recibo. Al llegar este el transmisor descarta la copia del paquete que le corresponde y transmite el siguiente paquete.

Las ventanas deslizantes pueden aumentar notablemente el rendimiento. Para entender por qué, considere la secuencia de transmisiones con un esquema de parada y continuación y otro de ventana deslizante. En el protocolo de parada y continuación, tras enviar un paquete, el protocolo espera un acuse de recibo antes de enviar el siguiente. En la secuencia de transmisiones al utilizar ventana deslizante, el protocolo envía todos los paquetes de la ventana y luego espera los acuses de recibo.

- **Protocolos de capa de red: X.25 (R, E)**

X.25

Cada red X.25 consta de dos o más conmutadores de paquetes X.25 interconectados mediante líneas arrendadas. Las computadoras conectadas a los conmutadores pueden enviar y recibir paquetes.

Debido a que X.25 se inventó antes de que fueran comunes las computadoras personales, muchas redes X.25 se diseñaron para conectar terminales ASCII a computadoras remotas de tiempo compartido. La red da una comunicación de dos vías. Cuando el usuario ingresa datos con el teclado una interfaz X.25 los captura, los coloca en paquetes X.25 y los transmite por la red. Cuando el programa que se ejecuta en la computadora genera una salida, la pasa a la interfaz X.25, que la coloca en un paquete X.25 y lo transmite a la pantalla del usuario. Aunque el X.25 puede usarse para comunicación entre computadoras, la tecnología es cara para el desempeño que ofrece y tiene límites de velocidad.

Muchas redes públicas antiguas siguen un estándar llamado X.25 que la CCITT desarrolló durante la década de 1970 para proveer una interfaz entre las redes públicas de conmutación de paquetes y sus clientes. La mayor parte de las redes X.25 trabajan a velocidades de hasta 64 kbps, lo cual las hace obsoletas para muchos propósitos. No obstante, su uso aún es extenso.

X.25 está orientado a la conexión y trabaja con los circuitos virtuales tanto conmutados como permanentes. Un circuito virtual conmutado se crea cuando una computadora envía un paquete a la red y pide que se haga una llamada a una computadora remota. Una vez establecida la conexión, los paquetes se pueden enviar por ella y siempre llegan en orden. X.25 proporciona control de flujo para asegurar que un emisor rápido no pueda abrumar a un receptor lento u ocupado.

Un circuito virtual permanente se usa de la misma forma que uno comutado pero se establece previamente por un acuerdo entre el cliente y la portadora; siempre esta presente y no requiere una llamada que lo establezca para poder usarlo. Un circuito de este tipo es similar a una línea rentada.

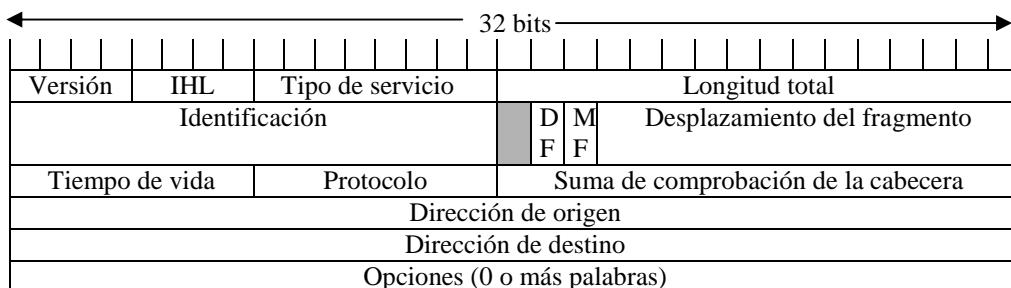
2. Conjunto de protocolos TCP/IP

• Protocolos a nivel de red: IP, ARP, RARP, ICMP (R, E)

IP (Internet Protocol, protocolo de Internet).

El pegamento que mantiene unida la Internet es el protocolo de capa de red, IP. Este se diseña desde el principio con la interconexión de redes en mente.

Un datagrama IP consiste en una parte de cabecera y una parte de texto. La cabecera tiene una parte fija de 20 bytes y una parte opcional de longitud variable. Se transmite en orden big endian: de izquierda a derecha, comenzando por el bit de orden mayor del campo de versión. En las máquinas little endian (Pentium), se requiere conversión por software tanto para la transmisión como para la recepción.



El campo de versión lleva el registro de la versión del protocolo al que pertenece el datagrama. Dado que la longitud de la cabecera no es constante, se incluyen un campo en la cabecera IHL para indicar la longitud en palabras de 32 bits. El valor mínimo es de 5, cifra que aplica cuando no hay opciones. El valor máximo de este campo de 4 bits es de 15. El campo de tipo de servicio permite al host indicar a la subred el tipo de servicio que quiere. El mismo campo contiene (de izquierda a derecha) un campo de procedencia; tres indicadores, D, T y R; y dos bits no usados. El campo de procedencia es una prioridad, de 0 (normal) a 7 (paquete de control de red). Los tres bits indicadores permiten al host especificar lo que le interesa más del grupo (retardo, rendimiento, confiabilidad). La longitud total incluye todo el datagrama: tanto la cabecera como los datos. La longitud máxima es de 65,535 bytes. Viene un bit sin uso y luego dos campos de 1 bit. DF significa no fragmentar; es una orden para los enruteadores de que no fragmenten el datagrama, porque el destino es incapaz de juntar las piezas de nuevo. MF significa más fragmentos. Todos los fragmentos, excepto el último tienen establecido este bit, que es necesario para saber cuándo han llegado todos los fragmentos de un datagrama. El desplazamiento del fragmento indica en qué parte del datagrama actual va este fragmento. Todos los fragmentos excepto el último del datagrama deben tener un múltiplo de 8 bytes, que es la unidad de fragmento elemental. El campo tiempo de vida es un contador que sirve para limitar la vida de un paquete. Se supone que este contador cuenta el tiempo en segundos, permitiendo una vida máxima de 255 seg. El campo de protocolo indica la capa superior de transporte a la que debe entregarse. La suma de comprobación de la cabecera verifica solamente la cabecera. Tal suma de comprobación es útil para detección de errores generados por palabras de memoria erróneas en un enruteador. El algoritmo es sumar todas las medias palabras de 16 bits a medida que llegan, usando aritmética de complemento a uno, y luego obtener el complemento a uno del resultado. La dirección de origen y la dirección de destino indican el número de red y el número de host. El campo de opciones se diseñó para proporcionar un recurso que permitiera que las versiones subsiguientes del protocolo incluyeran información no presente en el diseño original. El campo de opciones se rellena para completar múltiplos de 4 bytes. Actualmente hay cinco opciones definidas:

Opción	Descripción
Seguridad	Especifica qué tan secreto es el datagrama
Enrutamiento estricto desde el origen	Indica la trayectoria completa a seguir
Enrutamiento libre desde el origen	Da una lista de los enruteadores que no deben evitarse
Registrar ruta	Hace que cada enruteador agregue su dirección de IP
Marca de tiempo	Hace que cada enruteador agregue su dirección y su marca de tiempo

ARP (Address Resolution Protocol, protocolo de resolución de dirección).

Protocolo usado por una computadora para correlacionar una dirección IP con una dirección de hardware. Las computadoras que laman al ARP difunden una solicitud a la que responde la computadora objetivo.

Para que todas las computadoras estén de acuerdo con el formato exacto y el significado de los mensajes usados para resolver las direcciones, la familia de protocolos TCP/IP incluye un protocolo de resolución de direcciones (ARP). La norma ARP define dos tipos básicos de mensaje: solicitud y respuesta. Un mensaje de solicitud tiene una dirección IP y solicita la dirección de hardware correspondiente; una respuesta tiene tanto la dirección IP enviada con la solicitud como la dirección de hardware.

La norma ARP especifica la manera de enviarse mensajes ARP por una red. Especifica que debe colocarse un mensaje ARP de solicitud en un cuadro y difundirse a todas las computadoras de la red. Cada computadora recibe la solicitud y examina la dirección IP. La computadora mencionada en la solicitud transmite una respuesta: los demás procesan y descartan la solicitud sin transmitir respuestas.

Cuando una computadora transmite una respuesta ARP, esta no se difunde, sino que se pone en un cuadro y se envía directamente a la computadora que emitió la solicitud.

Aunque el formato de mensaje ARP es lo bastante general para permitir direcciones de protocolo y de hardware arbitrarias, el ARP casi siempre se usa para ligar direcciones IP de 32 bits a direcciones Ethernet de 48 bits.

Se muestra el formato de un mensaje ARP cuando el protocolo se utiliza con direcciones de protocolo IP (cuatro octetos) direcciones hardware Ethernet (seis octetos):

0	8	16	24	31
TIPO DIRECCIÓN		TIPO DIRECCIÓN PROTOCOLO		
LONG DIR H	LONG DIR P	OPERACIÓN		
DIR H TRANSMISOR (primeros 4 octetos)				
DIR H TRANSMISOR (últimos 2 octetos)		DIR P OBJETIVO (primeros 2 octetos)		
DIR P TRANSMISOR (últimos 2 octetos)		DIR H OBJETIVO (primeros 2 octetos)		
DIR H OBJETIVO (últimos 4 octetos)				
DIR P OBJETIVO (los 4 octetos)				

Cada línea de la figura corresponde a 32 bits de un mensaje ARP. Los primeros dos campos de 16 bits contienen valores que especifican los tipos de dirección de hardware y protocolo. El campo *tipo dirección hardware* contiene 1 cuando se usa ARP con Ethernet y el campo *tipo dirección protocolo* contiene 0x0800 cuando se usa ARP con IP. El segundo par de campos *long dir h* y *long dir p*, especifican la cantidad de octetos de una sola dirección de hardware y una de protocolo. El campo *operación* indica si el mensaje es una solicitud o una respuesta.

Cada mensaje ARP contiene campos para dos ligas de direcciones. Una liga corresponde al transmisor y la otra al destinatario, que ARP llama objetivo. Al enviar una solicitud, el transmisor no sabe la dirección de hardware del objetivo. Por lo tanto, el campo *dir h objetivo* de la solicitud ARP puede llenarse con ceros, pues el contenido no se usa. En la respuesta, la liga del objetivo se refiere a la computadora que envió la solicitud. Así, la información de objetivo no cumple ninguna función.

La ventaja ARP sobre los archivos de configuración es la sencillez. El administrador del sistema no necesita hacer mucho, excepto asignar a cada máquina una dirección IP y decidir sobre las máscaras de subred. El ARP hace lo demás.

Son posibles varias optimizaciones para hacer más eficiente el ARP. Por principio de cuentas, una vez que una computadora ha ejecutado el ARP, pone en caché el resultado por si necesita establecer contacto con la misma máquina pronto. La siguiente vez encontrara la proyección en su propio caché, eliminando por tanto la necesidad de una segunda difusión.

Otra optimización es hacer que cada máquina difunda su proyección al arranque. Esta difusión generalmente se hace en la forma de un ARP que busca su propia dirección IP. No debe haber una respuesta, pero un efecto secundario de la difusión de la difusión es crear una entrada en el caché de ARP de todos. Si llega una respuesta, es

que dos máquinas tienen asignada la misma dirección de IP. La nueva debe informar de ello al administrador del sistema y no debe arrancar.

Para permitir el cambio de las proyecciones, por ejemplo, cuando se descompone una tarjeta Ethernet y se reemplaza por una nueva, las entradas en el caché de ARP deben terminar su temporización al transcurrir algunos minutos.

RARP (Reverse Address Resolution Protocol, protocolo de resolución de dirección en reversa).

Los protocolos TCP/IP incluyen el RAR que sirve para que una computadora determine su propia dirección IP. La computadora manda una solicitud RARP al servidor, que regresa una respuesta RARP con la dirección IP de la computadora.

En el caso del RARP aunque la computadora al arrancar no sabe su dirección IP, si conoce su dirección de hardware. De hecho, el RARP solo puede emplearse en redes como Ethernet que hacen que la dirección de hardware de cada computadora sea permanente en la NIC de hardware. El RARP lee de la NIC la dirección de hardware de la computadora, la pone en un mensaje de solicitud RARP y manda la solicitud al servidor. El servidor extrae la dirección de hardware, consulta la base de datos que especifica la ligadura entre una dirección de hardware y una dirección IP y regresa un mensaje de respuesta RARP con la dirección IP de la computadora. Para permitir que el RARP transmita y reciba cuadros, el sistema debe inicializar el hardware de red antes de emplear el RARP para configurar el IP.

A cuestión de cómo una computadora puede comunicarse con un servidor antes de que conozca la dirección del servidor es fácil de responder: simplemente, la computadora difunde una solicitud, por ejemplo, al transmitir un cuadro con una solicitud RARP, la computadora difunde la dirección de hardware como dirección destino (el RARP solo puede emplearse en redes que manejan difusión). Cuando una red tiene varios servidores, la difusión de una solicitud puede dar como resultado varias respuestas. Si una computadora difunde una solicitud por una red a la que se conectan varios servidores cada enrutador recibirá una copia y regresara una respuesta.

El protocolo RARP permite que una estación de trabajo recién iniciada difunda su dirección Ethernet y diga “mi dirección Ethernet es xxx, sabe alguien mi dirección IP”. El servidor RARP ve esta solicitud, busca la dirección de Ethernet en sus archivos de configuración y envía de regreso la dirección IP correspondiente.

El uso de RARP es preferible a integrar una dirección IP en la imagen de memoria, pues permite usar la misma imagen para todas las máquinas. Si la dirección IP estuviera incluido en la imagen, cada estación de trabajo necesitaría su propia imagen.

Una desventaja del RARP es que usa una dirección de destino que contiene únicamente unos para llegar al servidor RARP. Sin embargo, tales difusiones no son reenviadas por los enrutadores, por lo que se requiere un servidor RARP en cada red. Para superar este problema, se ha inventado un protocolo alterno de arranque llamado BOOTP. A diferencia del RARP, BOOTP usa mensajes UDP, los cuales se reenvían a través de los enrutadores. Este protocolo también proporciona información adicional a una estación de trabajo sin disco, incluida la dirección IP del servidor de archivos que contiene la imagen de memoria, la dirección IP del enrutador predeterminado y la máscara de subred a usar.

ICMP (Internet Control Message Protocol, protocolo de control de mensajes de interred).

Protocolo usado por el IP para informar de errores y excepciones. El ICMP también incluye mensajes informativos usados por algunos programas como ping.

Se requiere del protocolo en una implementación normal de IP. Los dos protocolos son interdependientes. El IP emplea ICMP para enviar mensajes de error; el ICMP utiliza IP para transportar los mensajes.

El ICMP define cinco mensajes de error y cuatro informativos. Los cinco mensajes de error son:

- ✓ **Alivio de fuente (Source Quench).** Un enrutador envía un alivio de fuente cuando ha recibido tantos datagramas que ya no tiene más espacio en el búfer.
- ✓ **Tiempo excedido (Time Exceeded).** En dos casos se envía un mensaje de tiempo excedido. Cuando llega a cero el tiempo de vida de un datagrama, el enrutador lo descarta y transmite un mensaje de tiempo excedido.

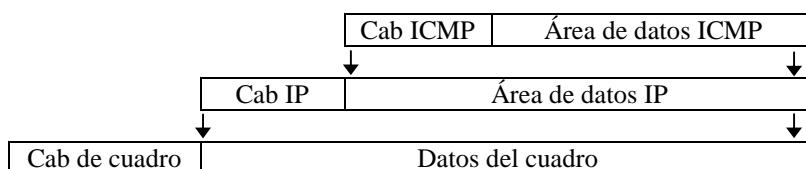
Además, envía un mensaje de tiempo excedido desde un host si el cronómetro expira antes de que lleguen todos los fragmentos de datagrama.

- ✓ **Destino inalcanzable (Destination Unreachable).** Cuando un enrutador determina que no puede entregarse un datagrama, envía un mensaje de destino inalcanzable al host que lo creó. El mensaje indica si el host o la red del destino es inalcanzable.
- ✓ **Redirigir (Redirect).** Cuando el host crea un datagrama destinado a una red remota, lo envía a un enrutador, que, a su vez, lo reenvía al destino. Si el enrutador determina que el host ha enviado una datagrama que debió expedir otro enrutador, utiliza un mensaje redirigir para obligar al host a cambiar su ruta.
- ✓ **Fragmentación requerida (Fragmentation Required).** Un host que crea un datagrama puede establecer un bit en la cabecera para indicar que no se permite la fragmentación. Si el enrutador encuentra que tal datagrama es mayor que la MTU de la red por la que se transmitirá, envía un mensaje de fragmentación requerida al transmisor y descarta el datagrama.

Además de los mensajes de error, el ICMP define cuatro mensajes informativos:

- ✓ **Solicitud / regreso de contestación (Echo Request/Reply).** Puede enviarse un mensaje de solicitud de respuesta al software ICMP de cualquier computadora. En respuesta a este mensaje, el ICMP debe enviar un mensaje de regreso de respuesta. La respuesta lleva los mismos datos que la solicitud.
- ✓ **Solicitud / regreso de máscara de dirección (Address Mask Request/Reply).** El host difunde una solicitud de máscara de dirección durante el proceso de arranque. Los enrutadores que reciben la solicitud regresan la máscara de dirección de la subred de 32 bits.

El ICMP usa IP para transportar los mensajes de error. Cuando un enrutador quiere enviar un mensaje, crea un datagrama IP y encapsula el mensaje ICMP en el datagrama. Es decir, el mensaje ICMP se pone en el área de datos del datagrama IP, que se reenvía como de costumbre, encapsulado en un cuadro. Se muestran los niveles de encapsulamiento:



Los mensajes ICMP siempre se producen como respuesta a un datagrama. O bien este se ha encontrado problemas, o bien lleva una solicitud ICMP a la que responde el enrutador. En ambos casos, el enrutador envía el mensaje ICMP de regreso a la fuente del datagrama.

El protocolo de mensajes de control incluye tanto mensajes sobre errores como mensajes informativos. El ICMP está integrado al IP: ICMP encapsula los mensajes en formato IP para transmitirlos e IP usa ICMP para reportar problemas.

• Protocolos de la capa de transporte: TCP, UDP (R, E)

TCP (Transmission Control Protocol, protocolo de control de transmisión).

Protocolo TCP/IP que proporciona a los programas de aplicación acceso al servicio de comunicación orientado a conexión. El TCP ofrece una entrega confiable y un flujo controlado. Más importante, el TCP se ajusta a las condiciones cambiantes de Internet adaptando un esquema de transmisión.

La confiabilidad es responsabilidad del protocolo de transportación: las aplicaciones interactúan con un servicio de transportación para enviar y recibir los datos. En el grupo TCP/IP, el protocolo de control de transmisión (TCP) ofrece un servicio de transportación confiable. En consecuencia, casi todas las aplicaciones de interred se diseñan para trabajar con TCP.

Desde el punto de vista de los programas de aplicación, el servicio que ofrece el TCP tiene siete características principales:

- ✓ **Orientación a conexión.** El TCP ofrece un servicio orientado a conexión en el que la aplicación debe solicitar primero una conexión al destino y luego emplearla para transferir datos.
- ✓ **Comunicación punto a punto.** Las conexiones TCP tienen dos puntos terminales.

- ✓ **Confiabilidad total.** El TCP garantiza que los datos enviados por una conexión se entregaran exactamente igual, sin datos faltantes ni desordenados.
- ✓ **Comunicación dúplex integral.** Las conexiones TCP admiten que los datos fluyan en ambos sentidos y que cualquiera de los programas de aplicación lo transmita en cualquier momento. El TCP puede manejar en búfer los datos de salida y de entrada, en ambos sentidos, lo cual permite que las aplicaciones envíen datos y luego reanuden el cómputo durante la transmisión.
- ✓ **Interfaz de flujo.** El TCP ofrece una interfaz de flujo en la que una aplicación transmite una secuencia continua de octetos por una conexión. Es decir, el TCP no maneja el concepto de registros, ni garantiza la entrega en pedazos del mismo tamaño de los que salieron del transmisor.
- ✓ **Arranque confiable de conexión.** El TCP requiere que, cuando se establece una conexión entre dos aplicaciones, ambas la acepten; los paquetes duplicados de conexiones previas no serán respuestas validas ni interferirán con la nueva conexión.

En suma: el TCP ofrece un servicio de transportación dúplex integral completamente confiable (sin duplicación ni perdidas), orientado a anexión, que permite que dos programas de aplicación inicien una conexión, transmitan en ambas direcciones y terminen la conexión. Cada conexión TCP inicia confiablemente y termina sin complicaciones, luego de entregar todos los datos.

Se dice que el TCP es un protocolo terminal a terminal porque ofrece una conexión directa entre la aplicación de la computadora local y otra remota. Las aplicaciones solicitan que el TCP establezca la conexión, transmita y reciba, y la cierre.

Las conexiones que ofrece el TCP se llaman conexiones virtuales porque se establecen en el software. El sistema de interred no ofrece apoyo de hardware o software a las conexiones, sino que son los módulos TCP de dos máquinas los que intercambian los mensajes y crean así la ilusión de una conexión.

El TCP se sirve del IP para llevar los mensajes. Cada mensaje TCP se encapsula en un datagrama IP y se transmite por la red. Cuando llega al host de destino, el IP pasa su contenido al TCP. Aun cuando el TCP aprovecha el IP para llevar los mensajes, este no los lee ni interpreta; por lo tanto, el TCP trata al IP como un sistema de comunicación de paquetes que conecta los host de los puntos terminales de la conexión. El IP trata los mensajes IP como datos por transferir.

El TCP emplea el mismo formato para todos los mensajes, incluyendo los datos, acuse de recibo y mensajes del acuerdo de tres vías para crear y terminar conexiones. En el TCP se usa el término segmento para referirse al mensaje.

0	4	10	16	24	31
PUERTO FUENTE			PUERTO DESTINO		
NÚMERO DE SECUENCIA					
NÚMERO DE ACUSE DE RECIBO					
LONG C	SIN USO	BITS DE CÓDIGO	VENTANA		
CIFRA DE COMPROBACIÓN			APUNTADOR URGENTE		
INICIO DE LOS DATOS			:		

Los campos *número de acuse de recibo* y *ventana* se refieren a datos de entrada: *número de acuse de recibo* indica el número de secuencia de los datos recibidos y *ventana* indica el espacio de búfer disponible para más datos. *Número de secuencia* se refiere a los datos de salida. Es el número de secuencia de los datos del segmento. El receptor lo utiliza para reordenar los segmentos que llegan en desorden y calcular un número de acuse de recibo. *Puerto destino* identifica el programa de aplicación del receptor que debe recibir los datos y *punto fuente* identifica el programa de aplicación que transmite los datos. Por último, el campo de *cifra de comprobación* cubre la cabecera y los datos del segmento TCP.

El TCP se diseña específicamente para proporcionar una corriente de bytes confiable a través de una interred no confiable. Una interred es diferente de una sola red porque las distintas partes pueden tener topologías, anchos de banda, retardos, tamaños de paquete y otros parámetros con grandes diferencias. Se diseña el TCP para adaptarse dinámicamente a las propiedades de la interred y para ser robusto ante muchos tipos de fallas.

Cada máquina que reconoce el TCP tiene una entidad de transporte TCP, ya sea un proceso de usuario o una parte del núcleo que maneja las corrientes TCP y tiene interfaz con la capa IP. Una entidad TCP acepta corrientes de datos de usuarios a los procesos locales, los divide en partes que no exceden 64 kb, y envía cada parte como datagrama IP independiente. Cuando llegan a una máquina datagramas IP que contienen datos TCP, son entregados a la entidad TCP, que reconstruye las corrientes originales de bytes.

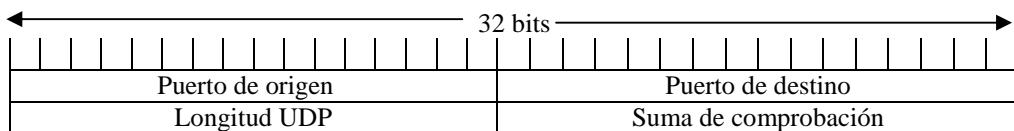
La capa IP no ofrece ninguna garantía de que los datagramas se entregaran adecuadamente, por lo que es responsabilidad del TC terminar de temporizar y retransmitirlos según se necesite. Los datagramas que si llegan pueden hacerlo desordenadamente; también es responsabilidad del TCP reensamblarlos en mensajes con la secuencia adecuada.

UDP (User Datagram Protocol, protocolo de datagrama de usuario).

Protocolo TCP/IP que proporciona a los programas de aplicación un servicio de comunicación sin conexiones.

Es un protocolo sin conexión, no confiable, para aplicaciones que no necesitan la asignación de secuencia ni el control de flujo del TCP y que desean utilizar los suyos propios. Este protocolo también se usa ampliamente para consultas de petición y respuesta de uso sola ocasión, del tipo cliente-servidor, y en aplicaciones en las que la entrega pronta es más importante que la entrega precisa, como las transmisiones de voz o video.

El UDP ofrece a las aplicaciones un mecanismo para enviar datagramas IP en bruto encapsulados sin tener que establecer una conexión. Un segmento UDP consiste en una cabecera de 8 bytes seguida de los datos. Los dos puertos sirven para identificar los puntos terminales de las máquinas de origen y destino. El campo de longitud UDP incluye la cabecera de 8 bytes y los datos. La suma de comprobación UDP incluye la pseudocabecera TCP, la cabecera UDP y los datos UDP, llenados a una cantidad par de bytes de ser necesario. Esta suma es opcional y se almacena como 0 si no se calcula.



3. Protocolos de alto nivel

• Modelo cliente-servidor (R, E, RP)

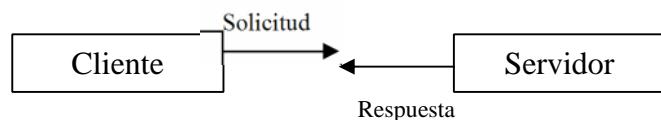
Es la interrelación entre dos programas cuando se comunican a través de una red. Un programa en una localidad envía una solicitud a otro programa en otra localidad y espera una respuesta. El programa que hace la solicitud se llama cliente, el que responde a la solicitud se llama servidor.

En el modelo cliente-servidor, la comunicación generalmente adopta la forma de un mensaje de solicitud del cliente al servidor pidiendo que se efectúe algún trabajo. A continuación el servidor hace el trabajo y devuelve la respuesta. Por lo regular, muchos clientes utilizan un número pequeño de servidores.

Es un concepto muy importante en las redes locales para aplicaciones que manejan grandes volúmenes de información. Son programas que dividen su trabajo en dos partes, una parte cliente que se realiza en el ordenador del usuario y otra parte servidor que se realiza en un servidor con dos fines:

- ✓ Aliviar la carga de trabajo del ordenador cliente.
- ✓ Reducir el tráfico de la red.

El cliente transmite una solicitud y el servidor transmite una respuesta:



• **Protocolos de control de la capa de aplicación: SMTP, SNMP, DNS, DHCP (R, E, RP)**

SMTP (Simple Mail Transfer Service, servicio sencillo de transferencia de correo).

Protocolo usado para transferencia de correo electrónico de una computadora a otra por medio de Internet. El SMTP es parte del grupo de programas de protocolo TCP/IP.

Cuando el programa de transferencia de correo contacta un servidor de una máquina remota, crea una conexión TCP por la que se comunica. Una vez establecida, ambos programas se valen del SMTP que permite al transmisor identificar un destinatario y transferir un mensaje de correo electrónico.

Aunque la transferencia de correo parece sencilla, el protocolo SMTP se encarga de muchos detalles; por ejemplo, necesita entrega confiable –el transmisor debe guardar una copia del mensaje hasta que el receptor lo haya almacenado en memoria no volátil–. Además, el SMTP permite al transmisor preguntar si existe un buzón de correo en la computadora del servidor.

En el Internet, el correo electrónico se entrega al hacer que la máquina de origen establezca una conexión TCP con el puerto 25 de máquina de destino. Escuchando en este puerto está un daemon de correo electrónico que habla con el SMTP. Este daemon acepta conexiones de entrada y copia mensajes de ellas a los buzones adecuados. Si no puede entregarse un mensaje, se devuelve al transmisor un informe de error que contiene la primera parte del mensaje que no pudo entregarse.

El SMTP es un protocolo ASCII sencillo. Tras establecer la conexión TCP con el puerto 25, la máquina transmisora, operando como cliente, espera que la máquina receptora, operando como servidor, hable primero. El servidor comienza por enviar una línea de texto que proporciona su identidad e indica si está preparado o no para recibir correo. Si no lo está, el cliente libera la conexión y lo intenta después.

Si el servidor está dispuesto a aceptar correo electrónico, el cliente anuncia de quién viene el mensaje, y a quién está dirigido. Si existe tal destinatario en el destino, el servidor da al cliente permiso para enviar el mensaje. Entonces el cliente envía el mensaje y el servidor acusa su recibo. Por lo general no se requieren sumas de comprobación porque el TCP proporciona una corriente de bits confiable. Si hay más correo electrónico, se envía ahora. Una vez que todo el correo electrónico ha sido intercambiado en ambas direcciones, se libera la conexión.

Aunque el protocolo SMTP está bien definido, pueden surgir algunos problemas. Uno se relaciona con la longitud de mensaje. Algunas implementaciones más viejas no pueden manejar mensajes mayores que 64 kb. Otro problema se relaciona con las terminaciones de temporización. Si el cliente y el servidor tienen terminaciones diferentes, uno de ellos puede terminar mientras el otro continua trabajando, terminando inesperadamente la conexión. Por último, en contadas ocasiones, pueden dispararse tormentas de correo infinitas. Para superar algunos de estos problemas, se ha definido el SMTP extendido (ESMTP).

SNMP (Simple Network Management Protocol, protocolo sencillo de administración de redes).

Protocolo que especifica la manera en que una estación de administración de red se comunica con el software agente de los dispositivos remotos, como enruteadores. El SNMP define el formato de los mensajes y su significado.

El protocolo estándar para la administración de una interred se llama SNMP. El protocolo SNMP define la comunicación de un administrador con un agente. Define el formato de las solicitudes que transmite el administrador a un agente y el formato de las respuestas del agente. Además, el SNMP define el significado de cada solicitud y respuesta posible. En particular, el SNMP indica que un mensaje SNMP se codifica empleando una norma conocida como notación de sintaxis abstracta.1 (ANS.1).

El protocolo SNMP no define un gran grupo de comandos, sino que usa el paradigma de obtención y almacenamiento en el que hay dos operaciones básicas: obtención, para obtener de un equipo u dato, y almacenamiento, para establecer un dato en un dispositivo. Cada objeto que puede obtenerse y almacenarse recibe un nombre único; el comando de una operación de obtención o almacenamiento debe especificar el nombre del objeto.

Cada objeto al que tiene acceso el SNMP debe definirse y recibir un nombre único. Además, el administrador y el agente deben acordar los nombres y significados de las operaciones de obtención y almacenamiento. En conjunto, el grupo de objetos a los que puede acceder el SNMP se conoce como base de información de administración (MIB).

De hecho, el SNMP no define una MIB. En cambio, la norma solo especifica el formato del mensaje y describe su

codificación; otra norma especifica las variables MIB y el significado de las operaciones de obtención y almacenamiento sobre cada una de ellas.

La forma normal de uso del SNMP es que la estación administradora envíe una solicitud a un agente pidiéndole información o mandándole actualizar su estado de cierta manera. Idealmente, la respuesta del agente simplemente es la información solicitada o la confirmación de que ha actualizado su estado según lo solicito. Los datos se envían empleando la sintaxis de transferencia del ANS.1.

El SNMP define siete mensajes que pueden enviarse. Se muestran los seis mensajes, el séptimo es el de respuesta:

Mensaje	Descripción
Get-request	Solicita el valor de una o más variables
Get-next-request	Solicita la variable que sigue a esta
Get-bulk-request	Obtiene una tabla grande
Set-request	Actualiza una o más variables
Inform-request	Mensaje de administrador a administrador describiendo la MIB local
SnmpV2-trap	Informe de interrupciones de agente a administrador

Los primeros tres mensajes solicitan devolución de valores de variables. El primer formato nombra explícitamente las variables que quiere. El segundo solicita la siguiente variable, permitiendo al administrador ir paso a paso a través de la MIB en orden alfabético. El tercero es para diferenciar transferencia grandes, como tablas. Después viene un mensaje que permite al administrador actualizar las variables de un agente, hasta el límite permitido por la especificación del objeto. Sigue una solicitud informal que permite a un administrador indicarle a otro las variables que está manejando. Por último, viene el mensaje enviado de un agente a un administrador cuando ha habido una interrupción.

DNS (Domain Name System, sistema de nombres de dominio).

Sistema automatizado que sirve para traducir nombres de computadoras a direcciones IP equivalentes. Un servidor DNS responde a una consulta buscando el nombre y devolviendo la dirección.

La esencia del DNS es la invención de un esquema de nombres jerárquico basado en dominio y una base de datos distribuida para implementar este esquema de nombres. El DNS se usa principalmente para relacionar las direcciones de host y destinos de correo electrónico con las direcciones IP, pero también puede usarse con otros fines.

El modo de usar DNS es el siguiente. Para relacionar un nombre con una dirección IP, un programa de aplicación llama a un procedimiento de biblioteca llamado resolvedor, pasándole el nombre como parámetro. El resolvedor envía un paquete UDP a un servidor DNS local, que entonces busca el nombre y devuelve la dirección IP al resolvedor, que entonces lo devuelve al solicitante. Con la dirección IP, el programa puede entonces establecer una conexión TCP con el destino, o enviarle paquetes UDP.

El esquema de nombres de Internet se llama sistema de nombres de dominio (DNS). En cuanto a la sintaxis, un nombre de computadora consta de una secuencia de segmentos separados por puntos.

Los nombres de dominio son jerárquicos; la parte más significativa del nombre está a la derecha. La parte de la izquierda del nombre es el nombre de una computadora. Los otros segmentos del nombre de dominio identifican el grupo al que pertenece el nombre. Aparte de especificar la manera de elegir los segmentos más significativos, el sistema de nombres de dominio no indica una cantidad exacta de segmentos en el nombre ni lo que representan. En cambio, cada organización escoge los segmentos para las computadoras que le pertenecen, así como lo que representan.

El sistema de nombres de dominio si especifica los datos del segmento más significativo, que se llama nivel superior del DNS:

Nombre de dominio	Asignado a
-------------------	------------

com	Organización comercial
edu	Institución educativa
gov	Organización gubernamental
mil	Grupo militar
net	Centro principal de soporte de redes
org	Organización diferente a las anteriores
arpa	Dominio ARPA temporal
int	Organización internacional
Código de país	País

Cuando una organización quiere participar en el sistema de nombres de dominio, debe solicitar un nombre con uno de los dominios de nivel superior. Casi todas las corporaciones eligen registrarse con el dominio com. Una vez asignado un dominio a una organización, se reserva el sufijo para ella, y ninguna otra recibirá el mismo sufijo de nombre. Además de la conocida estructura organizacional, el DNS permite que las organizaciones se valgan de un registro geográfico. Otros países han adoptado una combinación de nombres de dominio geográfico y organizacionales.

Una de las características principales del DNS es la autonomía, puesto que el sistema se diseña para permitir que las organizaciones controlen todos los nombres de sus computadoras sin informar a una autoridad central. La jerarquía de nombres colabora con la autonomía e tanto que permite que las organizaciones controlen todo los nombres con un sufijo exclusivo.

Además de los nombres jerárquicos, el DNS sirve de interacción cliente-servidor para fomentar la autonomía. En esencia, el DNS opera como una gran base de datos distribuida. Casi todas las organizaciones con conexión a Internet operan un servidor de nombres de dominio. Cada servidor tiene información que liga al servidor con otros servidores de nombre de dominio; el grupo de servidores resultante funciona como una gran base coordinada de datos de nombres.

Cuando una aplicación necesita traducir un nombre a dirección IP, la aplicación se vuelve cliente del sistema de nombres. El cliente pone el nombre que quiere traducir en un mensaje de solicitud DNS y lo transmite a un servidor DNS. El servidor extrae de la solicitud el nombre, lo traduce a una dirección IP equivalente que regresa a la aplicación en un mensaje de respuesta.

La traducción de un nombre de dominio a su dirección IP equivalente se llama resolución de nombre, y se dice que el nombre se resuelve a una dirección. El software para llevar a cabo la traducción se llama resolutor de nombres.

Cada resolutor se configura con la dirección de un servidor local de nombres de dominio. Para volverse un cliente del servidor DNS, el resolutor pone el nombre especificando en un mensaje de solicitud DNS y lo transmite al servidor local. El resolutor espera entonces que el servidor mande un mensaje de respuesta DNS.

Hay dos optimizaciones primarias para el DNS: replica y caché. Hay muchas replicas de los servidores raíz en todas partes del mundo. Cuando un sitio nuevo se une a Internet, configura su servidor DNS local con una lista de servidores raíz. El servidor del sitio usa el servidor con mayor capacidad de respuesta en el momento. En la práctica, el servidor más cercano es el que responde mejor.

La caché DNS es más importante que la replica porque afecta a casi todos los sistemas. Todos los servidores guardan un caché de nombres. Al buscar uno nuevo, pone una copia de la ligadura en la caché. Antes de contactar con otro servidor para solicitar una ligadura, se hace una búsqueda en la caché. Si ahí está la respuesta, se usa la información en caché para generar la contestación.

DHCP (Dynamic Host Configuration Protocol, protocolo de configuración dinámica de host).

Protocolo que usan las computadoras para obtener información de configuración. El DHCP permite asignar una dirección IP a una computadora sin requerir que un administrador configure la información sobre la computadora en la base de datos de un servidor.

Para automatizar la configuración, el IETF diseña el protocolo dinámico de configuración de hosts (DHCP). El DHCP no necesita que un administrador añada entradas correspondientes a todas las computadoras de la base de datos empleada por un servidor. En su lugar, ofrece un mecanismo que permite a una computadora unirse a una red nueva y obtener una dirección IP sin intervención manual. El concepto se ha denominado conectividad plug-and-

play. Además, el DHCP se ajusta a las computadoras que ejecutan software servidor así como a las que ejecutan software cliente:

- ✓ Cuando se mueve una computadora que ejecuta software cliente a una red nueva, puede valerse del DHCP para obtener la información de configuración sin intervención manual.
- ✓ El DHCP permite que a las computadoras no móviles que ejecutan software servidor sean asignadas direcciones IP permanentes; las direcciones no cambiaran al rearranque de las computadoras.

Para ajustarse a ambos tipos de computadoras, el DHCP no puede valerse de un esquema de licitación; en cambio, emplea un enfoque cliente-servidor. Cuando arranca la computadora difunde una solicitud DHCP a la que responde un servidor con una respuesta DHCP. El administrador puede configurar el servidor para que tenga dos tipos de direcciones: direcciones permanentes asignadas a computadoras servidoras y direcciones para ser asignadas a solicitud. Cuando arranca una computadora y transmite una solicitud al DHCP, el servidor consulta su base de datos para encontrar la información de configuración. Si la base de datos tiene una entrada específica para la computadora, regresa la correspondiente a ella. De no encontrarla, selecciona la siguiente dirección IP del agrupo y la asigna.

De hecho, las direcciones asignadas a solicitud no son permanentes. Más bien, el DHCP emite un arrendamiento para la dirección durante un periodo finito. Al expirar el arrendamiento, la computadora debe renegociar con el DHCP una extensión. Normalmente el DHCP la aprobará; sin embargo, un sitio puede seleccionar una política administrativa que niegue la extensión. Si el DHCP rechaza la solicitud de extensión, la computadora debe dejar de emplear la dirección.

Si las computadoras de una red utilizan DHCP para obtener información de configuración durante el arranque, un suceso que provoque el arranque simultáneo de todas ellas puede provocar que la red y el servidor se inunde de solicitudes. Para evitar el problema, el DHCP emplea la técnica: cada computadora espera un tiempo aleatorio antes de transmitir o retransmitir una solicitud.

El protocolo DHCP tiene dos pasos: uno en el que la computadora difunde un mensaje de descubrimiento para encontrar un servidor DHCP y otro en el que la computadora selecciona uno de los servidores que respondió con su mensaje y manda una solicitud al servidor. Para evitar que la computadora repita ambos pasos en cada arranque y cada vez que necesita ampliar el arrendamiento, el DHCP emplea caché. Cuando una computadora descubre el servidor DHCP, guarda su dirección en una caché en disco. De igual modo, una vez obtenida una dirección IP, la guarda en caché. Esta acción ahorra tiempo y reduce el tráfico en la red.

Formato de mensaje DHCP:

0	8	16	24	31
OP	TIPOH	LONGH	SALTOS	
IDENTIFICADOR DE TRANSACCIÓN				
SEGUNDOS TRANSCURRIDOS		BANDERAS		
DIRECCIÓN IP CLIENTE				
SU DIRECCIÓN IP				
DIRECCIÓN IP SERVIDOR				
DIRECCIÓN IP ENRUTADOR				
DIRECCIÓN HARDWARE CLIENTE (16 OCTETOS)				
:				
NOMBRE HOST SERVIDOR (64 OCTETOS)				
:				
NOMBRE ARCHIVO ARRANQUE (128 OCTETOS)				
:				
OPCIONES (VARIABLE)				
:				

Aunque el DHCP permite que las computadoras obtengan sus direcciones IP sin intervención manual, no interactúa con el sistema de nombres de dominio. Como resultado, una computadora no puede conservar su nombre al cambiar de dirección. Cabe notar, que la computadora no necesita pasar a una red nueva para que cambie su nombre. Hasta que se formule un protocolo estándar, los sitios que emplean DHCP deben valerse de un mecanismo

no estándar para cambiar la base de datos DNS o estar preparados para que los nombres de computadoras cambien con las modificaciones de dirección.

- **Protocolos de aplicación para transferencia de archivos: FTP, TFTP, http (R, E, RP)**

FTP (File Transfer Protocol, protocolo de transferencia de archivos).

El servicio de Internet de transferencia de archivos más difundido es el FTP. El FTP permite la transferencia de archivos arbitrarios e incluye un mecanismo que controla la propiedad de los archivos y las restricciones de acceso. Del mismo modo, dado que esconde los detalles de los sistemas de computo, el FTP se ajusta bien a la heterogeneidad – puede emplearse para transferir copias de los archivos entre pares arbitrarios de computadoras–.

El FTP es una de las aplicaciones más viejas que todavía se emplean en Internet. Originalmente definido como parte de los protocolos ARPANET, el FTP se encuentra tanto en el TCP como el IP. Al aparecer TCP/IP, desarrollo una versión nueva del FTP que funciona como los nuevos protocolos de Internet.

El servicio de transferencia de archivos de mayor uso de Internet emplea el FTP. El FTP es un protocolo de propósito general para copiar archivos arbitrarios de una computadora a otra. El FTP requiere que un usuario establezca una conexión de control con una computadora remota antes de poder transferir archivos. Para obtener autorización, el usuario debe dar una clave de acceso y una contraseña. La conexión de control permanece hasta que el usuario decida cerrarla.

Aunque una clave de acceso y una contraseña pueden ayudar a evitar el acceso no autorizado a los archivos, tal autorización también puede ser inconveniente. Para que cualquier usuario acceda a un archivo, muchas instalaciones siguen la convención de establecer una cuenta especial exclusiva del FTP. La cuenta, que tiene la clave de acceso anonymous, otorga a cualquier usuario un acceso mínimo a los archivos. Se emplea el término FTP anónimo para describir el proceso de acceso con la clave anonymous.

El FTP permite la transferencia de archivos en ambas direcciones. Tras establecer una conexión con una computadora remota, el usuario puede traer copias de archivos remotos o transferir copias de archivos locales a la máquina remota. Por supuesto, tales transferencias estás sujetas a los permisos de acceso.

El FTP tiene dos modos de transferencia básico: textual y binario. El usuario debe seleccionar una clase de transferencia, con lo que queda en efecto el modo durante toda la transferencia de archivos.

La transferencia textual es la empleada para los archivos de texto básicos. Un archivo de texto contiene una secuencia de caracteres divididos en líneas. Casi todos los sistemas de computo emplean el juego de caracteres ASCII o EBCDIC para representar los caracteres de un archivo de texto. La única alternativa a la transferencia de texto en el FTP es la transferencia binaria, que debe utilizarse con todos los archivos que no sean de texto.

El FTP no interpreta el contenido de un archivo transferido en modo binario ni traduce entre una representación y otra; más bien, la transferencia binaria sencillamente hace una copia.

El protocolo FTP se usa para acceder a archivos mediante FTP, el protocolo de transferencia de archivos de Internet. Numerosos servidores FTP de todo el mundo permiten que gente de cualquier lugar de Internet establezca una sesión y descargue los archivos colocados en el servidor FTP.

TFTP (Trivial File Transfer Protocol, protocolo trivial de transferencia de archivos).

Protocolo usado para transferir un archivo de una computadora a otra. El TFTP es más sencillo que el FTP, pero no tiene tanta capacidad.

El TFTP es diferente del FTP en varios sentidos. Primero, la comunicación entre cliente y servidor TFTP emplea UDP en lugar de TCP. Segundo, el TFTP solo maneja transferencia de archivos, esto es, el TFTP no maneja interacción ni tiene un grupo grande de comandos. Además, el TFTP no permite al usuario listar el contenido de un directorio ni interrogar al servidor para determinar los nombres de los archivos disponibles. Tercero, el TFTP no tiene autorizaciones. El cliente no manda una clave de acceso ni una contraseña; los archivos solo pueden transferirse si tienen permisos que permitan el acceso global.

Aunque es menos poderoso que el FTP, el TFTP tiene dos ventajas. Primera, puede utilizarse en medios en los que esta disponible el UDP pero no el TCP. Segunda, el TFTP necesita menos que el código FTP. Aunque tales ventajas no son de importancia en una computadora de propósito general, puede serlo en una pequeña o en un dispositivo de hardware de propósito general.

El TFTP es muy útil para efectuar el proceso de arranque de un dispositivo sin disco para almacenar el software del sistema. Todo lo que necesita el dispositivo es una conexión de red y una pequeña cantidad de memoria de solo lectura (ROM) en las que estén pregrabados el TFTP, el UDP y el IP. Al energizarse, el dispositivo ejecuta el código de la ROM, que difunde una solicitud TFTP por la red. El servidor TFTP de la red se configura para responder a la solicitud transmitiendo un archivo con el programa que se ejecutara. El dispositivo recibe el archivo, la carga en memoria e inicia la ejecución del programa.

HTTP (HyperText Transport Protocol, protocolo de transportación de hipertexto).

Protocolo usado para transportar una pagina de la WWW de una computadora a otra.

Cuando un visualizador interactúa con un servidor de Web, ambos programas emplean el http. En principio, el http es directo: permite que el visualizador solicite un elemento específico, que entonces regresa el servidor. Para que los visualizadores y servidores puedan operar sin ambigüedades, el http define el formato de las solicitudes del visualizador al servidor, así como el formato de las respuestas del servidor.

El protocolo de transferencia de hipertextos (http) es el protocolo base del World Wide Web (WWW) y se puede utilizar en cualquier aplicación cliente-servidor que suponga la utilización de hipertextos. El nombre es más bien confuso ya que http no es un protocolo para transferir hipertexto; en lugar de eso es un protocolo para transmitir información con la eficiencia necesaria para hacer que el hipertexto salte. Los datos transferidos por el protocolo pueden ser texto propiamente dicho, hipertexto, audio, imágenes o cualquier información accesible a través de Internet.

Http es un protocolo cliente-servidor orientado a transacciones. El uso más común de http es entre un navegador (browser) web y un servidor web. Para proporcionar seguridad, http hace uso de TCP. Sin embargo, http es un protocolo sin estados: cada transacción se trata independientemente. Por consiguiente, una implementación típica creará una conexión nueva entre el cliente y el servidor con cada transacción y después la cierra tan pronto como se completa la transacción, aunque la especificación no impone esta relación uno-a-uno entre la transacción y los tiempos de vida de la conexión.

Otra característica importante de http es que es flexible en cuanto a los formatos que puede tratar. Cuando un cliente emite una solicitud a un servidor, puede incluir una lista de prioridades de formatos con lo que puede operar, y el servidor responde con el formato adecuado.

Se definen tres tipos de sistemas intermedios en la especificación http: representante (proxy), pasarela (gateway), y túnel (tunnel).

- ✓ **Representante.** Un representante actúa en nombre del cliente y presenta solicitudes de otros clientes a un servidor. El representante actúa como un servidor cuando interactúa con un cliente y como cliente cuando interactúa con un servidor. Existen varios escenarios que requieren el uso de un representante:
 1. Intermediario de seguridad: el cliente y el servidor pueden estar separados por un intermediario de seguridad, como es el caso de los cortafuegos, con el representante en el lado del cliente del cortafuegos. Normalmente, el cliente forma parte de una red protegida por un cortafuegos y el servidor es externo a esta red protegida. En este caso, el servidor se debe autenticar al cortafuegos para establecer una conexión con el representante. El representante acepta respuestas después de que hayan pasado por el cortafuegos.
 2. Diferentes versiones de http: si el cliente y el servidor ejecutan diferentes versiones de http, el representante puede implementar ambas versiones y realizar las traducciones requeridas.
- ✓ **Pasarela.** Es un servidor que respecto al cliente actúa como si fuera un servidor original. Actúa en nombre de otros servidores que no son capaces de comunicarse directamente con un cliente. Existen varios escenarios en los que pueden utilizar pasarelas:
 1. Intermediario de seguridad: el cliente y el servidor pueden estar separados por un intermediario de seguridad, como es el caso de los cortafuegos, con la pasarela en el lado del servidor del cortafuegos. Normalmente, el servidor está conectado a la red protegida por un cortafuegos y el cliente es externo a esta red protegida. En este caso, el cliente se debe autenticar a la pasarela, que puede pasar la solicitud al servidor.

2. **Servidor no-HTTP:** los clientes web se han construido con la capacidad de contactar con servidores de otros protocolos aparte de HTTP, como servidores FTP o Gopher. Esta capacidad también la puede proporcionar la pasarela. El servidor pasarela contacta con el servidor relevante para obtener el resultado deseado. Este resultado se convierte entonces en una forma adecuada para HTTP y se transmite de vuelta al cliente.
- ✓ **Túnel.** A diferencia del representante y la pasarela, el túnel no realiza operaciones con las solicitudes y respuestas HTTP. Así es que un túnel simplemente es un punto de retransmisión entre dos conexiones TCP, y los agentes HTTP se transfieren sin cambiarlos como si hubiera una única conexión HTTP entre el agente usuario y el servidor origen. Los túneles se utilizan cuando debe haber un sistema intermediario entre el cliente y el servidor pero no es necesario para ese sistema comprender el contenido de los mensajes.

El protocolo estándar de transferencia de la web es el HTTP. Cada interacción consiste en una solicitud ASCII seguida de una respuesta tipo MIME RFC 822. Aunque es muy común el uso del TCP para la conexión de transporte, no es muy requerido formalmente por el estándar.

El protocolo HTTP consiste en dos elementos bastante diferentes: el grupo de solicitudes de los visualizadores a los servidores y el grupo de respuestas en el otro sentido.

Todas las versiones más recientes de HTTP reconocen dos tipos de solicitud: solicitudes sencillas y solicitudes completas. Una solicitud sencilla es solo una línea GET que nombra la página deseada, sin la versión del protocolo. La respuesta es la página en bruto, sin cabeceras, sin MIME y sin codificación. Las solicitudes completas se indican por la presencia de la versión del protocolo en la línea de solicitud GET. Las solicitudes pueden consistir en múltiples líneas seguidas de una línea en blanco para indicar el final de la solicitud. La primera línea de una solicitud completa contiene el comando, la página deseada y el protocolo / versión. Las líneas subsiguientes contienen cabeceras RFC 822.

Aunque el HTTP se diseñó para usarse en la Web, ha sido intencionalmente más general de lo necesario con miras a aplicaciones futuras orientadas a objetos. Por esta razón, la primera palabra de la línea de solicitud completa es simplemente el nombre del método (comando) a ejecutar con la página de la Web (u objeto general). Los métodos interconstruidos se listan a continuación:

Método	Descripción
GET	Solicita leer una página de Web
HEAD	Solicita leer la cabecera de una página de Web
PUT	Solicita almacenar una página de Web
POST	Adiciona a un recurso nombrado
DELETE	Elimina la página de Web
LINK	Conecta dos recursos existentes
UNLINK	Rompe una conexión entre dos recursos

Después de acceder a objetos generales, también pueden estar disponibles métodos adicionales específicos para ese objeto. Los nombres son sensibles a mayúsculas y minúsculas, por lo que GET es un método legal, pero get no.

Cada solicitud recibe una respuesta que consiste en la línea de estado, y posiblemente, información adicional. La línea de estado puede tener 200 (OK) o cualquiera de varios códigos de error, por ejemplo 304 (no modificado), 400 (lista errónea) o 402 (prohibido).

• Protocolos de aplicación de emulación de terminal: TELNET, terminal virtual (R, E, RP)

Telnet

Telnet facilita la posibilidad de conexión remota, mediante la cual el usuario en una terminal o computador personal se conecta a un computador remoto y trabaja como si estuviera conectado directamente a ese computador. Este protocolo se diseñó para trabajar con terminales poco sofisticados en modo scroll (avance de pantalla). En realidad, Telnet se implementa en dos módulos: el usuario Telnet interactúa con el módulo de E/S para comunicarse con terminal local. Esto convierte las particularidades de los terminales reales en una definición normalizada de terminal de red, y viceversa. El servidor Telnet interactúa con la aplicación, actuando como sustituto del gestor del terminal, para que de esta forma el terminal remoto le parezca local a la aplicación. El tráfico entre el terminal del usuario y el servidor Telnet se transmite sobre una conexión TCP.

El protocolo Telnet sirve para establecer una conexión en línea con una máquina remota. Se usa de la misma manera que el programa Telnet, puesto que la mayoría de los visualizadores simplemente llaman al programa Telnet como aplicación ayudante.

Terminal virtual.

Existen cientos de tipos de terminales incompatibles en el mundo. Considere la situación de un editor de pantalla completa que debe trabajar en una red con muchos tipos diferentes de terminal, cada uno con formatos diferentes de pantalla, secuencias de escape para insertar y eliminar texto, mover cursor, etc.

Una forma de resolver este problema es definir una terminal virtual de red abstracta que los editores y otros programas puedan manejar. Para cada tipo de terminal, se debe escribir un programa para establecer la correspondencia entre las funciones de la terminal virtual de red y las de la terminal real. Este software debe emitir la secuencia apropiada de órdenes a la terminal real. Todo el software de terminal virtual está en la capa de aplicación.

C. Intercomunicación de redes

I. Interconectividad

1. Teoría de interconexiones

• Clases de direcciones IP (R, E, RP)

Dentro del protocolo TCP/IP el direccionamiento lo especifica el protocolo IP. Este consiste en que cada host recibe un número único de 32 bits. Cada paquete enviado por la interred contiene la dirección IP del transmisor y la del receptor. Por lo tanto, para transmitir información por una interred TCP/IP, la computadora debe conocer la dirección IP de la computadora remota a la que se destinan los datos. Cada dirección IP se divide en 2 partes:

- ✓ **Prefijo:** Identifica la red física a la que está conectada la computadora.
- ✓ **Sufijo:** Identifica las computadoras de esa red.

A cada red física de la interred se asigna una cifra única conocida como número de red, que aparece como prefijo en la dirección de cada computadora conectada a la red. Además, a cada computadora de la red se asigna un sufijo de dirección única.

Debido a que las interredes pueden incluir varias tecnologías de red, es posible construir una interred a partir de unas cuantas redes grandes y otra que conste de muchas redes pequeñas. Además una interred puede contener tanto redes grandes como pequeñas. Para lograrlo se diseñó un esquema de direccionamiento que divide el espacio de direcciones IP en tres clases primarias en las que el prefijo y el sufijo tienen diferentes tamaños.

Los primeros 4 bits determinan la clase a la que pertenece la dirección e indican cómo se divide el resto en prefijo y sufijo.

	0	1	2	3	8	16	24	31
A	0	Prefijo						
B	1	0	Prefijo					
C	1	1	0	Prefijo				
D	1	1	1	0	Dirección Multidestino			
E	1	1	1	1	Reservada para Uso Futuro			

Las clases A, B y C se llaman clases primarias porque se usan para direcciones de host. La clase D se utiliza para multitransmisión, lo que permite la entrega a un grupo de computadoras. Para usar multitransmisión IP, un grupo de hosts debe acordar compartir una dirección multitransmisión. Una vez establecido el grupo multitransmisión, se entregan a los hosts del grupo copias de los paquetes enviados a esta dirección.

• Protocolos para resolución de direcciones: ARP y RARP (R, E)

- ✓ **ARP (Protocolo de Resolución de Direcciones):** Las direcciones IP sirven para identificar cada computadora de una red, pero éstas no pueden usarse para enviar paquetes porque el hardware de la capa de enlace no entiende de direcciones de Internet. La mayoría de los hosts están conectados a alguna LAN mediante una tarjeta de interfaz que solo entiende de direcciones de LAN.

El ARP es un software para traducir la dirección de protocolo a una dirección de hardware equivalente. ARP define 2 tipos básicos de mensaje: solicitud y respuesta. Un mensaje de solicitud tiene una dirección IP y solicita la dirección de hardware correspondiente; una respuesta tiene tanto la dirección IP enviada con la solicitud como la dirección de hardware.

- ✓ **RARP (Protocolo de Resolución de Direcciones en Reversa):** En ocasiones se ocupa que dada una dirección Ethernet se obtenga la dirección IP correspondiente. El software RARP permite que una estación de trabajo difunda su dirección Ethernet., el servidor RARP ve la solicitud, busca la dirección Ethernet en sus archivos de configuración y envía de regreso la dirección IP correspondiente.

• Subredes y máscaras (R, E, RP)

Como todos los hosts de una red deben tener el mismo número de red se pueden causar problemas a medida que crecen las redes. Para evitar este problema la solución es permitir la división de una red en varias partes para uso interno, pero aún actuar como una sola red ante el mundo exterior. A estas partes se les llaman subredes.

Al introducirse subredes, se cambian las tablas de enrutamiento, agregando entradas con forma de [red, subred, 0] y [red, subred, host]. Por tanto un enrutador de la subred sabe cómo llegar a todas las demás subredes y también cómo llegar a todos los hosts de dicha subred, no tiene que saber los detalles sobre los hosts de otras subredes. Cada enrutador debe hacer un AND booleano con la máscara de subred [255.255.255.0] para deshacer el número de host y buscar la dirección resultante en sus tablas (tras determinar la clase de red de la que se trata).

• DNS (R, E, RP)

DNS.

Es la invención de un esquema de nombres jerárquico basado en dominio y una base de datos distribuida para implementar este esquema de nombres. El DNS se usa principalmente para relacionar las direcciones de host y destinos de e-mail con las direcciones IP.

Para relacionar un nombre con una dirección IP, un programa de aplicación llama a un procedimiento llamado revolvedor pasándole el nombre como parámetro. El revolvedor envía un paquete UDP a un servidor DNS local, que entonces busca el nombre y devuelve la dirección IP al revolvedor, que entonces lo devuelve al solicitante. Con la dirección IP, el programa puede entonces establecer una conexión TCP con el destino, o enviarle paquetes UDP.

Un nombre de computadora consta de una secuencia de segmentos alfanuméricos separados por puntos. Los nombres de dominio son jerárquicos; la parte más significativa del nombre está a la derecha. La parte de la izquierda del nombre es el nombre de una computadora. Los otros segmentos del nombre de dominio identifican al grupo al que pertenece el nombre.

DNS solo especifica los datos del segmento más significativo que se llama nivel superior, la cantidad de segmentos en el nombre y lo que representan los escoge la organización. Los posibles dominios de nivel superior son:

Com	Organización comercial.
Edu	Institución educativa.
Gov	Organización gubernamental.
Mil	Grupo militar.
Net	Centro principal de soporte de redes.
Org	Organización diferente a las anteriores.
Arpa	Dominio ARPA temporal.
Int	Organización internacional.
código del país	País

DNS obedece al modelo cliente/servidor, ya que opera como una gran base de datos distribuida. Casi todas las organizaciones con conexiones Internet operan un servidor de nombres de dominio. Cada servidor tiene información que liga al servidor con otros servidores de nombre de dominio; el grupo de servidores resultante funciona como una gran base de datos de nombres.

Cuando una aplicación necesita traducir un nombre a dirección IP, la aplicación se vuelve cliente del sistema de nombres. El cliente pone el nombre que quiere traducir en un mensaje de solicitud DNS y lo transmite a un servidor DNS. El servidor extrae de la solicitud el nombre, lo traduce a una dirección IP equivalente que regresa a la aplicación en un mensaje de respuesta. Hay 2 optimizaciones primarias para el DNS: réplica y caché.

Existen muchas réplicas de los servidores raíz en el mundo. Cuando un sitio nuevo se une a Internet, configura su servidor DNS local con una lista de servidores raíz y se utiliza el servidor con mayor capacidad de respuesta en el momento, el cual por lo general termina siendo el más cercano.

La caché DNS (que afecta a casi todos los sistemas) al buscar un servidor nuevo pone una copia de la ligadura en la caché. Antes de contactar con otro servidor para solicitar una liga, se hace una búsqueda en la caché. Si esta se encuentra en la caché, se usa la información en caché para generar la contestación.

- **Llamadas a procedimientos remotos (R, E)**

- ✓ **FTP:** Permite la transferencia de archivos arbitrarios e incluye un mecanismo que controla la propiedad de los archivos y las restricciones de acceso.

La mayoría de los usuarios llaman interactivamente al FTP, ejecuta un cliente FTP que establece una comunicación con un servidor y transfiere archivos. Algunos sistemas llaman al FTP automáticamente si que el usuario interactúe con el cliente FTP. Cuando un usuario llama interactivamente al FTP, se comunica con una interfaz manejada por comandos. El FTP presenta una indicación (prompt) a la que responde el usuario ingresando un comando. El FTP lo ejecuta y presenta otra indicación.

El FTP tiene comandos que permiten al usuario especificar la computadora remota, dar autorización, determinar los archivos remotos disponibles y solicitar la transferencia de archivos. Algunos comandos FTP se ejecutan inmediatamente, mientras que otros pueden tardar bastante.

Se requiere que el usuario del nombre de dominio de una computadora remota, a fin de establecer una conexión TCP con ella. Dicha conexión (llamada conexión de control) sirve para enviar comandos. Una vez abierta una conexión, el FTP solicita al usuario la información de autorización a la computadora remota; para darla, éste debe proporcionar una clave de acceso y una contraseña. La clave de acceso, que debe corresponder a una cuenta válida de la computadora remota, determina los archivos que se accederán.

FTP permite también el acceso limitado a los archivos a varios usuarios no registrados bajo la clave de acceso *anonymus*.

FTP permite la transferencia de archivos en ambas direcciones, o sea que tras establecer una conexión con una computadora remota, el usuario puede traer copias de archivos remotos o transferir copias de archivos locales a la máquina remota, todo esto sujeto a los archivos de acceso. Cuando la transferencia finaliza, se utiliza la conexión de control para indicar el fin, además esta misma conexión estará disponible para aceptar nuevas órdenes de transferencia.

Los archivos que pueden ser enviados a través de FTP son de dos tipos: archivos de texto y archivos de no texto (todos los demás tipos de archivos).

- ✓ **TELNET:** El usuario puede conectarse a través de una computadora personal a una computadora remota y trabaja como si estuviera conectado directamente a esa computadora remota. El usuario TELNET interactúa con el módulo de E/S para comunicarse con terminal local. Este traduce lo efectuado por el usuario para enviarlo por la red. El servidor TELNET interactúa con la aplicación, actuando como un sustituto del gestor terminal, para que de esta forma el terminal remoto le parezca local a la aplicación.
- ✓ **TFTP:** Utiliza para la comunicación cliente/servidor UDP en lugar de TCP. Solo maneja la transferencia de archivos y no tiene un grupo grande de comandos. El cliente no manda una clave de acceso ni una contraseña, los archivos solo pueden transferirse si tienen permisos de acceso global. Es utilizado en medios en los que está disponible el UDP pero no el TCP y en computadores con hardware de propósito especial.
- ✓ **NFS:** Permite a una aplicación abrir un archivo remoto, moverse a una posición específica del archivo y leer y escribir a partir de ahí.

Un archivo de un servidor NFS puede ser accedido por varios clientes, mientras que un cliente modifica el archivo, NFS permite el bloqueo de este, para luego de ser modificado se libera y se permite el acceso a este archivo por otros clientes.

NFS se integra al sistema de archivos de la computadora. La integración permite que un programa de aplicación arbitrario opere sobre un archivo remoto: cuando una aplicación ejecuta una operación de archivo, el cliente NFS se comunica con la computadora remota y ejecuta ahí la operación.

• Programación con Sockets (R)

El paradigma de e/s de unix y la e/s de la red

En primer lugar hemos de distinguir entre los protocolos de interface y el TCP/IP, debido a que los estándares no especifican exactamente cómo es que interactúan los programas de aplicación con el software de protocolo. A pesar de la carencia de un estándar, veremos la interface del UNIX BSD como se emplea el TCP/IP en programación. En particular, la interface *Winsock* proporciona la funcionalidad socket para Ms Windows.

Unix fue desarrollado y diseñado como un sistema operativo de tiempo compartido para computadoras uniprocesador. Se trata, como ya es sabido, de un S.O. orientado a proceso, en el que cada programa de aplicación se ejecuta como un proceso de nivel de usuario. Derivados de los MULTICS, los primitivos sistemas de E/S de UNIX siguen un paradigma conocido como “*Open-Read-Write-Close*”: antes de que un proceso de usuario pueda ejecutar operaciones de E/S, llama a *Open* para especificar el archivo o dispositivo que se va a utilizar (recuérdese la independencia de dispositivo de UNIX) y obtiene el permiso. La llamada a *Open* devuelve un pequeño entero (el descriptor de archivo) que el proceso utiliza al ejecutar las operaciones de E/S en el archivo abierto. Una vez abierto un objeto, se pueden hacer las llamadas a *Read* y/o *Write*. Tanto *Read* como *Write* toman tres argumentos (descriptor de archivo, dirección del buffer y nº de bytes a transferir). Una vez completadas estas operaciones el proceso llama a *Close*.

Originalmente, todas las operaciones UNIX se agrupaban como se ha descrito anteriormente, y una de las primeras implementaciones de TCP/IP también utilizó este paradigma. Pero el grupo que añadió los protocolos TCP/IP al BSD decidió que, como los protocolos de red eran más complejos que los dispositivos convencionales de E/S, la interacción entre los programas de usuario y los protocolos de red debía ser más compleja. En particular, la interface de protocolo debía permitir a los programadores crear un código de servidor que esperaba las conexiones pasivamente, así como también un código cliente que formara activamente las conexiones. Para manejar datagramas, se decidió abandonar este paradigma.

La abstracción de socket

La base para la E/S de red en UNIX se centra en una abstracción conocida como *socket*.

El socket es la generalización del mecanismo de acceso a archivos de UNIX que proporciona un punto final para la comunicación. Al igual que con el acceso a archivos, los programas de aplicación requieren que el S.O. cree un socket cuando se necesite. El S.O. devuelve un entero que el programa de aplicación utiliza para hacer referencia al socket recientemente creado. La diferencia principal entre los descriptores de archivo y los descriptores de socket es que el sistema operativo enlaza un descriptor de archivo a un archivo o dispositivo del sistema cuando la aplicación llama a *Open*, pero puede crear sockets sin enlazarlos a direcciones de destino específicas.

Básicamente, el socket es una API en la que el servidor espera en un puerto predefinido y el cliente puede utilizar sin embargo un puerto dinámico.

Ejemplos:

- ✓ Creación de un socket:

resultado = socket (pf, tipo, protocolo)

El argumento PF especifica la familia de protocolo que se va utilizar con el socket (v.q. PF_INET para TCP/IP). El argumento tipo especifica el tipo de comunicación que se desea (v.q.SOCK_DGRAM para servicio de entrega de datagramas sin conexión, o SOCK_STREAM para servicio de entrega confiable de flujo).

- ✓ Envío de datos:

write (socket, buffer, length)

- ✓ Especificación de una dirección local:

bind (socket, localaddr, addrlen)

Inicialmente, un socket se crea sin ninguna asociación hacia direcciones locales o de destino. Para los protocolos TCP/IP, esto significa que ningún número de puerto de protocolo local se ha asignado y que ningún puerto de destino o dirección IP se ha especificado. En muchos casos, los programas de aplicación no se preocupan por las direcciones locales que utilizan, ni están dispuestos a permitir que el software de protocolo elija una para ellos. Sin embargo, los procesos del servidor que operan en un puerto “bien conocido” deben ser capaces de especificar dicho puerto para el sistema. Una vez que se ha creado un socket, el servidor utiliza una llamada del sistema BIND (enlace) para establecer una dirección local para ello.

BIND tiene la forma que se ha descrito arriba.

2. Teoría de enrutamiento

• **Spanning Tree y enrutamiento de origen (R, E)**

Los protocolos de encaminamiento permiten a los routers intercambiar información acerca del estado de la red y acelerar el proceso de recuperación en caso de que se produzcan fallos.

Spanning Tree (STP)

El protocolo de árbol de extensión (IEEE 802.1d) mantiene la red libre de bucles. Para conseguir esto el dispositivo debe reconocer un bucle en la topología y bloquear uno o más puertos redundantes. El protocolo de árbol de extensión explora constantemente la red para detectar cambios en la topología. Cuando esto sucede el árbol de extensión reconfigura los puertos del switch o el bridge.

Este algoritmo cambia una red física con forma de malla, en la que se pueden formar bucles, en una red lógica en árbol en la que no se puede producir ningún lazo.

Funcionamiento:

- ✓ Se elige un nodo raíz. En un dominio de difusión sólo puede existir un nodo raíz. Todos los puertos de un nodo raíz se encuentran en estado de retransmisión y se denominan puertos designados.
- ✓ Para cada nodo que no es raíz, hay un puerto raíz. Los puertos raíz se encuentran en estado de retransmisión y proporcionan conectividad, hacia atrás con el nodo raíz.
- ✓ En cada segmento hay un solo puerto designado. El puerto designado se selecciona en el dispositivo que posee el trayecto de menor coste hasta el nodo raíz.

Para intercambiar mensajes de configuración con otros dispositivos se envían tramas de multidifusión denominadas Bridge Protocol Data Unit (BPDU). Por omisión la BPDU se envía cada dos segundos.

Uno de los elementos incluidos en la BPDU es el ID de bridge. El ID de bridge está compuesto por una prioridad (2 bytes) más la dirección MAC del bridge. La prioridad predeterminada es 32768.

El nodo raíz es el que tiene el ID de bridge más bajo. Si la prioridad es la misma el nodo raíz será el que tenga la MAC más baja.

Este bridge raíz establece el camino de menor coste para todas las redes; cada puerto tiene un parámetro configurable: el *Span path cost*. Todos los demás caminos son bloqueados para propósitos de bridge.

El árbol de expansión (*Spanning tree*) permanece efectivo hasta que ocurre un cambio en la topología. Esto sucede cuando se da cuenta de ello. El máximo de tiempo de duración del árbol de expansión es de cinco minutos. Cuando ocurre uno de estos cambios, sucede que o bien el actual bridge raíz ha redefinido la topología del árbol de expansión o se ha elegido un nuevo bridge raíz.

Vulnerabilidades:

Para forzar el recálculo del árbol de extensión podemos enviar BPDUs diferentes al switch de forma constante.

Si enviamos BPDUs con el ID de bridge más bajo en la red nos convertiremos en el nodo raíz y nos llegará todo el tráfico de la red.

Enrutamiento de Origen

Da la trayectoria completa desde el origen hasta el destino como secuencia de direcciones IP. Se requiere que le datagrama siga esa ruta exacta. Esta opción se usa sobre todo cuando los administradores de sistemas envíen paquetes de emergencia porque las tablas de enrutamiento se han corrompido, o para hacer mediciones de tiempo.

La opción de enrutamiento libre desde el origen requiere que el paquete pase por los enrutadores indicados en la lista, y en el orden especificado, pero se le permite pasar a través de otros enrutadores en el camino. Normalmente esta opción solo indicará algunos enrutadores, para obligar a una trayectoria en particular.

• Enrutamientos: estático vs. dinámico, exterior vs. interior (R, E)

Enrutamiento estático vs dinámico

Para calcular las entradas de las tablas de enrutamiento se utilizan dos métodos básicos:

- ✓ **Enrutamiento Estático:** Un programa calcula e instala las rutas al arrancar el conmutador de paquetes; las rutas no cambian.
- ✓ **Enrutamiento Dinámico:** Un programa construye una tabla inicial de enrutamiento al arrancar el conmutador de paquetes; entonces, el programa modifica la tabla a medida que cambian las condiciones de la red.

Cada enrutamiento tiene ventajas y desventajas. Las ventajas principales del enrutamiento estático son su simplicidad y su baja sobrecarga en la red. La desventaja principal es la inflexibilidad (las rutas estáticas no pueden cambiarse con facilidad).

La mayor parte de las redes acuden al enrutamiento dinámico porque permite que la red maneje automáticamente los problemas. Por ejemplo, mediante programas se puede supervisar el tráfico de la red así como el estado del hardware de red. Los programas modifican las rutas para ajustarse a las fallas. Debido a que las redes grandes se diseñan con conexiones redundantes para el manejo de fallas ocasionales del hardware, la mayor parte de las redes grandes usan una forma de enrutamiento dinámico.

Interior vs. Exterior

Dado que cada red de una interred es independiente de las demás, con frecuencia se denomina sistema autónomo (AS, *Autonomous System*).

Un protocolo interior de dispositivos de encaminamiento (IRP) pasa la información de encaminamiento entre los dispositivos de encaminamiento dentro de un sistema autónomo. El protocolo que se usa dentro de un sistema autónomo no necesita que esté implementado fuera del sistema. Esta flexibilidad permite que los IRP se hagan a medida para aplicaciones y requisitos específicos.

Podría ocurrir que un conjunto de redes esté construido con más de un sistema autónomo. Por ejemplo todas las LAN que formen un complejo de oficinas. Este sistema se podría unir a otros sistemas autónomos a través de una

red de área amplia. En esta situación los dispositivos de encaminamiento en un sistema autónomo necesitan al menos un nivel mínimo de información referente a las redes fuera del sistema autónomo a las que se puede acceder. El protocolo que se utiliza para pasar información de encaminamiento entre sistemas autónomos diferentes se conoce como protocolo de dispositivo de encaminamiento exterior (ERP).

Un protocolo IRP necesita construir un modelo más bien detallado de la interconexión de los dispositivos de encaminamiento dentro de un AS para poder calcular el camino con el menor coste, esto significa que un protocolo ERP es más simple y utiliza menos información detallada que un protocolo IRP.

Otra diferencia entre el enrutamiento interior y el exterior es el costo. Dentro de una sola red, normalmente se aplica un solo algoritmo de cargo. Sin embargo, redes diferentes pueden estar bajo administraciones diferentes, y una ruta puede ser menos cara que otra. Del mismo modo, la calidad de servicio ofrecida por diferentes redes puede ser distinta, y ésta puede ser una razón para escoger una ruta y no otra.

- **Protocolos de enrutamiento: camino más corto, múltiple, centralizado, aislado, distribuido y jerárquico (R, E)**

Camino más Corto

Lo que se pretende en el enrutamiento de camino más corto es encontrar el camino más rápido y óptimo para que viaje la información, este cálculo se puede dar en base a varios criterios, como distancia, ancho de banda, tráfico medio, costo de comunicación, longitud media de las colas, retardo medio, etc. El algoritmo más usado para el cálculo del camino más corto entre un nodo fuente y los demás nodos es el algoritmo de Dijkstra, el cual construye una tabla de siguientes saltos durante el cálculo de las trayectorias más cortas.

Múltiple

Pueden establecerse múltiples caminos activos entre los segmentos LAN y seleccionar entre los caminos redundantes.

Se puede buscar entre diferentes caminos activos y determinar en un momento determinado cuál resulta más adecuado.

Si un router A realiza una transmisión que necesita enviarse al router D, puede enviar el mensaje al router C o al B, y el mensaje será enviado al router D. Los routers tienen la posibilidad de evaluar ambos caminos y decidir la mejor ruta para esta transmisión.

Centralizado

Hay dos lugares donde se puede decidir hacia dónde debe enviarse un paquete desde un nodo: una es en el propio nodo (encaminamiento distribuido) y otra en un nodo señalado para esta tarea (encaminamiento centralizado). Esta última forma tiene el inconveniente de que si este nodo se estropea, el encaminamiento de todos los nodos que dependen de este nodo de encaminamiento es imposible, y todos los nodos serán inservibles .

Aislado

Unos de los algoritmos usados en enrutamiento aislado es la **Inundación (Flooding)**, en el cual el paquete entrante se envía por cada línea de salida excepto por la que llegó.

Entre sus aplicaciones típicas tenemos: Militares (resistencia a fallos), Bases de Datos (actualizaciones concurrentes) o para comprobar la bondad de otros encaminamientos, ya que por inundación se encuentra siempre el mejor camino porque se prueban todos (ignoremos la gran sobrecarga).

Si no se controlan puede aparecer un número infinito de paquetes en la red. Algunos tipos de control son:

- ✓ Incluir un contador de saltos en la cabecera que se decrementa en cada proceso. Se descarta el paquete al caer a 0. Idealmente el contador debe inicializarse al peor caso (diámetro total de la red) o a la distancia entre emisor y receptor (si se conoce).
- ✓ El enrutador fuente numera los paquetes. Cada enrutador tiene una lista por cada enrutador fuente y número de paquete ya visto. Para evitar grandes listas se usa un contador n para indicar que todos los números de

secuencia hasta n se han visto. Se descartan los duplicados que lleguen (se reconocen por su número de secuencia y enrutador origen).

Otro tipo de enrutamiento es el **Flooding Selectivo**, el cual sólo repite el paquete por las líneas orientadas en la dirección de destino correcta (si se conoce la topología de la red).

Por último en el algoritmo de **Random Walk** cada enrutador elige aleatoriamente la línea de salida para cada paquete. Si añadimos un intento de seleccionar la línea en la dirección adecuada y la red está muy interconectada este algoritmo aprovecha muy bien las rutas alternativas y es muy robusto.

Distribuido

En el encaminamiento distribuido cada comutador de paquetes calcula localmente su tabla de enrutamiento y luego envía mensajes por la red a los comutadores de paquetes vecinos para informarles el resultado.

Las redes que usan cálculo distribuido de rutas disponen que cada comutador de paquetes envíe periódicamente su información de enrutamiento a los vecinos (por ejemplo, cada pocos segundos). Tras un periodo inicial de arranque, cada comutador de paquetes aprende las trayectorias más cortas a los demás destinos. Los mensajes periódicos continuos permiten que la red se adapte ante fallas de los comutadores o enlaces de comunicación. Tras una falla, el comutador de paquetes deja de recibir actualizaciones del hardware que falló. En comutador continúa recibiendo actualizaciones de los vecinos operativos y, si existe una trayectoria alternativa, modifica su tabla de enrutamiento para evitar el hardware fallido.

Jerárquico

Una red puede crecer hasta el punto en que ya no es factible que cada enrutador tenga una entrada para cada uno de los demás enrutadores, por lo que el enrutamiento tendrá que hacerse jerárquicamente, como ocurre en la red telefónica.

Al usarse el enrutamiento jerárquico, los enrutadores se dividen en lo que llamaremos regiones, donde cada enrutador conoce todos los detalles de la manera de enrutar paquetes a destinos dentro de su propia región, pero no sabe nada de la estructura interna de las otras regiones. Al interconectar diferentes redes es natural considerar cada una como región independiente, a fin de liberar a los enrutadores de una red de la necesidad de conocer la estructura topológica de las demás.

En las redes enormes puede ser insuficiente una jerarquía de dos niveles; puede ser necesario agrupar las regiones en cúmulos, los cúmulos en zonas, las zonas en grupos, etc., hasta que se nos agoten los nombres para los agregados.

• Conceptos básicos de: RIP, IGRP, EGP, BGP y OSPF (R, E)

RIP

RIP es una implementación directa del encaminamiento vector-distancia para LANs. Utiliza UDP como protocolo de transporte, con el número de puerto 520 como puerto de destino. RIP opera en uno de dos modos: *activo* (normalmente usado por "routers") y *pasivo* (normalmente usado por hosts). Los mensajes RIP se envían en datagramas UDP y cada uno contiene hasta 25 pares de números.

Tanto el modo activo como el pasivo escuchan todos los mensajes de broadcast y actualizan su tabla de encaminamiento según el algoritmo vector-distancia descrito antes.

Operaciones básicas.

- ✓ Cuando RIP se inicia envía un mensaje a cada uno de sus vecinos(en el puerto bien conocido 520) pidiendo una copia de la tabla de encaminamiento de vecino. Este mensaje es una solicitud (el campo "command" se pone a 1) con "address family" a 0 y "metric" a 16. Los "routers" vecinos devuelven una copia de sus tablas de encaminamiento.

- ✓ Cuando RIP está en modo activo envía toda o parte de su tabla de encaminamiento a todos los vecinos (por broadcastadcast y/o con enlaces punto a punto). Esto se hace cada 30 segundos. La tabla de encaminamiento se envía como respuesta ("command" vale 2, aun que no haya habido petición).
- ✓ Cuando RIP descubre que una métrica ha cambiado, la difunde por broadcastadcast a los demás "routers".
- ✓ Cuando RIP recibe una respuesta, el mensaje se valida y la tabla local se actualiza si es necesario.
- ✓ Para mejorar el rendimiento y la fiabilidad, RIP especifica que una vez que un "router" (o host) a aprendido una ruta de otro, debe guardarla hasta que conozca una mejor (de coste estrictamente menor). Esto evita que los "routers" oscilen entre dos o más rutas de igual coste.
- ✓ Cuando RIP recibe una petición, distinta de la solicitud de su tabla, se devuelve como respuesta la métrica para cada entrada de dicha petición fijada al valor de la tabla local de encaminamiento. Si no existe ruta en la tabla local, se pone a 16.
- ✓ Las rutas que RIP aprende de otros "routers" expiran a menos que se vuelvan a difundir en 180 segundos (6 ciclos de broadcastadcast). Cuando una ruta expira, su métrica se pone a infinito, la invalidación de la ruta se difunde a los vecinos, y 60 segundos más tarde, se borra de la tabla.

Limitaciones:

RIP no está diseñado para resolver cualquier posible problema de encaminamiento. Sin embargo, RIP está muy extendido y es probable que permanezca sin sustituir durante algún tiempo. Tiene las siguientes limitaciones:

- ✓ El coste máximo permitido en RIP es 16, que significa que la red es inalcanzable. De esta forma, RIP es inadecuado para redes grandes(es decir, aquellas en las que la cuenta de saltos puede aproximarse perfectamente a 16).
- ✓ RIP no soporta máscaras de subred de longitud variable (*variable subnetting*). En un mensaje RIP no hay ningún modo de especificar una máscara de subred asociada a una dirección IP.
- ✓ RIP carece de servicios para garantizar que las actualizaciones proceden de "routers" autorizados. Es un protocolo inseguro.
- ✓ RIP sólo usa métricas fijas para comparar rutas alternativas. No es apropiado para situaciones en las que las rutas necesitan elegirse basándose en parámetros de tiempo real tales como el retardo, la fiabilidad o la carga.
- ✓ El protocolo depende de la *cuenta hasta infinito* para resolver algunas situaciones inusuales. RIP especifica mecanismos para minimizar los problemas con la cuenta hasta infinito que permiten usarlo con dominios mayores, pero eventualmente su operatividad será nula. No existe un límite superior prefijado, pero a nivel práctico este depende de la frecuencia de cambios en la topología, los detalles de la topología de la red, y lo que se considere como un intervalo máximo de tiempo para que la topología de encaminamiento se estabilice.

IGRP

El protocolo IGRP es un protocolo interno IGP pero también se puede utilizar como protocolo externo. IGRP asigna el número de routers para coordinar su routing.

IGRP es un *interior-gateway protocol* (IGP) que trabaja con *vector de distancias*. El protocolo de enrutamiento de vector de distancias consiste en que cada router manda toda o una porción de su tabla de enrutamiento en mensajes de update en intervalos regulares a cada uno de sus routers vecinos. Como la información se prolifera a través de la red, los routers pueden calcular distancias para todos los nodos dentro de la interconexión.

- Las metas del IGRP son las siguientes:
 - ✓ Routing estable, ya que no produce bucles.
 - ✓ Rápida respuesta a cambios de la topología de la red.
 - ✓ No usa mas ancho de banda del que ocupa, pequeño overheat.
 - ✓ Reparte el tráfico entre rutas paralelas.
 - ✓ Toma en cuenta la tasa de errores y el nivel de tráfico de diferentes caminos.
 - ✓ Tiene capacidad de manejar diferentes "Tipos de Servicios" con un conjunto simple de información.
- IGRP puede manejar diferentes tipos de protocolos, aunque en la actualidad esta montado sobre TCP/IP.
- La métrica que utiliza IGRP se compone de los siguientes aspectos:
 - ✓ El **retardo de la topología (topológicas delay time)**.
 - ✓ El **ancho de banda (bandwidth of the narrowest bandwidth segment of the path)**.

- ✓ La ocupación de la línea (channel occupancy of the path).
- ✓ La fiabilidad (reliability of the path).

Tipos de Paquetes:

IGRP Aumentado usa los siguientes tipos de paquetes:

- ✓ *Hello and acknowledgment*--Paquetes Hello son multicast para el descubrimiento y recuperación de vecinos y no requiere acknowledgment. Un paquete *acknowledgment* es un paquete hello que no contiene data. Paquetes Acknowledgment contiene un número diferente de cero, y son siempre enviados con dirección unicast.
- ✓ *Update*--Paquetes Update son usados para convenir alcanzabilidad de destinos. Cuando un vecino es descubierto, paquetes update únicas son enviados, por lo que el vecino puede construir su tabla de topología. En otro caso, los updates son multicast.
- ✓ *Query and reply*--Paquetes Query and reply son enviados cuando un destino no posee sucesores factibles. Paquetes Query son siempre multicast. Paquetes Reply son enviados en respuesta a paquetes query para indicar al emisor que no es necesario recalcular la ruta porque hay sucesores factibles. Paquetes Reply son unicast al emisor del query.
- ✓ *Request*--Paquetes Request son usados para obtener información específica de uno o más vecinos. Paquetes Request packets son usados en rutas servidor aplicaciones y pueden ser multi o unicast.

EGP

Los EGPs("Exterior Gateway Protocols") se usan para intercambiar información de encaminamiento entre distintos sistemas autónomos.

EGP es el protocolo utilizado para el intercambio de información de encaminamiento entre pasarelas *exteriores*(que no pertenezcan al mismo sistema autónomo).

Las pasarelas EGP sólo pueden retransmitir información de accesibilidad para las redes de su sistema autónomo. La pasarela debe recoger esta información, habitualmente por medio de un IGP, usado para intercambiar información entre pasarelas del mismo sistema autónomo.

EGP se basa en el sondeo periódico empleando intercambios de mensajes *Hello/I Hear You*, para monitorizar la accesibilidad de los vecinos y para sondear si hay solicitudes de actualización. EGP restringe las pasarelas exteriores al permitirles anunciar sólo las redes de destino accesibles en el sistema autónomo de la pasarela. De esta forma, una pasarela exterior que usa EGP pasa información a sus vecinos EGP pero no anuncia la información de accesibilidad de estos(las pasarelas son vecinos si intercambian información de encaminamiento) fuera del sistema autónomo. Tiene tres características principales:

- ✓ Soporta un protocolo NAP("Neighbor Acquisition Protocol). Dos pasarelas se pueden considerar vecinas si están conectadas por una red que es transparente para ambas. EGP no especifica la forma en que una pasarela decide inicialmente que quiere ser vecina de otra. Para convertirse en vecina, debe enviar un mensaje "*Acquisition confirm*" como respuesta a un *Acquisition Request*. Este paso es necesario para obtener información de encaminamiento de otra pasarela.
- ✓ Soporta un protocolo NR("Neighbor Reachability"). La pasarela lo usa para mantener información en tiempo real sobre la accesibilidad de sus vecinos. El protocolo EGP proporciona dos tipos de mensajes para ese fin: un mensaje *Hello* y un mensaje *I Hear You* (respuesta a Hello).
- ✓ Soporta mensajes de actualización(o mensajes NR) que llevan información de encaminamiento. No se requiere ninguna pasarela para enviar mensajes NR a otra pasarela, excepto como respuesta a una petición de sondeo("poll request").

Para realizar estas tres funciones básicas, EGP define 10 tipos de mensajes:

- ✓ *Acquisition Request*: Sigue la solicitud que una pasarela se convierta en vecina .
- ✓ *Acquisition Confirm*: Respuesta afirmativa a un "acquisition request".
- ✓ *Acquisition Refuse*: Respuesta negativa a un "acquisition request".
- ✓ *Cease Request*: Solicitud de terminación de la relación de vecindad.
- ✓ *Cease Confirm*: Confirmación para que cesen las peticiones.
- ✓ *Hello*: Solicitud de respuesta e un vecino, si está vivo.
- ✓ *I Hear You*: Respuesta el mensaje Hello .
- ✓ *Poll Request*: Solicitud de la tabla de encaminamiento de la red .
- ✓ *Routing Update*: Información de accesibilidad de la red..

- ✓ *Error:* Respuesta a un mensaje incorrecto

Como se indicó arriba, los mensajes EGP asocian un descriptor "*distances*" a cada ruta. Pero EGP *no interpreta estos valores*. Simplemente sirven como indicación de la accesibilidad o inaccesibilidad de una red(un valor de 255 significa que la red es inalcanzable). El valor no se puede usar para calcular cuál es la más corta de dos rutas a menos que ambas pertenezcan al mismo AS. Por esta razón, EGP no se puede usar como algoritmo de encaminamiento. Como resultado sólo habrá una única ruta del exterior de la pasarela a una red.

BGP

Dentro de varios sistemas autónomos se usa el protocolo BGP (Border Gateway Protocol, protocolo de pasarela exterior). Fue diseñado para permitir muchos tipos de políticas de enrutamiento aplicables al tráfico inter-sistemas autónomos, las cuales se configuran manualmente en cada enrutador BGP.

Desde el punto de vista de un enrutador BGP, el mundo consiste en otros enrutadores BGP y en las líneas que los conectan. Se consideran conectados dos enrutadores BGP si comparten una red común. Dado el interés especial del BGP en el tráfico de tránsito, las redes se agrupan en una de tres categorías:

- ✓ *Redes de Punta:* Solo tienen una conexión al grafo BGP, no se pueden usar para tráfico en tránsito porque no hay nadie del otro lado.
- ✓ *Redes Multiconectadas:* Podrían usarse para el tráfico de tránsito excepto que se niegan.
- ✓ *Redes de Tránsito:* Son como los backbones, y están dispuestas a manejar los paquetes de terceros, posiblemente con algunas restricciones.

Los pares de enrutadores BGP se comunican entre ellos estableciendo conexiones TCP. Este tipo de operación proporciona comunicación confiable y esconde todos los detalles de la red por la que se pasa.

El BGP fundamentalmente es un protocolo de vector de distancia, solo que en lugar de mantener sólo el costo a cada destino, cada enrutador BGP lleva el registro de la trayectoria seguida. Del mismo modo, en lugar de dar periódicamente a cada vecino sus costos estimados a todos los destinos posibles, cada enrutador BGP le dice a sus vecinos la trayectoria exacta que está usando.

OSPF

El protocolo OSPF (Open Shortest Path First – abrir primero la trayectoria mas corta) se usa muy frecuentemente como protocolo de encaminamiento interior en redes TCP/IP. Cuando se diseñó se quiso que cumpliera los siguientes requisitos:

- ✓ Ser abierto en el sentido de que no fuera propiedad de una compañía.
- ✓ Que permitiera reconocer varias métricas, entre ellas, la distancia física y el retardo.
- ✓ Ser dinámico, es decir, que se adaptará rápida y automáticamente a los cambio de la topología.
- ✓ Ser capaz de realizar en encaminamiento dependiendo del tipo de servicio.
- ✓ Que pudiera equilibrar las cargas dividiendo la misma entre varias líneas.
- ✓ Que reconociera sistemas jerárquicos pues un único ordenador no puede conocer la estructura completa de Internet.
- ✓ Que implementara un mínimo de seguridad.

El protocolo OSPF reconoce tres tipos de conexiones y redes:

- ✓ Líneas punto a punto entre dos dispositivos de encaminamiento.
- ✓ Redes multiacceso con difusión (por ejemplo, la mayoría de redes LAN).
- ✓ Redes multiacceso sin difusión (por ejemplo, la mayoría de redes WAN de conmutación de paquetes).

Diremos que una red es multiacceso si tiene varios dispositivos de encaminamiento que se pueden comunicar con los demás.

La función del OSPF es encontrar la trayectoria mas corta de un dispositivo de encaminamiento a todos los demás. Cada dispositivo de encaminamiento tiene almacenada en una base de datos la topología de la red de la que forma parte. La representación de esta topología se expresa como un grafo dirigido.

Al arrancar un dispositivo de almacenamiento, este protocolo envía paquetes HELLO por todas sus líneas punto a punto y los retransmite a todos los demás dispositivos de encaminamiento. Gracias a las respuestas que recibe sabe cuales son sus dispositivos de encaminamiento **vecinos**. El OSPF se basa en el intercambio de información entre los dispositivos de encaminamiento **adyacentes**, que no es lo mismo que vecinos. Para que no todos los dispositivos tengan que hablar con los demás, se designa uno como adyacente a todos los demás y es este el que intercambia información con los restantes.

Por motivos de seguridad se determinada un dispositivo de encaminamiento como secundario por si el primario cae.

Normalmente, el dispositivo de encaminamiento inunda de mensajes de ACTUALIZACION DE ESTADO DEL ENLACE a todos sus dispositivos de encaminamiento adyacentes. Estos mensajes tienen un número de secuencia y además para hacerlos confiables son reconocidos por el mensaje RECONOCIMIENTO DE ESTADO DEL ENLACE. Además existen otros dos mensajes: DESCRIPCIÓN DE LA BASE DE DATOS que es utilizado para anunciar las actualizaciones que tiene el transmisor, y SOLICITUD DE ESTADO DE ENLACE que es utilizado para solicitar información a un compañero. Todos los mensajes utilizados en el OSPF se envían como paquetes IP en bruto.

3. Dispositivos para interconexión

• Modems (R, E, RP)

Un módem es un dispositivo que permite a los equipos comunicarse a través de una línea telefónica. Cuando los equipos están demasiado alejados como para conectarse a través de un cable estándar, se puede llevar a cabo la comunicación entre ellos mediante un *módem*. En un entorno de red, los módems actúan como un medio de comunicación entre redes y como una forma de conectar el mundo que existe más allá de la red local.

Funciones básicas de un módem:

Los equipos no se pueden conectar a través de una línea telefónica, puesto que éstos se comunican enviando pulsos electrónicos digitales (señales electrónicas) y una línea telefónica sólo puede enviar ondas (sonido) analógicas.

Un señal digital tiene un formato binario. La señal puede tener un valor de 0 ó 1. Una señal analógica se puede representar como una curva suavizada que puede representar un rango infinito de valores.

El módem que se encuentra en el PC emisor convierte las señales digitales en ondas analógicas y transmite estas ondas analógicas a través de la línea telefónica. El módem que recibe la señal, convierte las señales analógicas que le llegan en señales digitales para que las reciba el PC.

En otras palabras, un módem emisor *M*Odula las señales digitales en señales analógicas y un módem receptor *D*Emodula las señales que recibe en señales digitales.

Tipos de módems:

Existen tres tipos diferentes de módems, puesto que los distintos entornos de comunicación requieren diferentes métodos de envío de datos. Estos entornos se pueden dividir en dos áreas relacionadas con el ritmo de las comunicaciones: Asíncrona y Síncrona.

Comunicación asíncrona (Async)

La *comunicación asíncrona*, conocida como «*async*», es probablemente la forma de conexión más extendida. Esto es debido a que *async* se desarrolló para utilizar las líneas telefónicas. Cada carácter (letra, número o símbolo) se introduce en una cadena de bits. Cada una de estas cadenas se separa del resto mediante un bit de inicio de carácter y un bit de final de carácter. Los dispositivos emisor y receptor deben estar de acuerdo en la secuencia de bit inicial y final. El equipo destino utiliza los marcadores de bit inicial y final para planificar sus funciones relativas al ritmo de recepción, de forma que esté preparado para recibir el siguiente byte de datos.

Las transmisiones asíncronas en líneas telefónicas pueden alcanzar hasta 28.800 bps. No obstante, los métodos de compresión de datos más recientes permiten pasar de 28.800 bps a 115.200 bps en sistemas conectados directamente.

Debido al potencial de errores que puede presentar, *async* puede incluir un bit especial, denominado *bit de paridad*, que se utiliza en un esquema de corrección y comprobación de errores, denominado *comprobación de paridad*. En la comprobación de paridad, el número de bits enviados debe coincidir exactamente con el número de bits recibidos.

Los módems asíncronos, o serie, son más baratos que los módems síncronos, puesto que los asíncronos no necesitan la circuitería y los componentes necesarios para controlar el ritmo que de las transmisiones síncronas requieren los módems síncronos.

Comunicación síncrona

La *comunicación síncrona* confía en un esquema temporal coordinado entre dos dispositivos para separar los grupos de bits y transmitirlos en bloques conocidos como «tramas». Se utilizan caracteres especiales para comenzar la sincronización y comprobar periódicamente su precisión.

Dado que los bits se envían y se reciben en un proceso controlado (sincronizado) y cronometrado, no se requieren los bits de inicio y final. Las transmisiones se detienen cuando se alcanza el final de una trama y comienzan, de nuevo, con una nueva. Este enfoque de inicio y final es mucho más eficiente que la transmisión asíncrona, especialmente cuando se están transfiriendo grandes paquetes de datos. Este incremento en eficiencia es menos destacable cuando se envían pequeños paquetes.

Si aparece un error, el esquema de corrección y detección de errores síncrono genera una retransmisión.

La comunicación síncrona se utiliza en la mayoría de todas las comunicaciones de red y digitales. Por ejemplo, si está utilizando líneas digitales para conectar equipos remotos, debería utilizar módems síncronos, en lugar de asíncronos, para conectar el equipo a la línea digital. Normalmente, su alto precio y complejidad ha mantenido a los módems síncronos fuera del mercado de los equipos personales.

Línea digital abonada asimétrica (ADSL, Asymmetric Digital Subscriber Line)

La última tecnología de módem disponible es una línea digital abonada asimétrica (ADSL). Esta tecnología convierte las líneas telefónicas actuales de par trenzado en vías de acceso para las comunicaciones de datos de alta velocidad y multimedia. Estas nuevas conexiones pueden transmitir por encima de los 8 Mbps para el abonado y de hasta 1Mbps desde el propio abonado.

No obstante, ADSL no está exenta de inconvenientes. La tecnología requiere un hardware especial, incluyendo un módem ADSL en cada extremo de la conexión. Además, necesita un cableado de banda amplia, que está disponible actualmente en muy pocas localizaciones y existe un límite en la longitud de conexión.

• Repetidores (R, E, RP)

Operan en el nivel más bajo del modelo OSI. Se emplean para ampliar el alcance geográfico de una red, conectando dos o más LAN's. Operan amplificando *todas* las señales eléctricas que reciben, es decir son transparentes hasta los protocolos más altos. No proporciona ningún tipo de aislamiento entre redes. Sólo pueden proporcionar una gestión de redes simple. Su mayor ventaja es poder conectar redes con diferente medio de

transmisión como por ejemplo ethernet sobre cable coaxial a ethernet sobre fibra óptica. Más que repetidores lo que se suele utilizar son regeneradores que no solo amplifican, que supone amplificar la señal con el ruido adicional, sino que devuelven la señal digital original eliminando el ruido.

Cuando las señales viajan a través de un cable, se degradan y se distorsionan en un proceso denominado «atenuación». Si un cable es bastante largo, la attenuación provocará finalmente que una señal sea prácticamente irreconocible. La instalación de un repetidor permite a las señales viajar sobre distancias más largas.

Un repetidor funciona en el nivel físico del modelo de referencia OSI para regenerar las señales de la red y reenviarla a otros segmentos.

El repetidor toma una señal débil de un segmento, la regenera y la pasa al siguiente segmento. Para pasar los datos de un segmento a otro a través del repetidor, deben ser idénticos en cada segmento los paquetes y los protocolos Control lógico de enlace (LLC; *Logical Link Control*). Un repetidor no activará la comunicación, por ejemplo, entre una LAN (Ethernet) 802.3 y una LAN (Token Ring) 802.5.

Los repetidores no traducen o filtran señales. Un repetidor funciona cuando los segmentos que unen el repetidor utilizan el mismo método de acceso. Un repetidor no puede conectar un segmento que utiliza CSMA/CD con un segmento que utiliza el método de acceso por paso de testigo. Es decir, un repetidor no puede traducir un paquete Ethernet en un paquete Token Ring.

• Concentradores (*hubs*) (R, E, RP)

Modo para conectar hosts y extender los segmentos de redes locales de difusión. Tienen (entre 4 y 64) conectores a los que conecta hosts. También son utilizados para eludir limitaciones de distancia en un único segmento y proporcionar un modo de añadir hosts adicionales.

Es el componente hardware central de una topología en estrella. Además, los hubs se pueden utilizar para extender el tamaño de una LAN. Aunque la utilización de un hub no implica convertir una LAN en una WAN, la conexión o incorporación de hubs a una LAN puede incrementar, de forma positiva, el número de estaciones. Este método de expansión de una LAN es bastante popular, pero supone muchas limitaciones de diseño.

Es importante tener cuidado cuando se conectan los hubs. Los cables de paso se conectan de forma diferente que los cables estándares de enlace. Compruebe con los fabricantes si se necesita un cable de enlace estándar o un cable de paso.

• Commutadores (*switches*) (R, E)

Función similar a un routers, pero restringida a redes locales. La ventaja de estos sobre los concentradores es que pueden separar el tráfico entrante y transmitirlo solo hacia la red de salida relevante, reduciendo la congestión con otras redes a las que estas conectados.

Los commutadores tienen la funcionalidad de los concentradores a los que añaden la capacidad principal de dedicar todo el ancho de banda de forma exclusiva a cualquier comunicación entre sus puertos. Esto se consigue debido a que el commutador no actúa como repetidor multipuerto, sino que únicamente envía paquetes de datos hacia aquella puerta a la que van dirigidos. Esto es posible debido a que los equipos configuran unas tablas de encaminamiento con las direcciones.

Un Switch es un dispositivo de Networking situado en la capa 2 del modelo de referencia OSI. El switch se denomina puente multipuerto, así como el hub se denomina repetidor multipuerto. La diferencia entre el hub y el switch es que los switches toman decisiones basándose en las direcciones MAC y los hubs no toman ninguna decisión. Como los switches son capaces de tomar decisiones, así hacen que la LAN sea mucho más eficiente. Los switches hacen esto "comutando" datos sólo desde el puerto al cual está conectado el host correspondiente. A diferencia de esto, el hub envía datos a través de todos los puertos de modo que todos los hosts deban ver y procesar (aceptar o rechazar) todos los datos. Esto hace que la LAN sea mas lenta.

A primera vista los switches parecen a menudo similares a los hubs. Tanto los hubs como los switches tienen varios puertos de conexión (pueden ser de 8, 12, 24 o 48, o conectando 2 de 24 en serie), dado que una de sus funciones es la concentración de conectividad (permitir que varios dispositivos se conecten a un punto de la red).

La diferencia entre un hub y un switch está dada por lo que sucede dentro de cada dispositivo. El propósito del switch es concentrar la conectividad, haciendo que la transmisión de datos sea más eficiente. Por el momento, piense en el switch como un elemento que puede combinar la conectividad de un hub con la regulación de tráfico de un puente en cada puerto. El switch comunica paquetes desde los puertos (las interfaces) de entrada hacia los puertos de salida, suministrando a cada puerto el ancho de banda total. Básicamente un Switch es un administrador inteligente del ancho de banda.

• Multiplexores (R, E)

El multiplexor (MUX) es un dispositivo que puede encontrarse en casi todas las instalaciones. Su misión consiste en permitir que varios ETD o puertos compartan una misma línea de comunicación, por lo general un canal telefónico. Esto es posible siempre que en el canal tenga capacidad suficiente para permitir su uso compartido.

Son dispositivos que logran transmitir varios canales en un solo medio de transmisión reuniendo varias señales a baja velocidad y transmíténdolas posteriormente a todas a través de un canal de alta velocidad. Pudiendo ser estos analógicos (**FDM**) o digitales (**TDM**).

Son circuitos realmente importantes en el diseño de sistemas que requieran un cierto tráfico y comunicación entre distintos componentes y se necesite controlar en todo momento que componente es quien envía los datos.

En realidad se puede asimilar a un selector, por medio de unas entradas de control se selecciona la entrada que se desee reflejada en la salida. Esto se consigue utilizando principalmente puertas **XOR**, de ahí su nombre multiplexor.

El empleo de multiplexores permite reducir de forma sustancial el número necesario de canales de comunicaciones. Su precio suele estar justificado por el ahorro en costos de líneas que proporciona. Los multiplexores son muy útiles también en los entornos locales (sin línea telefónica) ya que permiten reducir la cantidad de cable que es necesario tender en un edificio para enlazar las distintas terminales con la computadora central.

Todas las terminales están conectadas a otro multiplexor por medio de uno solo. Los dos multiplexores tienen la capacidad de información por separado. El multiplexor del **nodo A** dispositivos conectados a él y los transmite por velocidad, el multiplexor del **nodo B** recibe la señal el canal y los envía a los dispositivos correctos.

Existen tres tipos de Multiplexación:

- ✓ **Multiplexación por División de Frecuencia:** Divide el ancho de banda de una línea entre varios canales, donde cada canal ocupa una parte del ancho de banda de frecuencia total. Es usada para la industria de comunicaciones. La técnica de FDM divide el ancho de banda total de entrada y salida en el mismo numero de canales en el circuito, dependiendo en el numero de puertos y dispositivos que sean soportados. El rango total de información de entrada de los dispositivos o terminales conectados al multiplexor no pueden exceder el rango de salida.
- ✓ **Multiplexación por División de Tiempo:** Aquí cada canal tiene asignado un periodo o ranura de tiempo en el canal principal y las distintas ranuras de tiempo están repartidas por igual en todos los canales. Tiene la desventaja de que en caso de que un canal no sea usado, esa ranura de tiempo no se aprovecha por los otros canales, enviándose en vez de datos bits de relleno. Los multiplexores que utilizan la tecnología TDM son dispositivos digitales que combinan varias señales digitales de dispositivos en un solo medio de transmisión digital.
- ✓ **Multiplexación por División de Tiempo Estadísticos:** Una versión mas eficiente de TDM es STDM. STDM funciona de la misma manera que TDM solo con la ventaja de que utiliza mejor el uso de canales que no están siendo utilizados y reconectando estos time slots a otros dispositivos conectados que puedan utilizar este ancho de banda que esta disponible.

• Puentes (R, E, RP)

Enlazan redes de distintos tipos. Algunos puentes comunican varias redes y se llama puente/routers ya que efectúan funciones de encaminamiento.

Al igual que un repetidor, un bridge puede unir segmentos o grupos de trabajo LAN. Sin embargo, un bridge puede, además, dividir una red para aislar el tráfico o los problemas. Por ejemplo, si el volumen del tráfico de uno o dos equipos o de un departamento está sobrecargando la red con los datos y ralentizan todas las operaciones, el bridge podría aislar a estos equipos o al departamento. Los bridges se pueden utilizar para:

- ✓ Extender la longitud de un segmento.
- ✓ Proporcionar un incremento en el número de equipos de la red.
- ✓ Reducir los cuellos de botella del tráfico resultantes de un número excesivo de equipos conectados.
- ✓ Dividir una red sobrecargada en dos redes separadas, reduciendo la cantidad de tráfico en cada segmento y haciendo que la red sea más eficiente.
- ✓ Enlazar medios físicos diferentes como par trenzado y Ethernet coaxial.

Los bridges trabajan a nivel de enlace de datos del modelo de referencia OSI y, por tanto, toda la información de los niveles superiores no está disponible para ellos. Más que distinguir entre un protocolo y otro, los bridges pasan todos los protocolos que aparecen en la red. Todos los protocolos se pasan a través de los bridges, de forma que aparecen en los equipos personales para determinar los protocolos que pueden reconocer.

Los bridges trabajan en el nivel MAC y, por ello, algunas veces se conocen como bridges de nivel MAC. Un bridge de nivel MAC:

- ✓ Escucha todo el tráfico.
- ✓ Comprueba la direcciones origen y destino de cada paquete.
- ✓ Construye una tabla de encaminamiento, donde la información está disponible.
- ✓ Reenvía paquetes de la siguiente forma:
 - ✓ Si el destino no aparece en la tabla de encaminamiento, el bridge reenvía el paquete a todos los segmentos.
 - ✓ Si el destino aparece en la tabla de encaminamiento, el bridge reenvía el paquete al segmento correspondiente (a menos que este segmento sea también el origen).

Envío de Paquetes:

Un bridge funciona considerando que cada nodo de la red tiene su propia dirección. Un bridge reenvía paquetes en función de la dirección del nodo destino.

Inicialmente, la tabla de encaminamiento del bridge está vacía. Cuando los nodos transmiten los paquetes, la dirección de origen se copia en la tabla de encaminamiento. Con esta información de la dirección, el bridge identifica qué equipos están en cada segmento de la red.

Los bridges generan sus tablas de encaminamiento en función de las direcciones de los equipos que han transmitido datos en la red. Los bridges utilizan, de forma específica, las direcciones de origen (dirección del dispositivo que inicia la transmisión) para crear una tabla de encaminamiento.

Cuando el bridge recibe un paquete, la dirección de origen se compara con la tabla de encaminamiento. Si no aparece la dirección de origen, se añade a la tabla. A continuación, el bridge compara la dirección de destino con la base de datos de la tabla de encaminamiento.

Si un bridge conoce la localización del nodo de destino, envía el paquete a dicha localización. Si no conoce el destino, envía el paquete a todos los segmentos.

Segmentación del tráfico de red:

Un bridge puede segmentar el tráfico mediante su tabla de encaminamiento. Un equipo en el segmento 1 (origen), envía datos a otro equipo (destino) también localizado en el segmento 1. Si la dirección de destino está en la tabla de encaminamiento, el bridge puede determinar que el equipo destino está también en el segmento 1. Dado que los equipos origen y destino están en el mismo segmento 1, se tiene que el paquete no se reenvía a través del bridge al segmento 2.

Por tanto, los bridges pueden utilizar las tablas de encaminamiento para reducir el tráfico de la red controlando los paquetes que se envían al resto de los segmentos. Este control (o restricción) del flujo del tráfico de red se conoce como «segmentación del tráfico de red».

Una red grande no está limitada a un solo bridge. Se pueden utilizar múltiples bridge para combinar diferentes redes pequeñas en una red más grande.

• **Enrutadores. Enrutadores multiprotocolo (R, E, RP)**

En una interred los routers pueden enlazarse mediante conexiones directas o pueden estar interconectados a través de subredes. Ellos son los responsables de reenviar paquetes de interred que llegan hacia las conexiones salientes correctas para lo cual se mantienen las tablas de encaminamiento.

En un entorno que está formado por diferentes segmentos de red con distintos protocolos y arquitecturas, el bridge podría resultar inadecuado para asegurar una comunicación rápida entre todos los segmentos. Una red de esta complejidad necesita un dispositivo que no sólo conozca la direcciones de cada segmento, sino también, que sea capaz de determinar el camino más rápido para el envío de datos y filtrado del tráfico de difusión en el segmento local. Este dispositivo se conoce como «router».

Los routers trabajan en el nivel de red del modelo de referencia OSI. Esto significa que pueden comutar y encaminar paquetes a través de múltiples redes. Realizan esto intercambiando información específica de protocolos entre las diferentes redes. Los routers leen en el paquete la información de direccionamiento de las redes complejas teniendo acceso a información adicional, puesto que trabajan a un nivel superior del modelo OSI en comparación con los bridges.

Los routers mantienen sus propias tablas de encaminamiento, normalmente constituidas por direcciones de red; también se pueden incluir las direcciones de los hosts si la arquitectura de red lo requiere. Para determinar la dirección de destino de los datos de llegada, las tablas de encaminamiento incluyen:

- ✓ Todas las direcciones de red conocidas.
- ✓ Instrucciones para la conexión con otras redes.
- ✓ Los posibles caminos entre los routers.
- ✓ El coste de enviar los datos a través de estos caminos.

Un router utiliza sus tablas de encaminamiento de datos para seleccionar la mejor ruta en función de los caminos disponibles y del coste.

Los routers requieren direcciones específicas. Entienden sólo los números de red que les permiten comunicarse con otros routers y direcciones NIC locales. Los routers no conversan con equipos remotos.

Cuando los routers reciben paquetes destinados a una red remota, los envían al router que gestiona la red de destino. En algunas ocasiones esto constituye una ventaja porque significa que los routers pueden:

- ✓ Segmentar grandes redes en otras más pequeñas.
- ✓ Actuar como barrera de seguridad entre los diferentes segmentos.
- ✓ Prohibir las «tormentas» de difusión, puesto que no se envían estos mensajes de difusión.

Dado que los routers sólo leen paquetes direccionalizados de red, no permiten pasar datos corruptos a la red. Por tanto, al no permitir pasar datos corruptos ni tormentas de difusión de datos, los routers implican muy poca tensión en las redes.

Los routers no ven la dirección del nodo de destino, sólo tienen control de las direcciones de red. Los routers pasarán información sólo si conocen la dirección de la red. Esta capacidad de controlar el paso de datos a través del router reduce la cantidad de tráfico entre las redes y permite a los routers utilizar estos enlaces de forma más eficiente que los bridges.

Selección de los caminos:

A diferencia de los bridges, los routers pueden establecer múltiples caminos activos entre los segmentos LAN y seleccionar entre los caminos redundantes. Los routers pueden enlazar segmentos que utilizan paquetes de datos y acceso al medio completamente diferentes, permitiendo utilizar a los routers distintos caminos disponibles. Esto significa que si un router no funciona, los datos todavía se pueden pasar a través de routers alternativos.

Un router puede escuchar una red e identificar las partes que están ocupadas. Esta información la utiliza para determinar el camino sobre el que envía los datos. Si un camino está ocupado, el router identifica un camino alternativo para poder enviar los datos.

Un router decide el camino que seguirá el paquete de datos determinando el número de saltos que se generan entre los segmentos de red.

Tipos de Routers:

Los tipos principales de routers son:

- ✓ **Estático.** Los routers estáticos requieren un administrador para generar y configurar manualmente la tabla de encaminamiento y para especificar cada ruta.
- ✓ **Dinámico.** Los routers dinámicos se diseñan para localizar, de forma automática, rutas y, por tanto, requieren un esfuerzo mínimo de instalación y configuración. Son más sofisticados que los routers estáticos, examinan la información de otros routers y toman decisiones a nivel de paquete sobre cómo enviar los datos a través de la red.

• Compuertas (gateways) (R, E)

Los gateways -el sistema de interconexión de redes más complejo- funciona en los tres niveles más altos del modelo OSI (sesión, presentación y aplicación). Los gateways pueden conectar redes de arquitecturas (pila de protocolos) completamente diferentes. Para hacer esto, los gateways convierten una arquitectura de red en otra sin afectar a los datos transmitidos.

Los gateways proporcionan muchos servicios de gestión de red y al igual que bridges y routers conectan tanto redes locales o redes extensas.

Los gateways activan la comunicación entre diferentes arquitecturas y entornos. Se encargan de empaquetar y convertir los datos de un entorno a otro, de forma que cada entorno pueda entender los datos del otro entorno. Un gateway empaqueta información para que coincida con los requerimientos del sistema destino. Los gateways pueden modificar el formato de un mensaje para que se ajuste al programa de aplicación en el destino de la transferencia. Por ejemplo, los gateways de correo electrónico, como el X.400, reciben mensajes en un formato, los formatean y envían en formato X.400 utilizado por el receptor, y viceversa.

Un gateway enlaza dos sistemas que no utilizan los mismos:

- ✓ Protocolos de comunicaciones.
- ✓ Estructuras de formateo de datos.
- ✓ Lenguajes.
- ✓ Arquitectura.

Los gateways interconectan redes heterogéneas; por ejemplo, pueden conectar un servidor Windows NT de Microsoft a una Arquitectura de red de los sistemas IBM (SNA). Los gateways modifican el formato de los datos y los adaptan al programa de aplicación del destino que recibe estos datos.

Los gateways son de tarea específica. Esto significa que están dedicados a un tipo de transferencia. A menudo, se referencian por su nombre de tarea (gateway Windows NT Server a SNA).

Un gateway utiliza los datos de un entorno, desmantela su pila de protocolo anterior y empaqueta los datos en la pila del protocolo de la red destino. Para procesar los datos, el gateway:

- ✓ Desactiva los datos de llegada a través de la pila del protocolo de la red.
- ✓ Encapsula los datos de salida en la pila del protocolo de otra red para permitir su transmisión.

Algunos gateways utilizan los siete niveles del modelo OSI, pero, normalmente, realizan la conversión de protocolo en el nivel de aplicación. No obstante, el nivel de funcionalidad varía ampliamente entre los distintos tipos de gateways.

Una utilización habitual de los gateways es actuar como traductores entre equipos

personales y miniequipos o entornos de grandes sistemas. Un gateway en un host que conecta los equipos de una LAN con los sistemas de miniequipo o grandes entornos (mainframe) que no reconocen los equipos conectados a la LAN.

Normalmente, los gateways se dedican en la red a servidores. Pueden utilizar un porcentaje significativo del ancho de banda disponible para un servidor, puesto que realizan tareas que implican una utilización importante de recursos, tales como las conversiones de protocolos. Si un servidor gateway se utiliza para múltiples tareas, será necesario adecuar las necesidades de ancho de banda y de RAM o se producirá una caída del rendimiento de las funciones del servidor.

Los gateways se consideran como opciones para la implementación, puesto que no implican una carga importante en los circuitos de comunicación de la red y realizan, de forma eficiente, tareas muy específicas.

D. Seguridad de la información

I. Seguridad de la información

1. Seguridad

- Conceptos básicos (R)

- Criptografía: llave secreta, llave pública, certificados de llaves públicas (R, E, RP)

Criptografía

Del griego kryptos (ocultar) y grafos (escribir), literalmente escritura oculta, la criptografía es la arte y ciencia de cifrar y descifrar datos utilizando las matemáticas. Haciendo posible intercambiar datos de manera que sólo puedan ser leídos por las personas a quienes van dirigidos.

Con más precisión deberíamos hablar de criptología, que sí hace referencia a las técnicas de cifrado, la criptografía, y sus técnicas complementarias: las del criptoanálisis, que son aquellos métodos que se utilizan para romper textos cifrados con objeto de recuperar la información original.

La finalidad de la criptografía es en primer lugar, garantizar el secreto en la comunicación entre dos entidades (personas, organizaciones, etc.); en segundo lugar, asegurar que la información que se envía es auténtica en un doble sentido: que el remitente sea realmente quien dice ser; y por último impedir que el contenido del mensaje enviado (habitualmente denominado criptograma) sea modificado en su tránsito.

Otro método para ocultar el contenido de un mensaje es ocultar que ha habido un mensaje. Con esteganografía se puede ocultar un mensaje en un archivo de sonido, una imagen o incluso en reparto de los espacios usados para justificar un texto plano. Normalmente aparte de ocultarlo el mensaje además se cifra.

Criptografía simétrica

También conocida como “criptografía clásica” o de “llave privada”. Este tipo de criptografía es anterior al nacimiento de los ordenadores. Método criptográfico que usa una misma clave para cifrar y para descifrar mensajes. Las dos partes que se comunican han de ponerse de acuerdo de antemano sobre la clave a usar. Una vez ambas tienen acceso a esta clave, el remitente cifra un mensaje usándola, lo envía al destinatario, y éste lo descifra con la misma.

Un buen sistema de cifrado pone toda la seguridad en la clave y ninguna en el algoritmo. En otras palabras, no debería ser de ninguna ayuda para un atacante conocer el algoritmo que se está usando. Sólo si el atacante obtuviera la clave, le serviría conocer el algoritmo. Los algoritmos de cifrado usados por ejemplo en el sistema GNU, GnuPG tienen estas propiedades.

Dado que toda la seguridad está en la clave, es importante que sea muy difícil adivinar el tipo de clave. Esto quiere decir que el abanico de claves posibles, o sea, el espacio de posibilidades de claves, debe ser amplio. Richard Feynman fue famoso en Los Álamos por su habilidad para abrir cajas de seguridad. Para alimentar la leyenda que había en torno a él, llevaba encima un juego de herramientas que incluían un estetoscopio. En realidad, utilizaba una gran variedad de trucos para reducir a un pequeño número la cantidad de combinaciones que debía probar, y a partir de ahí simplemente probaba hasta que adivinaba la combinación correcta. En otras palabras, reducía el tamaño de posibilidades de claves.

Hoy por hoy, los ordenadores pueden adivinar claves con extrema rapidez, y ésta es la razón por la cual el tamaño de la clave es importante en los criptosistemas modernos. El algoritmo de cifrado DES usa una clave de 56 bits, lo que significa que hay 2^{56} claves posibles. 2^{56} son 72.057.594.037.927.936 claves. Esto representa un número muy alto de claves, pero una máquina computadora de uso general puede comprobar todo el espacio posible de claves en cuestión de días. Un máquina especializada lo puede hacer en horas. Por otra parte, algoritmos de cifrado de diseño más reciente como 3DES, Blowfish e IDEA usan todas claves de 128 bits, lo que significa que existen 2^{128} claves posibles. Esto representa muchas, muchísimas más claves, y aun en el caso de que todas las máquinas del planeta estuvieran cooperando, todavía tardarían más tiempo que la misma edad del universo en encontrar la clave.

Criptografía asimétrica

También conocida como “criptografía moderna” o de “llave pública”. Este tipo de criptografía se desarrolló en los años 70 y utiliza complicados algoritmos matemáticos relacionados con números primos y curvas elípticas. Método criptográfico que usa un par de claves para el envío de mensajes. Las dos claves pertenecen a la misma persona a la

que se ha enviado el mensaje. Una clave es pública y se puede entregar a cualquier persona. La otra clave es privada y el propietario debe guardarla de modo que nadie tenga acceso a ella. El remitente usa la clave pública del destinatario para cifrar el mensaje, y una vez cifrado, sólo la clave privada del destinatario podrá descifrar este mensaje.

Los sistemas de cifrado de clave pública o sistemas de cifrado asimétricos se inventaron con el fin de evitar por completo el problema del intercambio de claves de los sistemas de cifrado simétricos. Con las claves públicas no es necesario que el remitente y el destinatario se pongan de acuerdo en la clave a emplear. Todo lo que se requiere es que, antes de iniciar la comunicación secreta, el remitente consiga una copia de la clave pública del destinatario. Es más, esa misma clave pública puede ser usada por cualquiera que desee comunicarse con su propietario. Por tanto, se necesitarán sólo n pares de claves por cada n personas que deseen comunicarse entre sí

Criptografía híbrida

Método criptográfico que usa tanto un cifrado simétrico como uno asimétrico. Emplea el cifrado de clave pública para compartir una clave para el cifrado simétrico. El mensaje que se esté enviando en el momento, se cifra usando la clave y enviándolo al destinatario. Ya que compartir una clave simétrica no es seguro, la clave usada es diferente para cada sesión.

• Autentificación: código de acceso y confirmación de identidad (R, E, RP)

Autentificación

Proceso utilizado para confirmar la identidad y autenticidad de una persona o probar la integridad de información específica.

Código de acceso

Secuencia alfanumérica que debe marcarse para obtener un servicio, prestación o para alcanzar determinada zona de la red.

Llamamos autentificación a la comprobación de la identidad de una persona o de un objeto. Hemos visto hasta ahora diversos sistemas que pueden servir para la autentificación de servidores, de mensajes y de remitentes y destinatarios de mensajes. Pero hemos dejado pendiente un problema: las claves privadas suelen estar alojadas en máquinas clientes y cualquiera que tenga acceso a estas máquinas puede utilizar las claves que tenga instaladas y suplantar la identidad de su legítimo usuario.

Por tanto es necesario que los usuarios adopten medidas de seguridad y utilicen los medios de autentificación de usuario de los que disponen sus ordenadores personales. Hay tres sistemas de identificación de usuario, mediante contraseña, mediante dispositivo y mediante dispositivo biométrico.

La **autentificación mediante contraseña** es el sistema más común ya que viene incorporado en los sistemas operativos modernos de todos los ordenadores. Los ordenadores que estén preparados para la **autentificación mediante dispositivo** sólo reconocerán al usuario mientras mantenga introducida una “llave”, normalmente una tarjeta con chip. Hay sistemas de generación de claves asimétricas que introducen la clave privada en el chip de una tarjeta inteligente.

Los **dispositivos biométricos** son un caso especial del anterior, en los que la “llave” es una parte del cuerpo del usuario, huella dactilar, voz, pupila o iris. Existen ya en el mercado a precios relativamente económicos ratones que llevan incorporado un lector de huellas dactilares.

• Métodos de protección (firewall, proxy) (R, E, RP)

Firewall

Cortafuegos. Mecanismo de seguridad en Internet frente a accesos no autorizados. Básicamente consiste en un filtro que mira la identidad de los paquetes y rechaza todos aquellos que no estén autorizados o correctamente identificados.

Un firewall es simplemente un filtro que controla todas las comunicaciones que pasan de una red a la otra y en función de lo que sean permite o deniega su paso. Para permitir o denegar una comunicación el firewall examina el tipo de servicio al que corresponde, como pueden ser el Web, el correo o el IRC. Dependiendo del servicio el firewall decide si lo permite o no.

Además, el firewall examina si la comunicación es entrante o saliente y dependiendo de su dirección puede permitirla o no.

Un firewall puede ser un dispositivo software o hardware, es decir, un aparato que se conecta entre la red y el cable de la conexión a Internet, o bien un programa que se instala en la máquina que tiene el MODEM que conecta con Internet. Incluso podemos encontrar ordenadores computadores muy potentes y con software específico que lo único que hacen es monitorizar las comunicaciones entre redes.

Elementos a tener en cuenta por el usuario a la hora de querer adquirir un firewall.

- ✓ Internet es el principal punto de contacto con agentes malignos para el sistema. No únicamente virus, sino también de aplicaciones potencialmente peligrosas para el sistema
- ✓ El propósito del firewall es restringir el acceso al sistema a personas ajenas a él o a la red.
- ✓ Es garantía de que la conexión a Internet es segura.

¿Contra qué protege un firewall?

- ✓ Evita accesos no permitidos y ajenos al sistema, generalmente provenientes de direcciones IP inseguras y vulnerables.
- ✓ Bloquea el tráfico generado de fuera hacia adentro.
- ✓ Establecen un punto de inflexión de seguridad y de auditoria o de control, que puede ser configurado tanto para elementos externos al sistema, como para personas que trabajen desde dentro del sistema

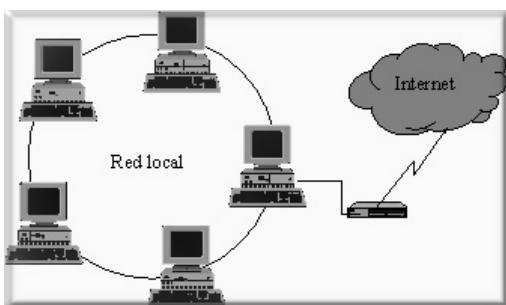
Proxy

Servidor de red que centraliza el tráfico entre Internet y una red privada, y que permite que los ordenadores de una red interna no estar conectados individualmente a la red. Además también impiden el acceso exterior a la red interna

¿Qué es un proxy?

Si tenemos una red local en la cual varios ordenadores están conectados entre sí. Uno de ellos tiene además una conexión a Internet, ya sea propia o a través de un proveedor de acceso.

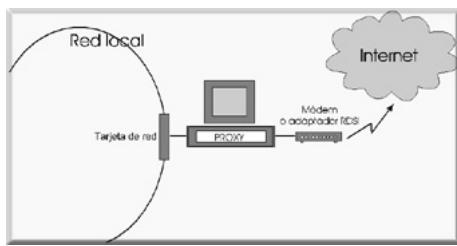
De este modo únicamente ese ordenador podría tener acceso a Internet. Evidentemente sería deseable que todos los demás ordenadores de la red local consigan acceder a Internet empleando esa única conexión.



Se puede utilizar un Router para dar salida a Internet, pero su costo es demasiado caro, por eso existe una posibilidad más simple que consiste en emplear un proxy, que es un programa que actúa como pasarela (gateway) entre redes, en el caso que nos ocupa entre la red local e Internet. Cada vez que alguno de los ordenadores de la red desea acceder a Internet realiza esta petición al proxy. El proxy utiliza la única conexión existente para enviar y recibir la información de Internet al ordenador que la solicita.

¿Cómo funciona un proxy?

Desde el punto de vista del usuario de la red local, el sistema funciona como si tuviera realmente un acceso directo a Internet. El usuario accede inmediatamente desde su ordenador a una página Web o recibe su correo electrónico, sin saber que el proxy existe.



En realidad, al abrir un programa como Internet Explorer o recoger el correo pendiente, la petición de servicio se realiza al proxy, no al servidor de Internet. El proxy es el encargado de redireccionar estas peticiones a la máquina correspondiente (el servidor de la página Web o el servidor de correo) y una vez recibida la información, de transmitirla al ordenador que la solicitó.

Ventajas de un proxy

- ✓ **Menor coste:** El programa y la instalación tienen un precio mucho menor que cualquier router.
- ✓ **Una sola línea telefónica:** Sólo es preciso disponer de una línea telefónica normal o RDSI
- ✓ **Fácil instalación:** La instalación emplea los dispositivos de la propia red local, por lo que se reduce la configuración de los programas.
- ✓ **Seguridad:** El proxy también actúa como una barrera (firewall) que limita el acceso a la red local desde el exterior.
- ✓ **Dirección IP única:** La dirección IP es la que identifica de forma única a cada máquina en Internet. Si se utiliza un proxy basta con una dirección IP para toda la red local en lugar de tener una IP para cada uno de los ordenadores.
- ✓ **Conección automática (autodialing):** No es necesario que el ordenador que actúa como proxy esté conectado permanentemente a Internet. Con esta función, cada vez que un usuario realiza una petición, el proxy establece la conexión. Del mismo modo el proxy la desconecta cuando no hay ninguna petición, todo ello automáticamente.
- ✓ **Menor tráfico de red:** El proxy almacena automáticamente en la memoria las páginas Web a las que se accede con mayor frecuencia, con lo que se reduce la cantidad de información que es necesario recuperar a través de Internet.

El proxy puede ofrecer a los ordenadores de la red local, todos los servicios disponibles en Internet incluyendo servicios avanzados de transmisión de audio y vídeo, Correo electrónico, World Wide Web, Transmisiones FTP, Telnet, News, Soporte del protocolo UDP: aplicaciones de streaming del tipo de Real Audio, IRC, Socks, DNS.

Todo esto se debe de instalar en la máquina Servidor y en los clientes, en la configuración de su navegador web, en el cual se debe de poner la dirección IP del servidor proxy y el puerto por el cual da salida a los servicios

SOFTWARE DE BASE

A. Traductores

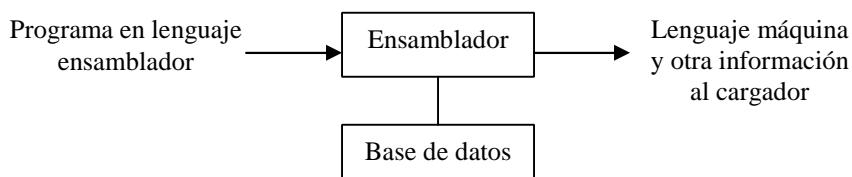
I. Traductores de bajo nivel

1. Ensambladores

Ensamblador.

Es un traductor que asigna direcciones absolutas a las variables simbólicas que el programador escogió, liberándolo de esa tarea.

Un ensamblado es un programa que acepta como entrada un programa en lenguaje ensamblador y produce su lenguaje máquina junto con información para el cargador. Sin embargo, el ensamblador debe además producir otra información para ser usada por el cargador. Por ejemplo, externamente define símbolos que deben ser anotados y aprobados por el cargador; el ensamblador no conoce la dirección (valor) de esos símbolos y el cargador se encarga de encontrar los programas conteniéndolos, cargándolos en el núcleo, y localizando los valores de esos símbolos en la llamada al programa.



• Tratamiento de operandos y modos de direccionamiento de la máquina objeto (RP)

Tratamiento de operandos

Las instrucciones de maquina operan sobre los datos, las categorías más importantes de datos son:

Numéricos

- ✓ Enteros o de punto fijo.
- ✓ Punto flotante.
- ✓ BCD (Decimal codificado ó binario).- Los datos BCD comprimidos presentan dos dígitos por byte y los no comprimidos presentan un dígito por byte.

Caracteres

Unidades de 8 bits que normalmente contienen representaciones ASCII de símbolo.

Datos lógicos

- ✓ Byte con o sin signo (8 bits).- El bit 7 es el bit de signo de un byte sin signo.
- ✓ Palabra con o sin signo (word-16 bits).- El bit 15 es el bit de signo de una word sin signo
- ✓ Doble palabra con o sin signo (dword-32 bits).- El bit 31 es el bit de signo de una dword sin signo
- ✓ Cuádruple palabra con o sin signo (qword-64 bits).- El bit 63 es el bit de signo de una qword sin signo

Direcciones

Son una forma de dato. En muchos casos, algún calculo se debe ejecutar sobre la referencia de un operando en una instrucción para determinar la dirección de la memoria principal o virtual. En este contexto, las direcciones se pueden considerar como enteros sin signo.

- ✓ Desplazamiento.- Son cantidades de 16 o 32 bits que contienen la distancia de la dirección base o la dirección referenciada.
- ✓ Punteros.- Consistentes en selectores de segmento de 16 bits y un desplazamiento de 16 o 32 bits.

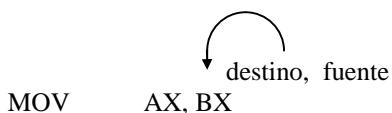
Estilo de programación

Un programa en ensamblador es una serie de instrucciones ejecutables que le dice al ensamblador que sentencia tiene que ejecutar. Cada sentencia esta compuesta por cuatro campos:

- Campo nombre.*- Se emplea con frecuencia como punto de entrada o regreso. Algunas veces es denominado campo del rotulo, asigna un nombre simbólico a la dirección del comienzo de memoria real de una instrucción de ensamblador. Debe comenzar con un carácter alfabético de 31 caracteres máximo.
- Campo operación.*- Contiene un mnemónico de 2 a 6 caracteres normalmente es una abreviatura en inglés, puede representar una instrucción maquina, una macroinstrucción, o una pseudo-operación.
- Campo operando.*- Contiene la posición o posiciones donde están los datos que van a ser manipulados por la instrucción de la operación, si la instrucción mide 1 o 2 operandos, están separados de la instrucción por lo menos de un espacio en blanco, si hay dos operandos estos están separados por una coma.
- Campo comentario.*- Se utiliza para documentar internamente el código, este debe comenzar con un punto y coma (;).

Formato general de una sentencia

[nombre] mnemónico [operando] [; comentario]



Operando destino.- La operación puede producir un resultado.

Operando fuente.- La operación puede involucrar uno o más operandos fuentes, esto es, operandos que son entradas para la operación.

Modos de direccionamiento.

Los procesadores permiten varias maneras de direccionar (llegar a) las celdas de memoria. La más simple de todas es el direccionamiento directo: basta con escribir la dirección de la celda deseada a continuación del código de la instrucción para que la unidad de control logre el acceso.

En otra forma, el direccionamiento inmediato, se usa un dato numérico que se escribe inmediatamente a la derecha de la instrucción que la requiere. La diferencia entre ambos tipos de operaciones consiste en que la primera usa la dirección donde está el dato, mientras que la segunda usa el dato mismo. Por ejemplo: *CARGA 22* mete al acumulador el valor contenido en la celda con dirección 22, mientras que *CARGA 22 (Inm)* mete el número 22 en el acumulador.

El direccionamiento indirecto no toma el número que está a la derecha de la instrucción como dirección para extraer de ella un valor, ni como dato inmediato, sino como una dirección de una celda a la que tendrá que ir para extraer otra dirección.

Por último, el direccionamiento indexado usa la dirección que está a la derecha de la instrucción para sumarla con el contenido de un registro especial de la CPU llamado índice. Esto permite la variación de direcciones para simular directamente el recorrido sobre los elementos de un vector.

Los modos de direccionamiento junto con sus formas en lenguaje ensamblador son:

MODO	FORMA	DIRECCIÓN
Absoluto	M	M
Registro	R	R
Indizado	c (R)	c + contenido (R)
Registro indirecto	*R	contenido (R)
Indizado indirecto	*c (R)	contenido (c + contenido (R))

Una posición de memoria M o un registro R se representa a sí mismo cuando se utiliza como fuente o como destino. Por ejemplo:

MOV R0, M

guarda el contenido del registro R0 en la posición de memoria M.

Un desplazamiento de dirección c desde el valor de registro R se escribe como c (R) Por tanto:

MOV 4 (R0), M

almacena el valor *contenido* ($4 + \text{contenido}(R0)$) en la posición de memoria M.

Las versiones indirectas de los dos últimos modos se indican mediante el prefijo *. Así,

MOV *4 (R0), M

Almacena el valor *contenido* (*contenido* ($4 + \text{contenido}(R0)$)) en la posición de memoria M.

Un modo final de direccionamiento permite que la fuente sea una constante:

MODO	FORMA	DIRECCIÓN
Literal	#c	c

Por tanto, la instrucción

MOV #1, R0

carga la constante 1 en el registro R0.

• **Ensambladores residentes y cruzados (E)**

Ensambladores cruzados.

Se denominan así los ensambladores que se utilizan en una computadora que posee un procesador diferente al que tendrán las computadoras donde va a ejecutarse el programa objeto producido.

Los ensambladores cruzados permiten a un programador desarrollar programas para diferentes sistemas sobre un computador.

Sin embargo, excepto en el caso de minicomputadores y grandes computadores que pueden ofrecer un simulador de microprocesador destinatario real, no se puede normalmente probar y depurar el código creado por un ensamblador cruzado sin ejecutarse sobre una maquina real que utilice este procesador. En cualquier caso, siempre se debe utilizar la sintaxis correcta, esto es, códigos OP, operandos, y así sucesivamente, para el microprocesador para el que el ensamblador cruzado está diseñado.

El empleo de este tipo de traductores permite aprovechar el soporte de medios físicos (discos, impresoras, pantallas, etc.), y de programación que ofrecen las máquinas potentes para desarrollar programas que luego los van a ejecutar sistemas muy especializados en determinados tipos de tareas.

Ensambladores residentes.

El polo opuesto del ensamblador cruzado es el ensamblador residente, que se ejecuta sobre una maquina que contiene el mismo procesador que el destinatario del código ensamblado. Un ensamblador residente ofrece al programador la ventaja de utilizar una única maquina para crear, probar, y depurar código.

Son aquellos que permanecen en la memoria principal de la computadora y cargan, para su ejecución, al programa objeto producido. Este tipo de ensamblador tiene la ventaja de que se puede comprobar inmediatamente el programa sin necesidad de transportarlo de un lugar a otro, como se hacía en cross-assembler, y sin necesidad de programas simuladores.

Sin embargo, puede presentar problemas de espacio de memoria, ya que el traductor ocupa espacio que no puede ser utilizado por el programador. Asimismo, también ocupará memoria el programa fuente y el programa objeto. Esto obliga a tener un espacio de memoria relativamente amplio. Es el indicado para desarrollos de pequeños sistemas de control y sencillos automatismos empleando microprocesadores.

La ventaja de estos ensambladores es que permiten ejecutar inmediatamente el programa; la desventaja es que deben mantenerse en la memoria principal tanto el ensamblador como el programa fuente y el programa objeto.

- **Ensamble condicional (RP)**

2. Macroprocesadores

Se podría pensar en dar al ensamblador la capacidad de repetir, por medio de una orden, grupos completos de instrucciones que deben aparecer en múltiples ocasiones. Esto es, compactar renglones repetitivos en uno solo que fungirá como su abreviatura, y pedir al ensamblador que lo expanda a la hora de la traducción. Este nuevo esquema recibe el nombre de macroprocesamiento, y es de importancia vital porque permite la sustitución textual de símbolos de un tipo con símbolos de otro. Esto forma parte de la idea central de la computación.

Una macro instrucción (abreviada como macro) es simplemente una conveniencia notacional para el programador. Una macro representa un grupo de sentencias comúnmente usados en el lenguaje fuente de programación. El macroprocesador reemplaza cada macro instrucción con el correspondiente grupo de sentencias de lenguaje fuente. Esto es llamado expandir las macros. Estas macro instrucciones permiten al programador escribir una versión corta del programa, y dejar los detalles mecánicos ser manejados por el macroprocesador.

La función de un macroprocesador esencialmente involucra la sustitución de un grupo de caracteres o líneas por otro. Excepto en unos pocos casos especializados, el macroprocesador no realiza el análisis del texto que maneja. El diseño y capacidades de un macroprocesador pueden ser influenciado por la forma de los enunciados del lenguaje de programación involucrado. Sin embargo, el significado de estos sentencias, y su traducción a lenguaje máquina, no son de interés durante la macro expansión. Esto significa que el diseño de un macroprocesador no está directamente relacionado a la arquitectura de la computadora sobre la cual esta corriendo.

El uso más común del macroprocesador es en programación en lenguaje ensamblador. Sin embargo, macroprocesadores pueden ser también usados con programación en lenguajes de alto nivel, lenguajes de comandos de sistemas operativos, etc. Además, hay macroprocesadores de propósito general que no están atados a ningún lenguaje en particular.

- **Bibliotecas de macros (E)**

Un macroprocesador trabaja con definiciones de renglones (o de símbolos) llamadas macros (o macrodefiniciones), que serán expandidas cuando se las llame. Una macrollamada, por tanto será, la invocación de una macrodefinición para producir nuevos renglones de texto.

Para convertir renglones en una macrodefinición, habrá que encerrarlos entre las palabras MACRO, seguida del nombre asignado, y FIN_MACRO. Cuando se requiera llamarla, solo habrá que nombrarla.

Un preprocesador puede permitir a un usuario definir macros, que son abreviaturas de construcciones más grandes.

Los procesadores de macros tratan dos clases de proposiciones: definición de macros y uso de macros. Las definiciones normalmente se indican con algún carácter exclusivo o palabra clave, como *define macro*. Constan de un nombre para la macro que esta definiendo y de un cuerpo, que constituye su definición, esto es, símbolos que se reemplazaran por valores (en este contexto, un valor es una cadena de caracteres). El uso de una macro consiste en dar un nombre a la macro y proporcionar parámetros reales, es decir, valores para sus parámetros formales. El procesador de macros sustituye los parámetros reales por los parámetros formales del cuerpo de la macro; después, el cuerpo transformado reemplaza el uso de la propia macro.

- **Expansión condicional (RP)**

Dos importantes pseudo-opciones de macroprocesador, AIF y AGO, permiten reordenamiento condicional de la secuencia de macro expansión. Estos permiten selección condicional de las instrucciones máquina que aparecen en una llamada a macro.

Considérese el siguiente programa:

```

        :
LOOP1 A      1, DATA1
        A      2, DATA2
        A      3, DATA3
        :
LOOP2 A      1, DATA3
        A      2, DATA2
        :
LOOP3 A      1, DATA1
        :
DATA1 DC    F'5'
DATA2 DC    F'10'
DATA3 DC    F'15'
        :

```

En este ejemplo, los operandos, etiquetas y el número de instrucciones generados cambian en cada secuencia. Este programa podría ser escrito como sigue:

```

        :
MACRO
&ARG0  VARY   &COUNT, &ARG1, &ARG2, &ARG3
&ARG0  A       1, &ARG1
        AIF    (&COUNT EQ 1) .FINI          Checa si & COUNT=1
        A       2, &ARG2
        AIF    (&COUNT EQ 2) .FINI          Checa si & COUNT=2
        A       3, &ARG3
.FINI   MEND
        :
LOOP1  VARY   3, DATA1, DATA2, DATA3
        :
        :
LOOP2  VARY   2, DATA3, DATA2
        :
        :
LOOP3  VARY   1, DATA1
        :
DATA1 DC    F'5'
DATA2 DC    F'10'
DATA3 DC    F'15'

```

Origen expandido

LOOP1 A 1, DATA1
A 2, DATA2
A 3, DATA3

LOOP2 A 1, DATA3
A 2, DATA2

LOOP3 A 1, DATA1

Las etiquetas que empiezan con un punto (.), tales como .FINI, son etiquetas macro y no aparecen en la salida del macroprocesador. La sentencia AIF (&COUNT EQ 1)FINI dirige al macroprocesador a saltar a la sentencia etiquetado .FINI si el parámetro correspondiente a &COUNT es 1; de otra manera, el

macroprocesador va a continuar con la sentencia siguiendo la pseudo-opción AIF.

AIF es una rama condicional; esta realiza una prueba aritmética solo si la condición probada es verdadera. AGO es una rama incondicional o sentencia “ir a”. Solo como instrucciones condicionales e incondicionales, AIF y AGO controlan la secuencia en la cual el macroprocesador expande sus sentencias en macro instrucciones.

II. Traductores de alto nivel

1. Intérpretes

Intérprete.

Cuando el traductor ejecuta inmediatamente el código obtenido recibe el nombre de intérprete. Un intérprete no genera código, sino que lo ejecuta tan pronto lo obtiene. Hay una gran diferencia de velocidad entre la ejecución de un programa (objeto) compilado y un programa (fuente) interpretado. La segunda siempre es más lenta que la primera, ya que el intérprete debe analizar, traducir y ejecutar cada instrucción, aún cuando esta forme parte de un ciclo de repeticiones.

En lugar de producir un programa objeto como resultado de una traducción, un intérprete realiza las operaciones que implica el programa fuente. Muchas veces los intérpretes se usan para ejecutar lenguajes de órdenes, pues cada operador que se ejecuta en un lenguaje de órdenes suele ser una invocación de una rutina compleja, como un editor o un compilador. Del mismo modo, algunos lenguajes de muy alto nivel, normalmente son interpretados, porque hay muchas cosas sobre los datos, como el tamaño y la forma de las matrices, que no se pueden deducir en el momento de la compilación.

Un intérprete procesa un programa fuente escrito en un lenguaje de alto nivel, tal como un compilador lo hace. La principal diferencia es que el intérprete ejecuta una versión del programa fuente directamente, en vez de traducirlo a código máquina.

Un intérprete usualmente realiza funciones de análisis léxico y sintáctico tal como el compilador, y luego traduce el programa fuente a una forma interna. Muchas diferentes formas internas pueden ser usadas. Hasta es posible usar el programa fuente original como una forma interna; sin embargo, generalmente es mucho más eficiente realizar algún preprocesamiento del programa antes de su ejecución.

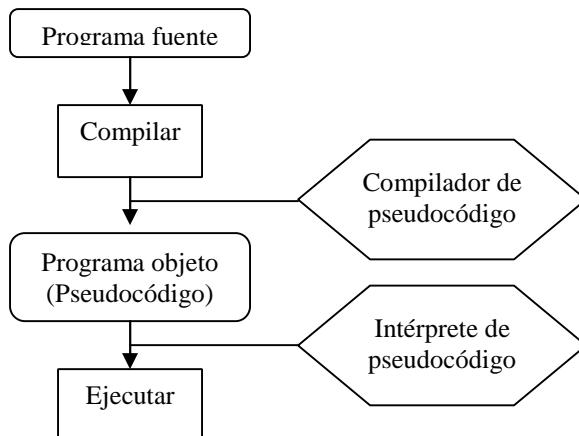
Después de traducir el programa fuente en su forma interna, el intérprete ejecuta las operaciones especificadas por el programa. Durante esta fase, un intérprete puede ser visto como un conjunto de subrutinas. La ejecución de estas subrutinas es controlada por la forma interna del programa.

El proceso de traducción de un programa fuente en alguna forma interna es más simple y rápido que compilarlo a un código máquina. Sin embargo, la ejecución del programa traducido por un intérprete es más lento que la ejecución del código objeto producido por un compilador. Así un intérprete de podría ser usado normalmente si la velocidad de ejecución es importante.

• Interpretación directa o mediante pseudocódigo (E, RP)

La importancia de la programación consiste en que este lenguaje funciona a la vez como vehículo descriptor y como modelo de la representación dada a la solución; las principales características de ese lenguaje son que es neutro y completo. El primer concepto denota su independencia respecto a alguna máquina en particular, y el segundo se refiere al poder del mismo para expresar cualquier idea computacional. Un lenguaje así recibe el nombre de pseudocódigo.

Un compilador de pseudocódigo es muy similar a un intérprete. En ambos casos, el programa fuente es analizado y convertido a una forma intermedia, la cual es luego ejecutada interpretativamente. Con un compilador de pseudocódigo, sin embargo, esta forma intermedia es el lenguaje máquina de una máquina hipotética, usualmente llamada pseudomáquina. El proceso de usar un compilador de pseudocódigo se ilustra:



El programa fuente es compilado, con el programa objeto resultante en pseudocódigo. Este programa en pseudocódigo es entonces leído y ejecutado bajo el control de un intérprete de pseudocódigo.

La ventaja principal de esto es la portabilidad del software. No es necesario para el compilador generar diferente código para diferentes computadoras, porque los programas objeto pseudocódigo pueden ser ejecutados en cualquier máquina que tenga un intérprete de pseudocódigo. Incluso el compilador por si mismo puede ser transportado si es escrito en el lenguaje que se compila. Para lograr esto, la versión fuente del compilador es compilado en pseudocódigo; este pseudocódigo puede entonces ser interpretado en otra computadora. De esta manera, un compilador de pseudocódigo puede ser usado sin modificación de una amplia variedad de sistemas si un intérprete de pseudocódigo es escrito en cada máquina. Además, escribir como un intérprete no es una tarea trivial, es más fácil que escribir un nuevo compilador para cada máquina.

• Lenguajes para aplicaciones específicas susceptibles o idóneos para interpretación (E)

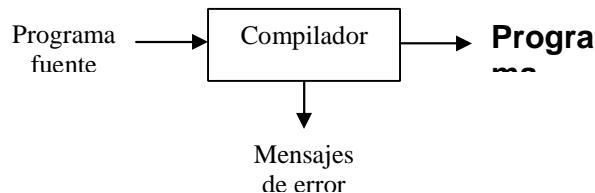
2. Compiladores

Compilador.

Traductor que forma parte del sistema y reside en el disco.

El compilador tiene que realizar análisis lexicográfico, análisis sintáctico y análisis semántico sobre las cadenas de entrada (programa fuente), para poder llegar a traducirlo al lenguaje de máquina o, por lo menos, al lenguaje ensamblador y obtener finalmente el mismo programa pero ya en lenguaje objeto.

Un compilador es un programa que lee un programa escrito en un lenguaje, el lenguaje fuente, y lo traduce a un programa equivalente en otro lenguaje, el lenguaje objeto. Como parte importante de este proceso de traducción, el compilador informa a su usuario de la presencia de errores en el programa fuente.



• Análisis lexicográfico (RP)

Consiste en reconocer todos y cada uno de los símbolos aislados que constituyen la frase; lo que, a su vez, implica reconocer las letras (y signos de puntuación) y reconocer las palabras. Reconocer no necesariamente significa

entender; para reconocer un símbolo lo único que se requiere es buscarlo (y encontrarlo) en un diccionario previamente especificado.

La tarea central de un analizador lexicográfico consiste en separar los componentes léxicos (o tokens) de entre el conjunto de símbolos de diversas clases (letras, dígitos, símbolos de puntuación, blancos y caracteres especiales) aunque sean invisibles, y es necesario aislar los componentes sintácticos de este conglomerado e caracteres. El modelo matemático de un analizador de este tipo recibe el nombre de autómata finito. Una autómata de este tipo es una función matemática que puede reconocer grupos de caracteres que constituyen un componente sintáctico.

En un compilador, el análisis lineal se llama análisis léxico o de exploración. Por ejemplo, en el análisis léxico los caracteres de la proposición asignada:

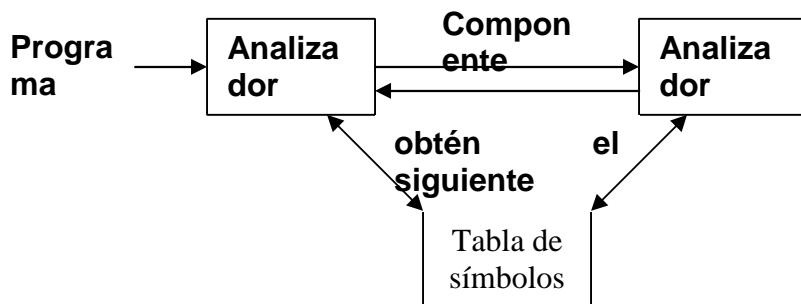
$$\text{posición} := \text{inicial} + \text{velocidad} * 60$$

se agruparían en los componentes léxicos siguientes:

1. El identificador *posición*
2. El símbolo de asignación *:=*
3. El identificador *inicial*
4. El signo de suma
5. El identificador *velocidad*
6. El signo de multiplicación
7. El número *60*

Los espacios en blanco que separan los caracteres de estos componentes léxicos normalmente se eliminan durante el análisis léxico.

Un analizador léxico es la primera fase de un compilador. Su principal función consiste en leer los caracteres de entrada y elaborar como salida una secuencia de componentes léxicos que utiliza el analizador sintáctico para hacer el análisis. Esta interacción suele aplicarse revirtiendo al analizador léxico en una subrutina o coroutines del analizador sintáctico. Recibida la orden “obtén el siguiente componente léxico” del analizador sintáctico, el analizador léxico lee los caracteres de entrada hasta que pueda identificar el siguiente componente léxico.



Como el analizador léxico es la parte del compilador que lee el texto fuente, también puede realizar ciertas funciones secundarias en la interfaz del usuario, como eliminar del programa fuente comentarios y espacios en blanco en forma de caracteres de espacio en blanco, caracteres TAB y de línea nueva. Otra función es relacionar los mensajes de error del compilador con el programa fuente. En algunos compiladores, el analizador léxico se encarga de hacer una copia del programa fuente en el que están marcados los mensajes de error. Si el lenguaje fuente es la base de algunas funciones de preprocessamiento de macros, entonces esas funciones del preprocesador también se pueden aplicar al hacer el análisis léxico.

En algunas ocasiones los analizadores léxicos se dividen en una cascada de dos fases; la primera, llamada examen y la segunda, análisis léxico. El examinador se encarga de realizar tareas sencillas, mientras que el analizar léxico es el que realiza las operaciones más complejas.

• Generación de código, códigos intermedios (RP)

La traducción o generación de código busca representar la frase original en términos de elementos de un lenguaje mucho más sencillo, que ya no está dotado de estructura. O sea, precisamente, traducir la frase fuente al lenguaje de máquina (o por lo menos al lenguaje ensamblador).

A la generación de código recién escrita se le conoce como generación de código intermedio, ya que no se trata de lenguaje máquina, sino de una especificación relativamente informal, en términos de elementos (temporales o no) que son producidos “a ciegas” por medio de reglas predefinidas.

Generación de código intermedio.

Después de los análisis sintáctico y semántico, algunos compiladores generan una representación intermedia explícita del programa fuente. Se puede considerar esta representación intermedia como un programa para una máquina abstracta. Esta representación intermedia debe tener dos propiedades importantes; debe ser fácil de producir y fácil de traducir al programa objeto.

La representación intermedia puede tener diversas formas. El código de tres direcciones consiste en una secuencia de instrucciones, cada una de las cuales tiene como máximo tres operandos. El programa fuente puede aparecer en código de tres direcciones como:

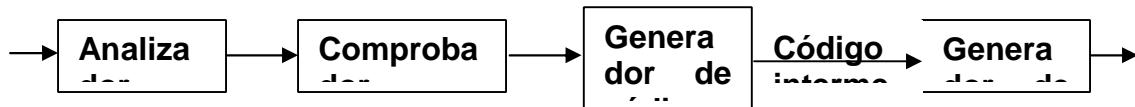
temp1 := entarea1(60)

```

temp2 := id3 * temp1
temp3 := id2 + temp2
id1 := temp3

```

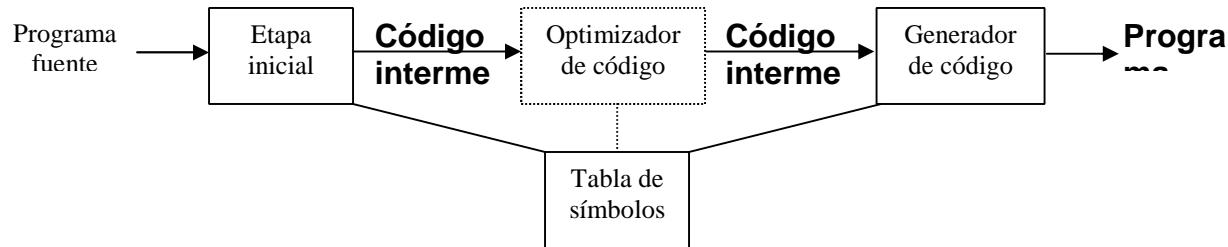
Esta representación intermedia tiene varias propiedades. Primera, cada instrucción de tres direcciones tiene a lo sumo un operador, además de la asignación. Por tanto, cuando se generan esas instrucciones, el compilador tiene que decidir el orden en que deben efectuarse las operaciones; segunda, el compilador debe generar un nombre temporal para guardar los valores calculados por cada instrucción; tercera, algunas instrucciones de tres direcciones tienen menos de tres operandos.



Generación de código.

La fase final de un compilador es la generación de código objeto, que por lo general consiste en código de máquina relocalizable o código ensamblador. Las posiciones de memoria se seleccionan para cada una de las variables usadas por el programa. Después, cada una de las instrucciones intermedias se traduce a una secuencia de instrucciones de máquina que ejecuta la misma tarea. Un aspecto decisivo es la asignación de variables a registros.

Toma como entrada una representación intermedia de un programa fuente y produce como salida un programa objeto equivalente. Las técnicas de generación de código se pueden utilizar con independencia de si se produce o no una fase de optimización antes de la generación de código, como en algunos compiladores, denominados de optimización. Dicha fase intenta transformar el código intermedio en una forma de la que se pueda producir código objeto más eficiente.



En tanto que los detalles dependen de la máquina objeto y del sistema operativo, aspectos como el manejo de la memoria, la selección de instrucciones, la asignación de registros y el orden de evaluación son inherentes en casi todos los problemas de generación de código.

- **Optimización de código (RP)**

Una operación importante que se puede hacer sobre el código intermedio recién generado es la optimización. Una vez generadas las líneas de código intermedio es posible “observarlas desde lejos” y tratar de eliminar redundancias y repeticiones. Esto no se puede hacer al tiempo de generación de código intermedio, porque equivaldría a quitar a las reglas predefinidas su carácter genérico.

El problema de convertir el código intermedio (quizá ya optimizado) en código objeto propiamente dicho, esto es, en lenguaje máquina, se le conoce como generación de código objeto.

La fase de optimización de código trata de mejorar el código intermedio, de modo que resulte un código de máquina más rápido de ejecutar. Algunas optimizaciones son triviales. Por ejemplo, un algoritmo natural genera el código intermedio:

temp1 := entarea1(60)

```
temp2 := id3 * temp1  
temp3 := id2 + temp2  
id1 := temp3
```

utilizando una instrucción para cada operador de la representación de árbol después del análisis semántico, aunque hay una forma mejor de realizar los mismos cálculos usando las dos instrucciones:

temp1 := id3 *60.0

```
id1 := id2 = temp1
```

Este sencillo algoritmo no tiene nada de malo, puesto que ese el problema se puede solucionar en la fase de optimización de código. Esto es, el compilador puede deducir que la conversión de 60 de entero a real se puede hacer de una vez por todas en el momento de la compilación, de modo que la operación *entarea1* se puede eliminar. Además *temp3* se usa solo una vez, para trasmitir su valor a *id1*. Entonces resulta seguro sustituir *id1* por *temp3*, a partir de lo cual la última proposición no se necesita y se obtiene el nuevo código.

Hay mucha variación en la cantidad de optimización de código que ejecutan los distintos compiladores. En los que hacen mucha optimización, llamados compiladores optimizadores, una parte significativa del tiempo del compilador se ocupa en esta fase. Sin embargo, hay optimizaciones sencillas que mejoran sensiblemente el tiempo de ejecución del programa objeto sin retardar demasiado la compilación.

- **Generadores de analizadores léxicos (E)**

Es un programa que recibe como entrada la especificación de la lexicografía de un lenguaje (cómo deben arreglarse sus componentes léxicos o tokens), y produce como salida un programa que hace el análisis.

Estas herramientas generan automáticamente analizadores léxicos, por lo general a partir de una especificación basada en expresiones regulares. La organización básica del analizador léxico resultante es en realidad un autómata finito.

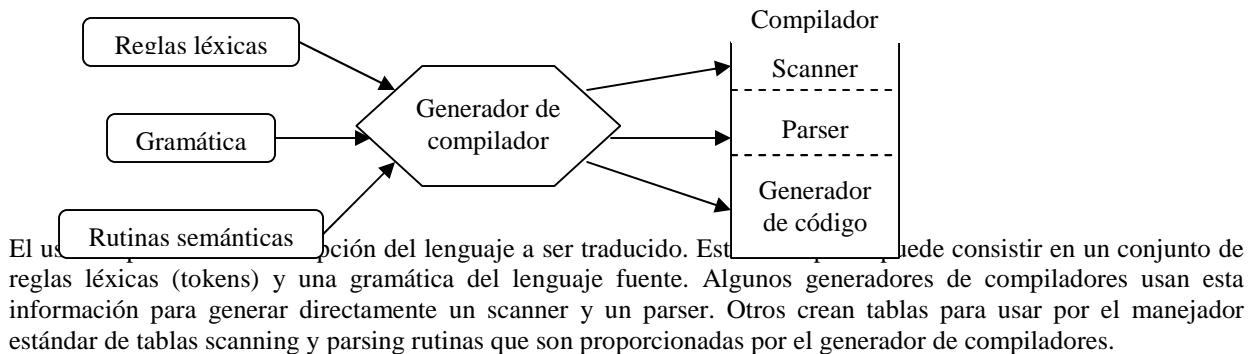
- **Generadores de compiladores (E)**

Aparecieron sistemas para ayudar en el proceso de escritura de compiladores, a menudo se hace referencia a estos como compiladores de compiladores, generadores de compiladores o sistemas generadores de traductores. Se orientan en torno a un modelo particular de lenguaje, y son más adecuados para generar compiladores de lenguajes similares al del modelo.

Estas herramientas generan automáticamente analizadores léxicos, por lo general a partir de una especificación basada en expresiones regulares. La organización básica del analizador léxico resultante es en realidad un autómata finito.

Un compilador de compiladores es un programa que recibe como entrada la gramática de un lenguaje de programación y produce como salida el analizador sintáctico para esa gramática.

Un generador de compiladores es una herramienta de software que ayuda en la tarea de construcción de un compilador. El proceso de usar un generador de compiladores se ilustra:



Los generadores de compiladores frecuentemente proveen lenguaje especial, notaciones, estructura de datos, y otras facilidades similares que pueden ser usadas en la escritura de rutinas semánticas.

La principal ventaja de usar un generador de compiladores es facilitar la construcción y prueba de compiladores. La cantidad de trabajo requerido por un usuario varía considerablemente de un generador de compiladores a otro dependiendo del grado de flexibilidad provisto. Compiladores que son generados de esta manera tienden a requerir mas memoria y compilan programas mas lentamente que los otros compiladores. Sin embargo, el código objeto generado por el compilador puede actualmente ser mejor cuando un generador de compilador es usado.

3. Tópicos de compilación

- **Tratamiento de recursividad (E, RP)**

Una definición es recursiva si dice como obtener conceptos nuevos empleando para tal fin el mismo concepto que se intenta definir. Este razonamiento parece fallido, parece que no llevara a ninguna parte por ser circular. En realidad los razonamientos recursivos se encuentran en la base misma de las matemáticas, porque son necesarios para describir conceptos centrales, como es el número.

El razonamiento recursivo tiene dos partes: la base y la regla recursiva de construcción. La base no es recursiva y es el punto tanto de partida como de terminación de la definición.

Un procedimiento es recursivo si puede comenzar una nueva activación antes de que haya terminado una activación anterior del mismo procedimiento.

En términos más formales, se dice que un lenguaje cuyas frases pueden ser generadas por un procedimiento es recursivamente enumerable, mientras que se le llama recursivo si existe un algoritmo para reconocerlo. Que un lenguaje sea recursivamente enumerable significa que se puede producir mediante una gramática; que sea recursivo significa que existe para él un algoritmo de reconocimiento: una máquina de Turing que llega a un estado final (un si o un no) cuando analiza las frases que lo componen.

- **Tratamiento de extensibilidad (RP)**

La extensibilidad es una propiedad importante de cualquier sistema operativo con la que sus diseñadores esperan mantenerse al día con los avances en la tecnología de cómputo.

- **Tratamiento de lenguajes orientados a objetos (definición de clase, herencia, instanciamiento) (RP)**

Los programas orientados a objetos se enfatizan en los datos, al contrario que la programación estructurada que se enfatiza en los algoritmos.

Clase.- Es una plantilla o modelo que define los datos y las funciones que actúan sobre esos datos (llamados métodos).

Objeto.- Elemento dato que es miembro de una clase.

Herencia.- Las clases se pueden subdividir en subclases, dicho de otro modo, una clase se puede deducir o derivar de otra clase. La clase original se denomina clase base y merced a la propiedad de herencia, una clase derivada o descendiente puede heredar las estructuras de datos y los métodos de su clase base. Además, la nueva clase puede añadir nuevos elementos datos y métodos a los que hereda de su clase base.

Instanciamiento.- Las clases son tipos de datos definidos por el usuario que definen un tipo de objeto. Un objeto es un modelo o instancia de este tipo o clase. De modo que un objeto es a una clase como una variable a un tipo. Por eso, a veces, el término objeto se utiliza indistintamente como instancia o modelo de una clase, y también como variable.

4. Ambientes integrados

• Depuración interactiva (E)

Permite la ejecución paso a paso ó por tramos del programa manteniendo el entorno que se va produciendo (valores de variables). El programador en cada parada de la ejecución de su programa puede comprobar e incluso modificar valores de las variables, con lo cual resulta muy interesante este tipo de ejecución para detección de errores y comprobación del buen funcionamiento del programa. Los puntos de parada en la ejecución son determinados por el propio programador mediante sentencias destinadas a tal fin.

Tipos de depuradores:

DUMP (Volcado de memoria).- Exhibición del contenido de la memoria. Cuando un programa tiene una terminación anormal, puede hacerse un volcado de la memoria para examinar el estado del programa en el momento del estallido, el programador mira dentro de los buffer para ver con qué elementos de datos estaba trabajando cuando falló. Los contadores comutadores, y señales del programa también pueden ser inspeccionados.

TRACE (Trazo).- Un trazo es un ayudante de la depuración consistiendo de una presentación que describe la crónica de acciones y resultados de los pasos individuales en un programa; la sección es algunas veces usada por un programa de control que produce esta clase de manifestación. El proceso de depuración precipita incontables problemas, cuyas correcciones requieren detalladamente registros de la ejecución del programa, un trazo es diseñado para proveer este tipo de información, tomando el programa del usuario y localizándolo bajo el control de una rutina especial, la cual monitorea el proceso del programa.

• Ayudas (E)

B. Sistemas operativos

I. Conceptos y estructuras básicas

1. Historia y evolución

• **Necesidad del sistema operativo, administración de recursos y desempeño del sistema (R)**

Un Sistema Operativo es un programa que actúa como intermediario entre el usuario y el hardware de un computador y su propósito es proporcionar un entorno en el cual el usuario pueda ejecutar programas. El objetivo principal de un Sistema Operativo es, entonces, lograr que el Sistema de computación se use de manera cómoda, y el objetivo secundario es que el hardware del computador se emplee de manera eficiente.

Un Sistema Operativo es una parte importante de cualquier sistema de computación. Un sistema de computación puede dividirse en cuatro componentes: el hardware, el Sistema Operativo, los programas de aplicación y los usuarios. El hardware (Unidad Central de Procesamiento(UCP), memoria y dispositivos de entrada/salida (E/S)) proporciona los recursos de computación básicos. Los programas de aplicación (compiladores, sistemas de bases de datos, juegos de video y programas para negocios) definen la forma en que estos recursos se emplean para resolver los problemas de computación de los usuarios.

Característica de un Sistema Operativo:

En general, se puede decir que un Sistema Operativo tiene las siguientes características:

- ✓ *Conveniencia.* Un Sistema Operativo hace más conveniente el uso de una computadora.
- ✓ *Eficiencia.* Un Sistema Operativo permite que los recursos de la computadora se usen de la manera más eficiente posible.
- ✓ *Habilidad para evolucionar.* Un Sistema Operativo deberá construirse de manera que permita el desarrollo, prueba o introducción efectiva de nuevas funciones del sistema sin interferir con el servicio.
- ✓ *Encargado de administrar el hardware.* El Sistema Operativo se encarga de manejar de una mejor manera los recursos de la computadora en cuanto a hardware se refiere, esto es, asignar a cada proceso una parte del procesador para poder compartir los recursos.
- ✓ *Relacionar dispositivos (gestionar a través del kernel).* El Sistema Operativo se debe encargar de comunicar a los dispositivos periféricos, cuando el usuario así lo requiera.
- ✓ *Organizar datos para acceso rápido y seguro.*
- ✓ *Manejar las comunicaciones en red.* El Sistema Operativo permite al usuario manejar con alta facilidad todo lo referente a la instalación y uso de las redes de computadoras.
- ✓ *Procesamiento por bytes de flujo a través del bus de datos.*
- ✓ *Facilitar las entradas y salidas.* Un Sistema Operativo debe hacerle fácil al usuario el acceso y manejo de los dispositivos de Entrada/Salida de la computadora.

Técnicas de Recuperación de Errores:

Evita que otros usuarios interfieran. El Sistema Operativo evita que los usuarios se bloqueen entre ellos, informándoles si esa aplicación está siendo ocupada por otro usuario.

Generación de Estadísticas:

Permite que se puedan compartir el hardware y los datos entre los usuarios.

El software de aplicación son programas que se utilizan para diseñar, tal como el procesador de palabras, lenguajes de programación, hojas de cálculo, etc.

El software de base sirve para interactuar el usuario con la máquina, son un conjunto de programas que facilitan el ambiente plataforma, y permite el diseño del mismo.

El Software de base está compuesto por :

- ✓ Cargadores.
- ✓ Compiladores.
- ✓ Ensambladores.
- ✓ Macros.

Funciones de un Sistema Operativo:

Las funciones más importantes que debe cumplir un sistema operativo son las siguientes:

- ✓ Aceptar los trabajos y conservarlos hasta su finalización.

- ✓ Detectar errores y actuar de modo apropiado en caso de que se produzcan.
- ✓ Controlar las operaciones de E/S.
- ✓ Controlar las interrupciones.
- ✓ Planificar la ejecución de tareas.
- ✓ Entregar recursos a las tareas.
- ✓ Retirar recursos de las tareas.
- ✓ Proteger la memoria contra el acceso indebido de los programas.
- ✓ Soportar el multiacceso.
- ✓ Proporcionar al usuario un sencillo manejo de todo el sistema.
- ✓ Aprovechar los tiempos muertos del procesador.
- ✓ Compartir los recursos de la máquina entre varios procesos al mismo tiempo.
- ✓ Administrar eficientemente el sistema de cómputo como un todo armónico.
- ✓ Permitir que los diferentes usuarios se comuniquen entre sí, así como protegerlos unos de otros.
- ✓ Permitir a los usuarios almacenar información durante plazos medianos o largos.
- ✓ Dar a los usuarios la facilidad de utilizar de manera sencilla todos los recursos, facilidades y lenguajes de que dispone la computadora.
- ✓ Administrar y organizar los recursos de que dispone una computadora para la mejor utilización de la misma, en beneficio del mayor número posible de usuarios.
- ✓ Controlar el acceso a los recursos de un sistema de computadoras.

Administracion de recursos.

Un computador es un conjunto de recursos para el traslado, almacenamiento y proceso de datos y para el control de estas funciones. El sistema operativo es el responsable de la gestión de estos recursos.

Administrando los recursos del computador, el sistema operativo tiene el control sobre las funciones básicas de la misma.

El sistema operativo funciona de la misma manera que el software normal de un computador, es decir, es un programa ejecutado por el procesador.

El sistema operativo abandona con frecuencia el control y debe depender del procesador para recuperarlo.

El sistema operativo es, nada mas que un sistema de computador. Como otros programas de computador, da instrucciones al procesador. La diferencia clave está en el propósito del programa. El sistema operativo dirige al procesador en el empleo de otros recursos del sistema y en el control del tiempo de ejecución de otros programas. Pero para que el procesador pueda hacer estas cosas, debe cesar la ejecución del programa del sistema operativo y ejecutar otros programas. Así pues, el sistema operativo cede el control al procesador para hacer algún trabajo “útil” y luego lo retoma durante el tiempo suficiente para preparar el procesador para llevar a cabo la siguiente parte del trabajo.

Una parte del sistema operativo está en la memoria principal. En esta parte está el núcleo (*kernel*), que incluye las funciones utilizadas con más frecuencia en el sistema operativo y, en un momento dado, puede incluir otras partes del sistema operativo que estén en uso. El resto de la memoria principal contiene datos y otros programas de usuario.

La asignación de este recurso (la memoria principal) es controlada conjuntamente por el sistema operativo y por el hardware de gestión de memoria en el procesador. El sistema operativo decide cuándo puede utilizarse un dispositivo de E/S por parte de un programa en ejecución y controla el acceso a la utilización de los archivos. El procesador es, en sí mismo, un recurso y es el sistema operativo el que debe determinar cuánto tiempo del procesador debe dedicarse a la ejecución de un programa de usuario en particular. En el caso de sistemas multiprocesador, la decisión debe distribuirse entre todos los procesadores.

Desempeño del sistema.

Un proceso es básicamente un programa en ejecución. Cada proceso tiene asociado un espacio de direcciones, una lista de posiciones de memoria desde algún mínimo (usualmente 0) hasta algún máximo, que el proceso puede leer y escribir. A cada proceso también se asocia un conjunto de registros, que incluyen el contador del programa, el apuntador de pila, y otros registros de hardware, así como toda la demás información necesaria para ejecutar el programa.

Periódicamente el sistema operativo decide dejar de ejecutar un proceso y comenzar a ejecutar otro, por ejemplo, porque el primero ya tuvo más tiempo de CPU del que le tocaba durante el segundo anterior.

Cuando un proceso se suspende temporalmente de esta manera, debe reiniciarse después en el mismo estado exactamente en que estaba en el momento en que se le detuvo. Esto implica que toda la información acerca del proceso se debe guardar explícitamente en algún lugar durante la suspensión. Cuando un proceso se suspende temporalmente, es necesario guardar todos estos apuntadores para que una llamada READ ejecutada después de reiniciarse el proceso lea los datos correctos. En muchos sistemas operativos, toda la información acerca de cada proceso, aparte del contenido de su propio espacio de direcciones, se almacena en una tabla del sistema operativo llamada tabla de procesos, que es un arreglo (o lista enlazada) de estructuras, una para cada proceso existente en ese momento.

Así, un proceso (suspendido) consiste en un espacio de direcciones, llamado imagen de núcleo y su entrada en la tabla de procesos, que contiene sus registros, entre otras cosas.

Las llamadas al sistema de administración para procesos clave son las que se ocupan de la creación y terminación de procesos.

Si un proceso puede crear uno o más procesos distintos (denominados procesos hijos) y éstos a su vez pueden crear procesos hijos, pronto llegamos a la estructura de árbol de procesos. Los procesos relacionados que están cooperando para realizar alguna tarea a menudo necesitan comunicarse entre sí y sincronizar sus actividades. Esta comunicación se llama comunicación entre procesos.

Un proceso que se comunica con otro proceso en una computadora distinta lo hace enviando mensajes por una red. A fin de prevenir la posibilidad de que un mensaje o su respuesta se pierda, el remitente puede solicitar que su propio sistema operativo le notifique cuando haya transcurrido cierto número de segundos, a fin de poder retransmitir el mensaje si todavía no ha llegado un acuse de recibo. Después de establecer este temporizador, el programa puede seguir realizando otros trabajos.

Cuando ha transcurrido el número de segundos que se especificó, el sistema operativo envía una señal al proceso. La señal hace que el proceso suspenda temporalmente lo que estaba haciendo, guarde sus registros en la pila, y comience a ejecutar un procedimiento especial de manejo de señales. Una vez que el manejador de señales termina, el proceso en ejecución se reinicia en el estado en que estaba justo antes de la señal.

• Generaciones de sistemas operativos (R)

Generación Cero (década de 1940).

Los primeros sistemas computacionales no poseían sistemas operativos. Los usuarios tenían completo acceso al lenguaje de la máquina. Todas las instrucciones eran codificadas a mano.

Primera Generación (década de 1950).

Los sistemas operativos de los años cincuenta fueron diseñados para hacer mas fluida la transición entre trabajos. Antes de que los sistemas fueran diseñados, se perdía un tiempo considerable entre la terminación de un trabajo y el inicio del siguiente. Este fue el comienzo de los sistemas de procesamiento por lotes, donde los trabajos se reunían por grupos o lotes. Cuando el trabajo estaba en ejecución, este tenía control total de la máquina. Al terminar cada trabajo, el control era devuelto al sistema operativo, el cual limpiaba y leía e iniciaba el trabajo siguiente.

Al inicio de los 50's esto había mejorado un poco con la introducción de tarjetas perforadas (las cuales servían para introducir los programas de lenguajes de máquina), puesto que ya no había necesidad de utilizar los tableros enchufables.

Además el laboratorio de investigación General Motors implementó el primer sistema operativo para la IBM 701. Los sistemas de los 50's generalmente ejecutaban una sola tarea, y la transición entre tareas se suavizaba para lograr la máxima utilización del sistema. Esto se conoce como sistemas de procesamiento por lotes de un sólo flujo, ya que los programas y los datos eran sometidos en grupos o lotes.

La introducción del transistor a mediados de los 50's cambió la imagen radicalmente.

Se crearon máquinas suficientemente confiables las cuales se instalaban en lugares especialmente acondicionados, aunque sólo las grandes universidades y las grandes corporaciones o bien las oficinas del gobierno se podían dar el lujo de tenerlas.

Para poder correr un trabajo (programa), tenían que escribirlo en papel (en Fortran o en lenguaje ensamblador) y después se perforaría en tarjetas. Enseguida se llevaría la pila de tarjetas al cuarto de introducción al sistema y la entregaría a uno de los operadores. Cuando la computadora terminara el trabajo, un operador se dirigiría a la impresora y desprendería la salida y la llevaría al cuarto de salida, para que la recogiera el programador.

Segunda Generación (a mitad de la década de 1960).

La característica de los sistemas operativos fue el desarrollo de los sistemas compartidos con multiprogramación, y los principios del multiprocesamiento. En los sistemas de multiprogramación, varios programas de usuario se encuentran al mismo tiempo en el almacenamiento principal, y el procesador se cambia rápidamente de un trabajo a otro. En los sistemas de multiprocesamiento se utilizan varios procesadores en un solo sistema computacional, con la finalidad de incrementar el poder de procesamiento de la maquina.

La independencia de dispositivos aparece después. Un usuario que desea escribir datos en una cinta en sistemas de la primera generación tenía que hacer referencia específica a una unidad de cinta particular. En la segunda generación, el programa del usuario especificaba tan solo que un archivo iba a ser escrito en una unidad de cinta con cierto numero de pistas y cierta densidad.

Se desarrolló sistemas compartidos, en la que los usuarios podían acoplarse directamente con el computador a través de terminales. Surgieron sistemas de tiempo real, en que los computadores fueron utilizados en el control de procesos industriales. Los sistemas de tiempo real se caracterizan por proveer una respuesta inmediata.

Tercera Generación (mitad de década 1960 a mitad década de 1970).

Se inicia en 1964, con la introducción de la familia de computadores Sistema/360 de IBM. Los computadores de esta generación fueron diseñados como sistemas para usos generales . Casi siempre eran sistemas grandes, voluminosos, con el propósito de serlo todo para toda la gente. Eran sistemas de modos múltiples, algunos de ellos soportaban simultáneamente procesos por lotes, tiempo compartido, procesamiento de tiempo real y multiprocesamiento.

Eran grandes y costosos, nunca antes se había construido algo similar, y muchos de los esfuerzos de desarrollo terminaron muy por arriba del presupuesto y mucho después de lo que el planificador marcaba como fecha de terminación.

Estos sistemas introdujeron mayor complejidad a los ambientes computacionales; una complejidad a la cual, en un principio, no estaban acostumbrados los usuarios.

Cuarta Generación (mitad de década de 1970 en adelante).

Los sistemas de la cuarta generación constituyen el estado actual de la tecnología. Muchos diseñadores y usuarios se sienten aun incómodos, después de sus experiencias con los sistemas operativos de la tercera generación.

Con la ampliación del uso de redes de computadores y del procesamiento en línea los usuarios obtienen acceso a computadores alejados geográficamente a través de varios tipos de terminales.

Los sistemas de seguridad se ha incrementado mucho ahora que la información pasa a través de varios tipos vulnerables de líneas de comunicación. La clave de cifrado esta recibiendo mucha atención; han sido necesario codificar los datos personales o de gran intimidad para que; aun si los datos son expuestos, no sean de utilidad a nadie mas que a los receptores adecuados.

El porcentaje de la población que tiene acceso a un computador en la década de los ochenta es mucho mayor que nunca y aumenta rápidamente.

El concepto de maquinas virtuales es utilizado. El usuario ya no se encuentra interesado en los detalles físicos de; sistema de computación que esta siendo accedita. En su lugar, el usuario ve un panorama llamado maquina virtual creado por el sistema operativo.

Los sistemas de bases de datos han adquirido gran importancia. Nuestro mundo es una sociedad orientada hacia la información, y el trabajo de las bases de datos es hacer que esta información sea conveniente accesible de una manera controlada para aquellos que tienen derechos de acceso.

2. Esquema básico

• Estructura interna: monolítico y modular (E)

Los sistemas modulares son también llamados sistemas monolíticos. Este tipo de organización es con mucho la más común; bien podría recibir el subtítulo de "el gran embrollo". La estructura consiste en que no existe estructura alguna. El sistema operativo se escribe como una colección de procedimientos, cada uno de los cuales puede llamar a los demás cada vez que así lo requiera. Cuando se usa esta técnica, cada procedimiento del sistema tiene una interfaz bien definida en términos de parámetros y resultados y cada uno de ellos es libre de llamar a cualquier otro, si este último proporciona cierto cálculo útil para el primero. Sin embargo incluso en este tipo de sistemas es posible tener al menos algo de estructura. Los servicios (llamadas al sistema) que proporciona el sistema operativo se solicitan colocando los parámetros en lugares bien definidos, como en los registros o en la pila, para después ejecutar una instrucción especial de trampa de nombre "llamada al núcleo" o "llamada al supervisor".

Esta instrucción cambia la máquina del modo usuario al modo núcleo y transfiere el control al sistema operativo. El sistema operativo examina entonces los parámetros de la llamada, para determinar cuál de ellas se desea realizar. A continuación, el sistema operativo analiza una tabla que contiene en la entrada k un apuntador al procedimiento que realiza la k -ésima llamada al sistema. Esta operación identifica el procedimiento de servicio, al cual se llama. Por último, la llamada al sistema termina y el control regresa al programa del usuario.

Esta organización sugiere una organización básica del sistema operativo:

1. Un programa principal que llama al procedimiento del servicio solicitado.
2. Un conjunto de procedimientos de servicio que llevan a cabo las llamadas al sistema.
3. Un conjunto de procedimientos utilitarios que ayudan al procedimiento de servicio.

En este modelo, para cada llamada al sistema existe un procedimiento de servicio que se encarga de él. Los procedimientos utilitarios hacen cosas necesarias para varios procedimientos de servicio, por ejemplo buscar los datos de los programas del usuario.

Las características fundamentales de este tipo de estructura son:

- ✓ Construcción del programa final a base de módulos compilados separadamente que se unen a través del ligador.
- ✓ Buena definición de parámetros de enlace entre las distintas rutinas existentes, que puede provocar mucho acoplamiento.
- ✓ Carecen de protecciones y privilegios al entrar a rutinas que manejan diferentes aspectos de los recursos de la computadora, como memoria, disco, etc.

Generalmente están hechos a medida, por lo que son eficientes y rápidos en su ejecución y gestión, pero por lo mismo carecen de flexibilidad para soportar diferentes ambientes de trabajo o tipos de aplicaciones.

• Tipos de sistemas: monousuario, multiusuario, distribuido, de red, de tiempo real, de propósito especial (E)

Sistema operativo monousuario.

Los sistemas operativos monousuarios son aquéllos que soportan a un usuario a la vez, sin importar el número de procesadores que tenga la computadora o el número de procesos o tareas que el usuario pueda ejecutar en un mismo instante de tiempo. Las computadoras personales típicamente se han clasificado en este renglón.

En otras palabras los sistemas monousuarios son aquellos que nada más puede atender a un solo usuario, gracias a las limitaciones creadas por el hardware, los programas o el tipo de aplicación que se esté ejecutando.

Sistema operativo multiusuario.

Los sistemas operativos multiusuarios son capaces de dar servicio a más de un usuario a la vez, ya sea por medio de varias terminales conectadas a la computadora o por medio de sesiones remotas en una red de comunicaciones.

No importa el número de procesadores en la máquina ni el número de procesos que cada usuario puede ejecutar simultáneamente.

En esta categoría se encuentran todos los sistemas que cumplen simultáneamente las necesidades de dos o más usuarios, que comparten mismos recursos. Este tipo de sistemas se emplean especialmente en redes. En otras palabras consiste en el fraccionamiento del tiempo (timesharing).

Sistema operativo distribuido.

Los sistemas operativos distribuidos abarcan los servicios de los de red, logrando integrar recursos (impresoras, unidades de respaldo, memoria, procesos, unidades centrales de proceso) en una sola máquina virtual que el usuario accesa en forma transparente. Es decir, ahora el usuario ya no necesita saber la ubicación de los recursos, sino que los conoce por nombre y simplemente los usa como si todos ellos fuesen locales a su lugar de trabajo habitual.

Las razones para crear o adoptar sistemas distribuidos se dan por dos razones principales: por necesidad (debido a que los problemas a resolver son inherentemente distribuidos) o porque se desea tener más confiabilidad y disponibilidad de recursos.

Los sistemas distribuidos deben de ser muy confiables, ya que si un componente del sistema se compone otro componente debe de ser capaz de reemplazarlo.

Entre los diferentes Sistemas Operativos distribuidos que existen tenemos los siguientes: Sprite, Solaris-MC, Mach, Chorus, Spring, Amoeba, Taos, etc.

Características de los Sistemas Operativos distribuidos:

- ✓ Colección de sistemas autónomos capaces de comunicación y cooperación mediante interconexiones hardware y software .
- ✓ Gobierna operación de un S.C. y proporciona abstracción de máquina virtual a los usuarios.
- ✓ Objetivo clave es la transparencia.
- ✓ Generalmente proporcionan medios para la compartición global de recursos.
- ✓ Servicios añadidos: denominación global, sistemas de archivos distribuidos, facilidades para distribución de cálculos (a través de comunicación de procesos internodos, llamadas a procedimientos remotos, etc.).

Sistema operativo de red.

Los sistemas operativos de red se definen como aquellos que tiene la capacidad de interactuar con sistemas operativos en otras computadoras por medio de un medio de transmisión con el objeto de intercambiar información, transferir archivos, ejecutar comandos remotos y un sin fin de otras actividades.

El punto crucial de estos sistemas es que el usuario debe saber la sintaxis de un conjunto de comandos o llamadas al sistema para ejecutar estas operaciones, además de la ubicación de los recursos que deseé accesar.

El primer Sistema Operativo de red estaba enfocado a equipos con un procesador Motorola 68000, pasando posteriormente a procesadores Intel como Novell Netware.

Los Sistemas Operativos de red mas ampliamente usados son: Novell Netware, Personal Netware, LAN Manager, Windows NT Server, UNIX, LANtastic.

Sistema operativo de tiempo real.

Los Sistemas Operativos de tiempo real son aquellos en los cuales no tiene importancia el usuario, sino los procesos. Por lo general, están subutilizados sus recursos con la finalidad de prestar atención a los procesos en el momento que lo requieran. se utilizan en entornos donde son procesados un gran número de sucesos o eventos.

Estos sistemas tienen como objetivo proporcionar tiempos más rápidos de respuesta, procesar la información sin tiempos muertos. En estos sistemas el administrador de memoria es relativamente menos solicitado debido a que muchos procesos residen permanentemente en memoria. El administrador de archivos se encuentra normalmente en grandes sistemas de tiempo real y su objetivo principal es manejar la velocidad de acceso, más que la utilización eficaz del almacenamiento secundario.

Los Sistemas Operativos de tiempo real, cuentan con las siguientes características:

- ✓ Se dan en entornos en donde deben ser aceptados y procesados gran cantidad de sucesos, la mayoría externos al sistema computacional, en breve tiempo o dentro de ciertos plazos.
- ✓ Se utilizan en control industrial, conmutación telefónica, control de vuelo, simulaciones en tiempo real., aplicaciones militares, etc.
- ✓ Objetivo es proporcionar rápidos tiempos de respuesta.
- ✓ Procesa ráfagas de miles de interrupciones por segundo sin perder un solo suceso.
- ✓ Proceso se activa tras ocurrencia de suceso, mediante interrupción.
- ✓ Proceso de mayor prioridad expropia recursos.
- ✓ Por tanto generalmente se utiliza planificación expropiativa basada en prioridades.
- ✓ Gestión de memoria menos exigente que tiempo compartido, usualmente procesos son residentes permanentes en memoria.
- ✓ Población de procesos estática en gran medida.
- ✓ Poco movimiento de programas entre almacenamiento secundario y memoria.
- ✓ Gestión de archivos se orienta más a velocidad de acceso que a utilización eficiente del recurso.

Sistema operativo de propósito especial.

Algunos de los Sistemas de Propósito Especial son los siguientes:

Sistema Operativo De Tiempo Real:

Se usa generalmente como un dispositivo de control de una aplicación dedicada en el que el procesamiento debe realizarse dentro de un tiempo dado. Las principales características que deben brindar son:

1. Garantizar la respuesta a eventos externos dentro de límites de tiempo reestablecidos.
2. Los parámetros más importantes son el tiempo de procesamiento de la entrada y un rápido almacenamiento de la misma.
3. Existen dos tipos de Sistemas de tiempo real: En los que el tiempo de respuesta no es muy critico, y en los que lo es.
4. El sistema de tiempo real incluye software y hardware compuesto por:

Sensores: Elementos que detectan mediante la alteración de sus características un cambio en el ambiente que miden.

Traductores: Traducen un cambio físico en una corriente o tensión eléctrica.

Conversores: Convierten las variaciones de corriente o tensión en pulsos binarios.

Interfaces: Puertos, registradores, alarmas, consolas, etc.

Procesadores: Son los responsables de las tareas de control, interrupciones, tiempos de almacenamiento, etc.

Un caso particular de estos sistemas es el del Sistema de Control de Procesos.

Sistema Operativo Con Tolerancia A Fallas:

Usado en aplicaciones donde se debe proveer un servicio continuo o cuyo mantenimiento es dificultoso o muy costoso. Se suele utilizar un conjunto de redundancias e los recursos y chequeos internos.

El sistema operativo detecta y corrige errores; y recupera el sistema habilitando reemplazos de los componentes en mal funcionamiento o vuelve atrás operaciones que motivaron perdida de datos.

Sistema Operativo Virtual:

Se trata de un tipo de sistemas operativos que presentan una interface a cada proceso, mostrando una máquina que parece idéntica a la máquina real subyacente. Estos sistemas operativos separan dos conceptos que suelen estar unidos en el resto de sistemas: la multiprogramación y la máquina extendida. El objetivo de los sistemas operativos de máquina virtual es el de integrar distintos sistemas operativos dando la sensación de ser varias máquinas diferentes.

El núcleo de estos sistemas operativos se denomina monitor virtual y tiene como misión llevar a cabo la multiprogramación, presentando a los niveles superiores tantas máquinas virtuales como se soliciten. Estas

máquinas virtuales no son máquinas extendidas, sino una réplica de la máquina real, de manera que en cada una de ellas se pueda ejecutar un sistema operativo diferente, que será el que ofrezca la máquina extendida al usuario.

• **Lenguajes de control (*shell*), interfaces gráficas y utilitarios de un sistema operativo (E)**

Lenguajes de control (*shell*):

Es la forma de interfaz entre el sistema operativo y el usuario en la que este escribe los comandos utilizando un lenguaje de comandos especial. Los sistemas con interfaces de líneas de comandos se consideran más difíciles de aprender y utilizar que los de las interfaces gráficas. Sin embargo, los sistemas basados en comandos son por lo general programables, lo que les otorga una flexibilidad que no tienen los sistemas basados en gráficos carentes de una interfaz de programación.

El Shell es el intérprete de comandos, a pesar de no ser parte del sistema operativo, hace un uso intenso de muchas características del sistema operativo y por tanto sirve como un buen ejemplo de la forma en que se pueden utilizar las llamadas al sistema. También es la interfaz primaria entre un usuario situado frente a su terminal y el sistema operativo.

Cuando algún usuario entra al sistema, un "shell" se inicia. El shell tiene la terminal como entrada y como salida estándar. Este da inicio al teclear solicitud de entrada, carácter como un signo de pesos, el cual indica al usuario que el shell está esperando un comando. En MS-DOS normalmente aparece la letra de la unidad, seguida por dos puntos (:), el nombre del directorio en que se encuentra y por último el signo de "mayor que" (>). Esto es: C:>

Interfaces graficas:

Es el tipo de visualización que permite al usuario elegir comandos, iniciar programas y ver listas de archivos y otras opciones utilizando las representaciones visuales (iconos) y las listas de elementos del menú. Las selecciones pueden activarse bien a través del teclado o con el mouse.

Para los autores de aplicaciones, las interfaces gráficas de usuario ofrecen un entorno que se encarga de la comunicación con el ordenador o computadora. Esto hace que el programador pueda concentrarse en la funcionalidad, ya que no está sujeto a los detalles de la visualización ni a la entrada a través del mouse o el teclado. También permite a los programadores crear programas que realicen de la misma forma las tareas más frecuentes, como guardar un archivo, porque la interfaz proporciona mecanismos estándar de control como ventanas y cuadros de diálogo. Otra ventaja es que las aplicaciones escritas para una interfaz gráfica de usuario son independientes de los dispositivos: a medida que la interfaz cambia para permitir el uso de nuevos dispositivos de entrada y salida, como un monitor de pantalla grande o un dispositivo óptico de almacenamiento, las aplicaciones pueden utilizarlos sin necesidad de cambios.

Utilitarios de un sistema operativo:

Administración de Archivos:

Hacen más fácil el manejo de los archivos. En los grandes días del DOS no costaba mucho mejorar los métodos para escribir, solamente texto que proveía el DOS. Muchos programas fueron escritos para ayudar al usuario a encontrar los archivos, crear y organizar directorios, copiar, mover y renombrar archivos.

Algunos hasta usan el ratón para apuntar y clickear para cumplir esas tareas. Se aprecia la amplitud de la mejora hasta hacer las mismas tareas desde la línea de comandos. La nueva interfase gráfica que viene con los sistemas operativos como el Windows 95/98, ha reducido la necesidad de programas alternativos para la administración de archivos.

Administración de Disco:

Involucran el formateo y la defragmentación de los discos. Defragmentar significa poner los archivos en el disco de manera que todo el conjunto quede en secuencia. Esto reduce el tiempo de acceso a los archivos. Algunos programas para administración del disco le permiten hasta especificar que ciertos archivos a lo que se accede frecuentemente, como el mismo sistema operativo y otros programas de uso constante, se hallen en la parte de adelante del disco. Todo lo que acelere las cosas tendrá sus clientes.

Administración de Memoria:

Un software que maneja los programas de la RAM, (se colocan los datos). Mueven ciertos ítems residentes en la memoria y los sacan de su camino. Esto puede aumentar con efectividad la memoria disponible reuniendo todas las porciones no utilizadas en un solo lugar, haciendo así que su cantidad resulte considerable.

Backup:

Un programa de Backup, que también restablece los datos resguardados es imprescindible, si se cuenta con algunos datos que se desea tener a mano por un momento. El software comprimirá esos datos para ocupar el menor espacio posible en la memoria. (Recuerde el problema del espacio escaso, que ya se vió antes en: Almacenamiento: Formateo de Discos)

Recuperación de Datos:

Tratarán de recuperar archivos borrados o dañados (corruptos).

Compresión de Datos:

Reducen el espacio libre generado por los esquemas de formateo, como se vio en Almacenamiento: Formateo de Discos.

Anti-virus:

Son otros de los programas que deberían tenerse. Estos monitorean al computador detectando la actividad de los virus que son pequeños y horribles programas que se copian a sí mismos en otros discos para difundirse hacia otras computadoras. Los virus pueden ser molestos o ampliamente destructivos de nuestros archivos.

• **Llamadas al sistema (E)**

La interfaz entre el sistema operativo y los programas del usuario se define por medio del conjunto de "instrucciones extendidas" que el sistema operativo proporciona. Estas instrucciones extendidas se conocen como llamadas al sistema. Las llamadas al sistema varían de un sistema operativo a otro (aunque los conceptos fundamentales tienden a ser análogos).

Las llamadas al sistema se clasifican normalmente en dos categorías generales: aquellas que se relacionan con procesos y las que lo hacen con el sistema de archivo.

Por Procesos

Un proceso es básicamente un programa en ejecución. Consta del programa ejecutable y la pila o stack del programa, su contador de programa, apuntador de pila y otros registros, así como la otra información que se necesita para ejecutar el programa. En si el proceso en el concepto de los sistemas operativos es como el sistema de tiempo compartido. Esto es, que en forma periódica, el sistema operativo decide suspender la ejecución de un proceso y dar inicio a la ejecución de otro, por ejemplo, porque el primero haya tomado ya más de su parte del tiempo de la CPU, en terrenos del segundo.

Cuando un proceso se suspende temporalmente, debe reiniciarse después exactamente en el mismo estado en que se encontraba cuando se detuvo. Esto significa que toda la información relativa al proceso debe guardarse en forma explícita en algún lugar durante la suspensión. En muchos sistemas operativos, toda la información referente a cada proceso, diferente del contenido de su espacio de direcciones, se almacena en una tabla de sistema operativo llamada tabla de procesos, la cual es un arreglo (lista enlazada) de estructuras, una para cada proceso en existencia.

Por lo tanto, un proceso (suspendido) consta de su espacio de direcciones, generalmente denominado imagen del núcleo (en honor de las memorias de imagen de núcleo magnético que se utilizaron en tiempos antiguos) y su registro de la tabla de procesos, que contiene sus registros entre otras cosas.

Por Sistema de Archivo

Una función importante del S.O. consiste en ocultar las peculiaridades de los discos y otros dispositivos de E/S y presentar al programador un modelo abstracto, limpio y agradable de archivos independientes del dispositivo. Las llamadas al sistema se necesitan con claridad para crear archivos, eliminarlos, leerlos y escribirlos. Antes de que se pueda leer un archivo, éste debe abrirse y después de que se haya leído debe cerrarse, de modo que las llamadas se dan para hacer estas cosas.

Antes de que un archivo pueda leerse o escribirse, éste debe abrirse, en cuyo instante se verifican los permisos. Si se permite el acceso, el sistema produce un entero pequeño llamado descriptor del archivo para utilizarse en operaciones subsiguientes. Si se prohíbe el acceso, se produce un código de error.

3. Arquitectura de un sistema operativo

• Manejo del procesador: políticas y técnicas para la gestión (scheduling) (RP)

Proceso

Un **programa** es una secuencia de instrucciones escrita en un lenguaje dado.

Un **proceso** es una instancia de ejecución de un programa, caracterizado por su contador de programa, su palabra de estado, sus registros del procesador, su segmento de texto, pila y datos, etc.

Es posible que un programa sea ejecutado por varios usuarios en un sistema multiusuario, por cada una de estas ejecuciones existirá un proceso, con su contador de programa, registros, etc. El sistema operativo necesita el concepto de proceso para poder gestionar el procesador mediante la técnica de multiprogramación o de tiempo compartido, de hecho, el proceso es la unidad planificable, o de asignación de la CPU.

Estados De Un Proceso Y Transiciones De Estado De Los Procesos:

Durante su vida, un proceso puede pasar por una serie de estados discretos, algunos de ellos son:

- ✓ *En ejecución:* El proceso ocupa la CPU actualmente, es decir, se está ejecutando.
- ✓ *Listo o preparado:* El proceso dispone de todos los recursos para su ejecución, sólo le falta la CPU.
- ✓ *Bloqueado:* Al proceso le falta algún recurso para poder seguir ejecutándose, además de la CPU. Por recurso se pueden entender un dispositivo, un dato, etc. El proceso necesita que ocurra algún evento que le permita poder proseguir su ejecución.

Solamente puede haber un proceso en ejecución a la vez, pero pueden existir varios listos y varios pueden estar bloqueados.

Se puede definir el scheduling -algunas veces traducido como -planificación- como el conjunto de políticas y mecanismos construidos dentro del sistema operativo que gobiernan la forma de conseguir que los procesos a ejecutar lleguen a ejecutarse. El scheduling está asociado a las cuestiones de:

- ✓ Cuándo introducir un nuevo proceso en el Sistema.
- ✓ Determinar el orden de ejecución de los procesos del sistema.

El scheduling está muy relacionado con la gestión de los recursos.

Planificación A Largo Plazo:

Este planificador está presente en algunos sistemas que admiten además de procesos interactivos trabajos por lotes. Usualmente , se les asigna una prioridad baja a los trabajos por lotes, utilizándose estos para mantener ocupados a los recursos del sistema durante períodos de baja actividad de los procesos interactivos. Normalmente, los trabajos por lotes realizan tareas rutinarias como el cálculo de nóminas; en este tipo de tareas el programador puede estimar su gasto en recursos, indicándoselo al sistema. Esto facilita el funcionamiento del planificador a largo plazo.

El objetivo primordial del planificador a largo plazo es el de dar al planificador de la CPU una mezcla equilibrada de trabajos, tales como los limitados por la CPU (utilizan mucho la CPU) o la E/S. Así, por ejemplo, cuando la utilización de la CPU es baja, el planificador puede admitir más trabajos para aumentar el número de procesos listos y, con ello, la probabilidad de tener algún trabajo útil en espera de que se le asigne la CPU. A la inversa, cuando la utilización de la CPU llega a ser alta, y el tiempo de respuesta comienza a reflejarse, el planificador a largo plazo puede optar por reducir la frecuencia de admisión de trabajos.

Normalmente, se invoca al planificador a largo plazo siempre que un proceso termina. La frecuencia de invocación depende, pues, de la carga del sistema, pero generalmente es mucho menor que la de los otros dos planificadores. Esta baja frecuencia de uso hace que este planificador pueda permitirse utilizar algoritmos complejos, basados en las estimaciones de los nuevos trabajos.

Planificación a Medio Plazo:

En los sistemas de multiprogramación y tiempo compartido varios procesos residen en la memoria principal. El tamaño limitado de ésta hace que el número de procesos que residen en ella sea finito. Puede ocurrir que todos los procesos en memoria estén bloqueados, desperdi ciéndose así la CPU. En algunos sistemas se intercambian procesos enteros (swap) entre memoria principal y memoria secundaria (normalmente discos), con esto se aumenta el número de procesos, y, por tanto, la probabilidad de una mayor utilización de la CPU.

El planificador a medio plazo es el encargado de regir las transiciones de procesos entre memoria principal y secundaria, actúa intentando maximizar la utilización de los recursos. Por ejemplo, transfiriendo siempre a memoria secundaria procesos bloqueados, o transfiriendo a memoria principal procesos bloqueados únicamente por no tener memoria.

Planificación a Corto Plazo:

El planificador a corto plazo, también conocido como distribuidor (dispatcher), es el de ejecución más frecuente y toma decisiones con un mayor detalle sobre el proceso que se ejecutará a continuación.

El planificador a corto plazo se ejecuta cuando ocurre un suceso que puede conducir a la interrupción del proceso actual o que ofrece la oportunidad de expulsar de la ejecución al proceso actual a favor de otro. Como ejemplo de estos procesos están: las interrupciones de reloj, interrupciones de E/S, llamadas al sistema y señales.

Los acontecimientos que pueden provocar la llamada al dispatcher dependen del sistema (son un subconjunto de las interrupciones), pero son alguno de estos:

- ✓ El proceso en ejecución acaba su ejecución o no puede seguir ejecutándose (por una E/S, operación WAIT, etc).
- ✓ Un elemento del sistema operativo ordena el bloqueo del proceso en ejecución.
- ✓ El proceso en ejecución agota su cuantum o cuanto de estancia en la CPU.
- ✓ Un proceso pasa a estado listo.

• Manejo de memoria secundaria: políticas y técnicas para la gestión (RP)

La Organización Y Gestión De La Memoria:

Para que un proceso pueda ejecutarse debe estar ubicado en la memoria principal del ordenador. Una parte del sistema operativo se va a encargar de gestionar la memoria principal, de forma que los procesos puedan residir en la memoria sin conflictos. La gestión de la memoria implica varias tareas, una de ellas es llevar un registro de qué zonas están libres (es decir, no están siendo utilizadas por ningún proceso), y qué zonas están ocupadas por qué procesos. Otra tarea importante surge en sistemas en los que no todos los procesos, o no todo el código y datos de un proceso, se ubican en la memoria principal. En estos sistemas, a menudo se debe pasar parte, o la totalidad del código y datos de un proceso, de memoria a disco, o viceversa; siendo el sistema operativo responsable de esta tarea. De esta forma se libera al usuario de realizar estas transferencias de información, de las cuales no es consciente.

Otros dos temas importantes en la gestión de la memoria son el de la carga de los programas de disco a memoria y el de la protección. Desde el momento en que varios procesos deben compartir la memoria del ordenador surge el problema de la protección. En general, se pretende que un proceso no pueda modificar las direcciones de memoria en las que no reside. Esto es así ya que en las direcciones de memoria donde no está ubicado el proceso pueden residir otros procesos, o código o estructuras de datos del S.O. Si un proceso puede modificar indiscriminadamente la memoria, podría, por ejemplo, cambiar el valor de una dirección de memoria donde residiera una variable de otro proceso, con la consecuente ejecución incorrecta del proceso propietario de la variable.

Algunos sistemas ni siquiera permiten que un proceso pueda leer las direcciones de memoria en las que no reside, con esto se consigue privacidad sobre el código y datos de los procesos.

Existen varias formas de gestionar la memoria. Por lo común, la forma de gestión dependerá de la máquina virtual que se quiera proporcionar y del hardware subyacente. Con independencia de la forma de gestión es necesario decidir qué estrategias se deben utilizar para obtener un rendimiento óptimo. Las estrategias de administración de la memoria determinan el comportamiento de una organización de memoria determinada cuando se siguen diferentes políticas: ¿ Cuándo se coge un nuevo programa para colocarlo en la memoria ? ¿ Se coge el programa cuando el sistema lo necesita, o se intenta anticiparse a las peticiones del sistema ? ¿ En qué lugar de la memoria principal se coloca el siguiente programa por ejecutar ? ¿ Se colocan los programas lo más cerca posible unos de otros en los espacios disponibles de la memoria principal para reducir al mínimo el desperdicio de espacio, o se colocan lo más rápido posible para reducir el tiempo empleado en tomar la decisión ?

Jerarquía De La Memoria:

Los programas y datos necesitan estar en la memoria principal para ser ejecutados, o para poder ser referenciados.

Los programas o datos que no se necesitan de inmediato pueden guardarse en la memoria secundaria hasta que se necesiten, y en ese momento se transfieren a la memoria principal para ser ejecutados o referenciados. Los soportes de memoria secundaria, como cintas o discos, son en general menos caros que la memoria principal, y su capacidad es mucho mayor. Normalmente, es mucho más rápido el acceso a la memoria principal que a la secundaria.

En los sistemas con varios niveles de memoria hay muchas transferencias constantes de programas y datos entre los distintos niveles. Estas transferencias consumen recursos del sistema, como tiempo de la CPU, que de otro modo podrían utilizarse provechosamente.

En los años sesenta se hizo evidente que la jerarquía de la memoria podía extenderse un nivel más, con una clara mejora del rendimiento. Este nivel adicional, la memoria caché, es una memoria de alta velocidad, mucho más rápida que la memoria principal.

La memoria caché es extremadamente cara, si se compara con la principal, por lo que sólo se utilizan memorias caché relativamente pequeñas.

La memoria caché introduce un nivel adicional de transferencia de información en el sistema. Los programas en memoria principal se pasan a la memoria caché antes de ejecutarse. En la memoria caché se pueden ejecutar mucho más rápido que en la principal. La esperanza de los diseñadores es que el trabajo extra requerido por la transferencia de programas sea mucho menor que el incremento del rendimiento obtenido por la ejecución más rápida en la caché.

• Manejo de dispositivos de E/S (RP)

Su control adecuado es indispensable para el funcionamiento de la interfaz y del sistema de archivos y para la comunicación de la máquina con el exterior, leyendo y escribiendo datos a través de diversos dispositivos.

Los componentes de E/S en un S.O está constituida básicamente por los manejadores de dispositivos de los cuales existe uno para cada tipo.

El código destinado a manejar la entrada y salida de los diferentes periféricos en un sistema operativo es de una extensión considerable y sumamente complejo. Resuelve la necesidades de sincronizar, atrapar interrupciones y ofrecer llamadas al sistema para los programadores.

Un manejador es una rutina que se encarga de programar a un controlador, y de esta manera realizar la entrada salida de datos.

Un manejador básicamente recibe órdenes generales de entrada salida, como "envía" el contenido del buffer a la impresora, y luego las convierte a secuencias de instrucciones que envía al controlador correspondiente, esperando sus respuesta.

Los dispositivos de entrada salida (o manejadores) se dividen, en general, en dos tipos:

- ✓ **Orientados a Bloques:** Los dispositivos orientados a bloques tienen la propiedad de que se pueden direccionar, esto es, el programador puede escribir o leer cualquier bloque del dispositivo realizando primero una operación de posicionamiento sobre el dispositivo. Los dispositivos más comunes orientados a bloques son los discos duros, la memoria, discos compactos y, posiblemente, unidades de cinta.
- ✓ **Orientados a Caracteres:** Los dispositivos orientados a caracteres son aquellos que trabajan con secuencias de bytes sin importar su longitud ni ninguna agrupación en especial. No son dispositivos direccionables. Ejemplos de estos dispositivos son el teclado, la pantalla o display y las impresoras.

En el manejo de los dispositivos de E/S es necesario, introducir dos nuevos términos:

Buffering (uso de memoria intermedia):

El buffering trata de mantener ocupados tanto la CPU como los dispositivos de E/S. La idea es sencilla, los datos se leen y se almacenan en un buffer, una vez que los datos se han leído y la CPU va a iniciar inmediatamente la operación con ellos, el dispositivo de entrada es instruido para iniciar inmediatamente la siguiente lectura. La CPU y el dispositivo de entrada permanecen ocupados. Cuando la CPU esté libre para el siguiente grupo de datos, el

dispositivo de entrada habrá terminado de leerlos. La CPU podrá empezar el proceso de los últimos datos leídos, mientras el dispositivo de entrada iniciará la lectura de los datos siguientes.

Para la salida, el proceso es análogo. En este caso los datos de salida se descargan en otro buffer hasta que el dispositivo de salida pueda procesarlos.

Este sistema soluciona en forma parcial el problema de mantener ocupados todo el tiempo la CPU y los dispositivos de E/S. Ya que todo depende del tamaño del buffer y de la velocidad de procesamiento tanto de la CPU como de los dispositivos de E/S.

El manejo de buffer es complicado. Uno de los principales problemas reside en determinar tan pronto como sea posible que un dispositivo de E/S a finalizado una operación. Este problema se resuelve mediante las interrupciones. Tan pronto como un dispositivo de E/S acaba con una operación interrumpe a la CPU, en ese momento la CPU detiene lo que está haciendo e inmediatamente transfiere el control a una posición determinada. Normalmente las instrucciones que existen en esta posición corresponden a una rutina de servicio de interrupciones. La rutina de servicio de interrupción comprueba si el buffer no está lleno o no está vacío y entonces inicia la siguiente petición de E/S. La CPU puede continuar entonces el proceso interrumpido.

Spooling:

El problema con los sistemas de cintas es que una lectora de tarjetas no podía escribir sobre un extremo mientras la CPU leía el otro. Los sistemas de disco eliminaron esa dificultad, moviendo la cabeza de un área del disco a otra.

Un sistema de discos, las tarjetas se leen directamente desde la lectora sobre el disco. La posición de las imágenes de las tarjetas se registran en una tabla mantenida por el sistema operativo. En la tabla se anota cada trabajo una vez leído. Cuando se ejecuta un trabajo sus peticiones de entrada desde la tarjeta se satisfacen leyendo el disco. Cuando el trabajo solicita la salida, ésta se copia en el buffer del sistema y se escribe en el disco. Cuando la tarea se ha completado se escribe en la salida realmente.

Esta forma de procesamiento se denomina spooling, utiliza el disco como un buffer muy grande para leer tan por delante como sea posible de los dispositivos de entrada y para almacenar los ficheros hasta que los dispositivos de salida sean capaces de aceptarlos.

La ventaja sobre el buffering es que el spooling solapa la E/S de un trabajo con la computación de otro. Es una característica utilizada en la mayoría de los sistemas operativos.

• Sistema de archivos: archivos y directorios. Estructura, organización y tipos (RP)

Archivos:

Se dice que el sistema de archivos es el esqueleto sobre el que descansa gran parte del sistema operativo y de las aplicaciones.

Los archivos comienzan a surgir cuando el sistema crece y se requiere de mayor capacidad y rapidez; en ese momento se necesitan dispositivos de acceso directo como son los discos en todas sus variedades.

Un archivo desde el punto de vista del S.O es un conjunto de datos que deseamos manejar juntos.

Físicamente es una colección de bloques de disco o cinta, que incluyen a los datos mismos y además otros datos para dar unidad a los bloques que lo forman. Estos últimos se denominarán Apuntadores Bloques de Datos.

Los apuntadores a bloques de datos se organizan en varias formas:

- ✓ Como un dato al final de cada bloque, enlazándolos en una lista ligada.
- ✓ Como un bloque de apuntadores a los bloques.
- ✓ Como una lista ligada externa que apunta a los bloques de datos.

Tipos de Archivos:

- ✓ *Archivos Regulares:* Son los que contienen información sobre el usuario.
- ✓ *Directorios:* Archivos de sistema que sirven para mantener la estructura del sistema de archivos.
- ✓ *Archivos Especiales por Caracteres:* Están relacionados con E/S y sirven para modelar dispositivos de E/S en serie como las terminales, impresoras y redes.

- ✓ *Archivos Especiales por Bloques:* Sirven para modelar discos.
- ✓ *Archivos Normales:* Los archivos normales por lo general son archivos ASCII o bien archivos binarios. Los archivos ASCII consisten en líneas de texto y su gran ventaja es que pueden exhibirse e imprimirse tal como están y se pueden editar con un editor normal de texto. Los archivos binarios están compuestos por un listado incomprendible lleno de lo que parece ser basura, pero por lo regular, estos archivos tienen una estructura interna.

Componentes:

Un sistema de archivos, de cualquier tipo, tiene dos componentes básicas:

- ✓ Una que permite el acceso a los archivos.
- ✓ Y otra que controla el uso de éstos.

Cada una conformada por un conjunto de estructuras y una colección de operaciones.

Llamada A Las Funciones De Acceso Y Control:

El control de archivos incluye dos aspectos: el control físico del espacio disponible y el control lógico de ellos. El primer incluye los Apuntadores a Bloques de Datos, el Bit Map y la tabla de archivos dañados. El segundo incluye los directorios y tablas de archivos activos.

El propósito del control lógico es el manejo de archivos en forma unificada y homogénea, sin atender a los aspectos físicos.

Para algunas funciones, especialmente las que tienen que ver con la componente de control, se requiere de parámetros como estos:

- a) identificación del archivo (dispositivo, directorio, nombre, tipo, versión)
- b) modo de empleo (por registro, por bloque, por carácter).
- c) forma de acceso (secuencial o directo).
- d) uso propuesto (lectura, escritura, adición de registros).
- e) organización (montón, secuencial, secuencial indizado, etc).

- ✓ *Modulo de Acceso:* El módulo de acceso es el que se encarga de las operaciones básicas sobre un archivo y es el enlace con el manejador del dispositivo donde éste se localiza. Las operaciones básicas que realiza son cuatro:

- a) Abrir. Esta operación consiste de los siguientes pasos: localizar el archivo en el directorio y de ahí en el índice; con ello, se tiene la dirección inicial del archivo.
 - b) Cerrar. Pasos inversos a la operación de abrir.
 - c) Leer. Esta operación transfiere datos a la rutina llamadora, de acuerdo a la solicitud. Generalmente los datos se transmiten del buffer a la tarea, registro por registro, pero pueden ser también por variable o por carácter.
 - d) Escribir. Todo lo que el usuario escribe se va acumulando en el buffer y cuando este se llena, se le envía al archivo a través del manejador del dispositivo correspondiente, después de verificar que exista espacio disponible.

- ✓ *Modulo de Control:* En este módulo importan más sus estructuras que sus procesos.

Para los discos magnéticos las principales estructuras son:

- A) Control físico del espacio
 - a) bit map
 - b) archivo de bloques dañados
- B) Control lógico
 - c) directorios
 - d) encabezados
 - e) tabla de archivos activos
 - f) descriptor de volumen y bloque de carga

Para las *cintas magnéticas* aún representan un medio económico de almacenar y transferir grandes volúmenes de datos y programas. Sin embargo, la limitación de su acceso secuencial permite un manejo más simple. Básicamente, su organización puede darse en tres formas:

- a) archivos sin etiquetas

- b) archivos con etiquetas estándar (ANSI)
- c) archivos con etiquetas del fabricante del equipo.

En general los principales procesos contenidos en la componente de control son:

- a) Borrar: Este proceso se realiza eliminando lógicamente el archivo, es decir, se le marca "borrado", pero físicamente no se limpian los bloques que antes ocupaba.
- b) Buscar: Para buscar un archivo se recorren los directorios hasta encontrarlo o hasta encontrar una marca de fin de directorio.
- c) Crear: Busca un lugar en el directorio que corresponda, tomando el primer lugar marcado libre por una operación de borrado o después la marca de fin de directorio.
- d) Conformar: Es el proceso de estructuración de un disco para que pueda recibir archivos.
- e) Anicilizar: Es el proceso de hacer reconocible al disco por el sistema de archivos.
- f) Pedir espacio: Cuando se va a crear un nuevo bloque de datos en un archivo, debe pedirse. El proceso de asignación incluye la visita al bit map para localizar el segmento apropiado y la actualización de los bloques requiere de bloques adicionales, para nuevos apuntadores a bloques de datos.
- g) Renombrar: Este es un proceso sencillo y basta con localizar el archivo y cambiarle el nombre en el índice y los directorios.

Arreglar lo de modulo de control y acceso y añadir de las copias acceso a archivos

Directorios:

Cuando un dispositivo es compartido por varios usuarios, es conveniente organizar sus archivos en grupos. También cuando un usuario tiene archivos de varios tipos, aplicaciones o proyectos, resulta conveniente distinguirlos. Así surgen los **directorios** como una organización de tipo jerárquico impuesta sobre los archivos.

La organización de los archivos puede hacerse de diversas formas y empleando diferentes atributos para clasificarlos. Los más frecuentes son la clave o cuenta del usuario y el tipo de archivo. Sin embargo también existen sistemas con directorios arbitrarios.

En general se podría decir que un directorio es un archivo que contiene información acerca de un grupo de archivos que se desea manejar como una unidad.

La estructura básica de un directorio es con una raíz, una serie de directorios a uno o más niveles y, al final, los descriptores de archivos, que vienen a ser las hojas. La representación interna de un directorio se realiza generalmente como árbol binario donde los nodos hermanos forman una lista que se revisa secuencialmente y solo se cambia a otro archivo cuando se pasa a un nodo descendiente.

Debido a este tipo de estructura, deben evitarse dos extremos al estructurar los archivos en directorios: si hay pocos niveles, la búsqueda secuencial será larga; si hay demasiados niveles, los cambios de nodo serán muchos haciendo lento el proceso de localización de archivos.

• Seguridad y protección (RP)

En los sistemas operativos se requiere tener una buena seguridad informática, tanto del hardware, programas y datos, previamente haciendo un balance de los requerimientos y mecanismos necesarios. Con el fin de asegurar la integridad de la información contenida.

Dependiendo de los mecanismos utilizados y de su grado de efectividad, se puede hablar de sistemas seguros e inseguros. En primer lugar, deben imponerse ciertas características en el entorno donde se encuentra la instalación de los equipos, con el fin de impedir el acceso a personas no autorizadas, mantener un buen estado y uso del material y equipos, así como eliminar los riesgos de causas de fuerza mayor, que puedan destruir la instalación y la información contenida.

En la actualidad son muchas las violaciones que se producen en los sistemas informáticos, en general por acceso de personas no autorizadas que obtienen información confidencial pudiendo incluso manipularla. En ocasiones este tipo de incidencias resulta grave por la naturaleza de los datos; por ejemplo si se trata de datos bancarios, datos oficiales que puedan afectar a la seguridad de los estados, etc.

El software mal intencionado que se produce por diversas causas, es decir pequeños programas que poseen gran facilidad para reproducirse y ejecutarse, cuyos efectos son destructivos nos estamos refiriendo a los virus informáticos.

Para esto, se analizan cuestiones de seguridad desde dos perspectivas diferentes la seguridad externa y la seguridad interna.

Todos los mecanismos dirigidos a asegurar el sistema informático sin que el propio sistema intervenga en el mismo se engloban en lo que podemos denominar seguridad externa.

La seguridad externa puede dividirse en dos grandes grupos:

- ✓ **Seguridad física.** Engloba aquellos mecanismos que impiden a los agentes físicos la destrucción de la información existente en el sistema; entre ellos podemos citar el fuego, el humo, inundaciones descargas eléctricas, campos magnéticos, acceso físico de personas con no muy buena intención, entre otros.
- ✓ **Seguridad de administración.** Engloba los mecanismos más usuales para impedir el acceso lógico de personas físicas al sistema.

Todos los mecanismos dirigidos a asegurar el sistema informático, siendo el propio sistema el que controla dichos mecanismos, se engloban en lo que podemos denominar seguridad interna.

4. Desempeño de un sistema operativo

• Herramientas matemáticas asociadas: teoría de colas, cálculo de probabilidades, procesos de Markov (RP)

Teoría de colas:

Los modelos analíticos son representaciones matemáticas de los sistemas con los cuales el evaluador de desempeño puede sacar conclusiones rápidas y precisas acerca del comportamiento de un sistema. Los modelos de colas son útiles para modelar sistemas en los que las solicitudes esperan en líneas, o colas hasta que se les puede atender.

Las colas pueden ser limitadas o ilimitadas. Los clientes llegan a un sistema de colas procedentes de una fuente que puede ser finita o infinita. Los tiempos entre llegadas sucesivas se llaman tiempos entre llegadas.

Procesos de Markov:

Son útiles para modelar sistemas que pueden estar en uno de un conjunto de estados discretos mutuamente exclusivos y colectivamente exhaustos. Dado que un proceso de Harkov está en un estado determinado, su comportamiento futuro no depende de su historia antes de llegar a ese estado. Solo tomamos en cuenta procesos de Harkov que presentan un comportamiento de estado estable.

Un caso especial de los procesos de Harkov, el de los procesos de nacimiento y muerte resulta especialmente aplicable al modelado de sistemas de cómputo. Los procesos de nacimiento y muerte son mucho más fáciles de resolver que los procesos de Harkov generales.

• Algoritmos de scheduling (RP)

Cuando hay más de un proceso ejecutable, el sistema operativo debe decidir cuál ejecutará primero. La parte del sistema que toma esta decisión se denomina planificador; el algoritmo que usa se denomina algoritmo de planificación.

Planificación Round Robin:

A cada proceso se le asigna un intervalo de tiempo, llamado cuanto, durante el cual se le permite ejecutarse. Si el proceso todavía se está ejecutando al expirar su cuento, el sistema operativo se apropiá de la CPU y se la da a otro proceso. Si el proceso se bloquea o termina antes de expirar el cuento, la comutación de CPU naturalmente se efectúa cuando el proceso se bloquee. Todo lo que el planificador tiene que hacer es mantener una lista de procesos ejecutable. Cuando un proceso gasta su cuento, se le coloca al final de la lista.

Planificación por prioridad:

A cada proceso se le asigna una prioridad, y se permite que se ejecute el proceso ejecutable que tenga la prioridad

más alta. A fin de evitar que los procesos de alta prioridad se ejecuten indefinidamente, el planificador puede reducir la prioridad de los procesos que actualmente se ejecutan en cada tic del reloj. Si esta acción hace que la prioridad se vuelva menor que la del siguiente proceso con más alta prioridad, ocurrirá una comutación de procesos.

Colas múltiples:

Para aprovechar al máximo el tiempo de ejecución de los procesos, se establecieron lo que son clases de prioridad. Los procesos de la clase más alta se ejecutaban durante un cuanto. Los procesos de la siguiente clase más alta se ejecutaban durante dos cuantos. Los procesos de la siguiente clase se ejecutaban durante cuatro cuantos, y así sucesivamente. Cada vez que un proceso agotaba todos los cuantos que tenía asignados, se le degradaba una clase.

El primer trabajo mas corto:

Son perfectos para los trabajos por lotes cuyos tiempos de ejecución se conocen por adelantado. Primero se realizan los procesos con tiempo de ejecución menor. Así, un proceso corto saltará a la cabeza de la cola, sobrepasando a trabajos largos.

Planificación en tiempo real:

Un sistema de tiempo real es uno en el que el tiempo desempeña papel esencial.. Por lo regular, uno o más dispositivos físicos externos a la computadora generan estímulos, y la computadora debe reaccionar a ellos de la forma apropiada dentro de un plazo.

5. Dispositivos y servicios especiales

• Construcción de drivers (E, RP)

Su control adecuado es indispensable para el funcionamiento de la interfaz y del sistema de archivos y para la comunicación de la máquina con el exterior, leyendo y escribiendo datos a través de diversos dispositivos. Los componentes de E/S en un S.O. Está constituida básicamente por los manejadores de dispositivos de los cuales existe uno para cada tipo.

Dispositivos:

A cada dispositivo físico reconocido le asigna un nombre interno. Un dispositivo físico puede tener varios nombres y algunos nombres pueden ser asignados dinámicamente a diferentes dispositivos. Al conjunto de nombres internos de los dispositivos se les denomina dispositivos lógicos. El manejo de dispositivos lógicos permite cambiar el destino de un archivo, sin modificar el programa que lo genera, asignándole el nombre a otro archivo. En un sistema pueden existir dispositivos especiales, tal como el dispositivo del sistema (aquel donde reside el sistema operativo). Por otra parte, los dispositivos pueden agruparse en tres tipos, de acuerdo a su uso:

- a) asignados o dedicados
- b) compartidos y
- c) virtuales

- ✓ Dispositivos asignados. Estos son aquellos que únicamente pueden atender a una tarea, por un tiempo relativamente largo. Por ello se asigna a la tarea que los requiere y son de su uso exclusivo hasta que los libere o concluya su ejecución.
- ✓ Dispositivos compartidos. Existen dispositivos que por su naturaleza permiten que las tareas los empleen por tiempos muy cortos, de tal forma que el efecto real es de que varias tareas los comparten (como el caso de los discos magnéticos y los discos ópticos).
- ✓ Dispositivos virtuales. Emulador. Es la simulación de un dispositivo asignado en otro compartido.

Controladores:

La mayoría de los dispositivos se conectan a la computadora a través de un controlador. Existen controladores que soportan un solo dispositivo. Cada controlador recibe órdenes del procesador, generadas por el SO u otro programa; los datos los extrae directamente de memoria. El controlador realiza sus instrucciones haciendo trabajar a los dispositivos que tiene asociados, y supervisando su operación. Los resultados se envían de nuevo al procesador generalmente a través de interrupciones. Desde el punto de vista del S.O. un controlador puede verse como un conjunto de registros de uso especial, que realizan una o más de las siguientes funciones: transmisión de

órdenes, envío y recibo de datos y control del estado de los dispositivos conectados. Cuando se desea realizar una operación de entrada o salida, es necesario averiguar el estado del dispositivo, enviar las órdenes necesarias y enviar o recibir los datos.

Relación con el resto del sistema operativo:

Los manejadores correspondientes a dispositivos, son los programas que realizan el proceso de transferencia de datos y órdenes con el controlador. Su acción es iniciada desde algún otro programa, especialmente del subsistema de archivos. En la entrada-salida, como se maneja usualmente desde un lenguaje de alto nivel, las órdenes siempre pasan por el sistema de archivos, donde se determina el dispositivo y el archivo a que se refiere la orden.

Estructura de datos:

En el manejo de los dispositivos existen una cuantas estructuras básicas: los buffers que recibe del sistema de archivos y que regresa al mismo; los registros del controlador y los dispositivos, que debe consultar y modificar, y las que le son propias: descriptores de dispositivos y cola de solicitudes pendientes. Los registros de controladores y dispositivos son partes de estos que sirven para comunicarse con el resto de la computadora. Su consulta y actualización se realiza por medio de subrutinas primitivas de la parte central del SO. Estos registros contienen los siguientes elementos: orden específica recibida, resultado de una operación, parámetros recibidos, estado del dispositivo y códigos de error. Los descriptores son tablas que contienen información acerca de los dispositivos. Los elementos mínimos de un descriptor de dispositivos son los siguientes:

- ✓ nombre del dispositivo (lógico)
- ✓ controlador del que depende
- ✓ estado actual (listo, en falla)
- ✓ apuntadores a lista de solicitudes pendientes

La cola de dispositivos pendientes se refiere a aquellas solicitudes originadas en las tareas, a través del sistema de archivos o de otras partes del sistema operativo, y que deben ser satisfechas por algún dispositivo.

Manejadores:

Un manejador es una rutina que se encarga de programar a un controlador, y de esta manera realizar la entrada salida de datos. Un manejador básicamente recibe órdenes generales de entrada salida, como "envía" el contenido del buffer a la impresora, y luego las convierte a secuencias de instrucciones que envía al controlador correspondiente, esperando sus respuestas. Los manejadores pueden agruparse en dos categorías:

- ✓ Aquellas que manejan registros más o menos grandes, que son enviados o recibidos en memoria, (ej. disco o cinta magnética).
- ✓ Aquellos que manejan cadenas de caracteres enviados (o recibidos) uno a uno (ej. las terminales).

Otra diferencia se da entre los dispositivos asignados y compartidos. En el primer caso la activación del manejador queda bajo la responsabilidad de la tarea solicitante, mientras que en el segundo el manejador permanece activo mientras haya solicitudes en la cola asignada a él, venga de donde sea. Un manejador puede verse como un paquete que contiene varias rutinas.

Las principales son:

- ✓ atención a la interrupción del dispositivo
- ✓ inicio de atención a solicitud
- ✓ tiempo límite del dispositivo
- ✓ cancelación de una solicitud

II. Sistemas operativos especializados

1. Tipos especiales de sistemas operativos

• Intercomunicación entre procesos (E)

• Sistemas operativos de red: servidores de archivos y de impresión, arquitectura cliente-servidor, arquitectura par a par (E)

Sistemas operativos de red.

Son aquellos sistemas que mantienen a dos o más computadoras unidas a través de algún medio de comunicación (físico o no), con el objetivo primordial de poder compartir los diferentes recursos y la información del sistema.

Ventajas:

- ✓ Aumento de la productividad: Los usuarios pueden utilizar herramientas que le son familiares, como hojas de cálculo y herramientas de acceso a bases de datos. Mediante la integración de las aplicaciones cliente/servidor con las aplicaciones personales de uso habitual, los usuarios pueden construir soluciones particularizadas que se ajusten a sus necesidades cambiantes. Una interfaz gráfica de usuario consistente reduce el tiempo de aprendizaje de las aplicaciones.
- ✓ Menores costes de operación: Permiten un mejor aprovechamiento de los sistemas existentes, protegiendo la inversión. Por ejemplo, la compartición de servidores (habitualmente caros) y dispositivos periféricos (como impresoras) entre máquinas clientes permite un mejor rendimiento del conjunto.
- ✓ Proporcionan un mejor acceso a los datos: La interfaz de usuario ofrece una forma homogénea de ver el sistema, independientemente de los cambios o actualizaciones que se produzcan en él y de la ubicación de la información. El movimiento de funciones desde un ordenador central hacia servidores o clientes locales origina el desplazamiento de los costes de ese proceso hacia máquinas más pequeñas y por tanto, más baratas.
- ✓ Mejora en el rendimiento de la red: Las arquitecturas cliente/servidor eliminan la necesidad de mover grandes bloques de información por la red hacia los ordenadores personales o estaciones de trabajo para su proceso. Los servidores controlan los datos, procesan peticiones y después transfieren sólo los datos requeridos a la máquina cliente. Entonces, la máquina cliente presenta los datos al usuario mediante interfaces amigables. Todo esto reduce el tráfico de la red, lo que facilita que pueda soportar un mayor número de usuarios. Tanto el cliente como el servidor pueden escalarse para ajustarse a las necesidades de las aplicaciones. Las UCPs utilizadas en los respectivos equipos pueden dimensionarse a partir de las aplicaciones y el tiempo de respuesta que se requiera. La existencia de varias UCPs proporciona una red más fiable: un fallo en uno de los equipos no significa necesariamente que el sistema deje de funcionar. En una arquitectura como ésta, los clientes y los servidores son independientes los unos de los otros con lo que pueden renovarse para aumentar sus funciones y capacidad de forma independiente, sin afectar al resto del sistema. La arquitectura modular de los sistemas cliente/servidor permite el uso de ordenadores especializados (servidores de base de datos, servidores de ficheros, estaciones de trabajo para CAD, etc.). Permite centralizar el control de sistemas que estaban descentralizados, como por ejemplo la gestión de los ordenadores personales que antes estuvieran aislados.

Inconvenientes:

Hay una alta complejidad tecnológica al tener que integrar una gran variedad de productos. Requiere un fuerte rediseño de todos los elementos involucrados en los sistemas de información (modelos de datos, procesos, interfaces, comunicaciones, almacenamiento de datos, etc.). Además, en la actualidad existen pocas herramientas que ayuden a determinar la mejor forma de dividir las aplicaciones entre la parte cliente y la parte servidor. Es más difícil asegurar un elevado grado de seguridad en una red de clientes y servidores que en un sistema con un único ordenador centralizado. A veces, los problemas de congestión de la red pueden degradar el rendimiento del sistema por debajo de lo que se obtendría con una única máquina (arquitectura centralizada). También la interfaz gráfica de usuario puede a veces ralentizar el funcionamiento de la aplicación. El quinto nivel de esta arquitectura (bases de datos distribuidas) es técnicamente muy complejo y en la actualidad hay muy pocas implantaciones que garanticen un funcionamiento totalmente eficiente. Existen multitud de costes ocultos (formación en nuevas tecnologías, licencias, cambios organizativos, etc.) que encarecen su implantación.

Arquitectura cliente/servidor y downsizing

Muchas organizaciones están transportando sus aplicaciones a plataformas más pequeñas (downsizing) para conseguir la ventaja que proporcionan las nuevas plataformas físicas más rentables y la arquitectura cliente/servidor. Este transporte siempre supone un coste, debido a la necesidad de rediseñar las aplicaciones y de re-entrenar a los usuarios en los nuevos entornos.

Independencia de Bases de Datos:

Las arquitecturas cliente/servidor permiten aprovechar los conceptos de cliente y servidor para desarrollar aplicaciones que accedan a diversas bases de datos de forma transparente. Esto hace viable cambiar la aplicación en la parte servidora, sin que la aplicación cliente se modifique. Para que sea posible desarrollar estas aplicaciones es necesaria la existencia de un estándar de conectividad abierta que permita a los ordenadores personales y estaciones de trabajo acceder de forma transparente a bases de datos corporativas heterogéneas.

Relación con los Sistemas Abiertos:

Las arquitecturas cliente/servidor se asocian a menudo con los sistemas abiertos, aunque muchas veces no hay una relación directa entre ellos. De hecho, muchos sistemas cliente/servidor se pueden aplicar en entornos propietarios. En estos entornos, el equipo físico y el lógico están diseñados para trabajar conjuntamente, por lo que, en ocasiones se pueden realizar aplicaciones cliente/servidor de forma más sencilla y fiable que en los entornos que contienen plataformas heterogéneas. El problema surge de que los entornos propietarios ligan al usuario con un suministrador en concreto, que puede ofrecer servicios caros y limitados. La independencia del suministrador que ofrecen los entornos de sistemas abiertos, crea una competencia que origina mayor calidad a un menor precio. Pero, por otra parte, debido a la filosofía modular de los sistemas cliente/servidor, éstos se utilizan muchas veces en entornos de diferentes suministradores, adecuando cada máquina del sistema a las necesidades de las tareas que realizan. Esta tendencia está fomentando el crecimiento de las interfaces gráficas de usuario, de las bases de datos y del software de interconexión. Debido a esto, se puede afirmar que los entornos cliente/servidor facilitan el movimiento hacia los sistemas abiertos. Utilizando este tipo de entornos, las organizaciones cambian sus viejos equipos por nuevas máquinas que pueden conectar a la red de clientes y servidores. Los suministradores, por su parte, basan uno de los puntos clave de sus herramientas cliente/servidor en la interoperabilidad.

Relación con Orientación a Objetos:

No hay una única forma de programar aplicaciones cliente/servidor; sin embargo, para un desarrollo rápido de aplicaciones cliente/servidor y para obtener una reducción importante de costes, la utilización de la tecnología cliente/servidor puede considerarse en conjunción con la de orientación a objetos.

Características:

Entre las principales características de la arquitectura Cliente/Servidor, se pueden destacar las siguientes:

- ✓ El servidor presenta a todos sus clientes una interfaz única y bien definida.
- ✓ El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- ✓ El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- ✓ Los cambios en el servidor implican pocos o ningún cambio en el cliente.

Todos los sistemas desarrollados en arquitectura Cliente/Servidor poseen las siguientes características distintivas de otras formas de software distribuido:

- ✓ La arquitectura Cliente/Servidor es una infraestructura versátil modular y basada en mensajes que pretende mejorar la portabilidad, la interoperabilidad y la escalabilidad del cómputo; además es una apertura del ramo que invita a participar a una variedad de plataformas, hardware y software del sistema.

• **Sistemas operativos distribuidos: sistema de archivos distribuidos, memoria distribuida, balanceo de carga (E)**

Sistemas distribuidos

Antes de hablar de sistemas operativos distribuidos primero hay que hablar un poco acerca de sistemas distribuidos. No hay un consenso en la literatura acerca de la definición de un sistema distribuido. Sin embargo, lo que es aceptado por todos son dos aspectos que lo caracterizan:

- ✓ Esta formado por un conjunto de máquinas autónomas conectadas entre sí.
- ✓ Los usuarios perciben el sistema como si fuera un solo computador.
- ✓ El surgimiento de los sistemas distribuidos fue posible gracias a la invención de las redes de computadores junto con el desarrollo tecnológico de enlaces de comunicación de alta velocidad. Esto permitió la construcción de las redes tipo LAN y WAN.

Dentro de las principales motivaciones que conlleva a la construcción de sistemas distribuidos se encuentran:

- ✓ Economía.
- ✓ Velocidad
- ✓ Distribución Inherente
- ✓ Confiabilidad
- ✓ Crecimiento incremental o escalabilidad

Estas motivaciones surgen de un análisis comparativo entre la utilización de sistemas centralizados ver sus la utilización de sistemas distribuidos. Un sistema centralizado se entiende como un sistema que posee una sola CPU con sus respectivos recursos al cual se le pueden conectar terminales. Por otro lado, el uso de sistemas distribuidos frente al uso de computadores autónomos trabajando aisladamente presenta otro conjunto de ventajas:

- ✓ Información compartida
- ✓ Dispositivos compartidos
- ✓ Facilidad de comunicación
- ✓ Flexibilidad

Hasta el momento solo hemos hablado de las maravillas que proporcionan los sistemas distribuidos ahora mencionaremos cuales son sus principales desventajas.

- ✓ Software
- ✓ Redes
- ✓ Seguridad
- ✓ Aspectos del Hardware

Desde el punto de vista del Hardware presente en un sistema distribuido debemos considerar las diversas formas en que es posible interconectar varios computadores o bien varias CPUs. Desde esta perspectiva existe una taxonomía general propuesta por Flynn (1972). Flynn propuso 4 diferentes categorías en que se podían clasificar los sistemas hardware existentes de acuerdo a dos parámetros. Número de flujos de instrucciones y número de flujos de datos. La siguiente es la clasificación que propuso.

- ✓ **SISD.** Un flujo de instrucción con un flujo de datos.
- ✓ **SIMD.** Un flujo de instrucción varios flujos de datos.
- ✓ **MISD.** Varios flujos de instrucciones un flujo de datos. (No se usa)
- ✓ **MIMD.** Varios flujos de instrucciones y varios flujos de datos.

De acuerdo a estas categorías un sistema distribuido se encuentra dentro de la última mencionada. Y en particular dentro de esta categoría se pueden encontrar dos tipos distintos de la forma en que se puede interconectar el hardware. Es así, como tenemos los siguientes grandes grupos. Dentro de cada uno de ellos se distinguen dos maneras distintas de interconexión.

- ✓ Multiprocesadores.
- ✓ Multicomputadores.

Cada uno de estos tipos, permite una interconexión de sus componentes tipo bus y tipo conmutada. Desde el punto de vista de las velocidades de comunicación que pueden alcanzar estos dos grandes grupos se tienen dos conceptos asociados: sistemas fuertemente acoplados y sistemas débilmente acoplados

Multiprocesador:

Los multiprocesadores corresponden a un conjunto de CPUs conectadas entre sí que utilizan un espacio de direccionamiento virtual común.

Multiprocesadores con conexión de bus:

En este caso las CPUs están interconectadas entre sí, mediante un bus. Cada vez que una CPU quiere realizar un acceso de lectura o escritura debe accesar el bus. Con la existencia de un único bus compartido por múltiples CPUs, se puede producir un problema de rendimiento en el sentido que no es posible que mas de una CPU accese el canal de comunicaciones simultáneamente. Este problema por lo general se ataca mediante la utilización de memoria caché. Sin embargo, el uso de memoria caché también produce otros problemas relacionados con la coherencia de la información. Para solucionar este último problema normalmente se utilizan caches monitores.

Multiprocesadores con conexión conmutada

En este caso, la memoria se divide en diversos bancos, y las CPUs se interconectan con ellas no mediante un canal tipo bus sino de otras manera, una de ellas se le conoce como conexión crossbar y la otra como red omega. En el primer caso, en cada vértice de la matriz formada se ubica un conmutador, los cuales se cierran de acuerdo a los requerimientos de acceso entre CPU y banco de Memoria. La red Omega en cambio contiene conmutadores 2x2, cada uno de los cuales tiene dos entradas y dos salidas. Los principales problemas que enfrentan los multiprocesadores están limitados en la cantidad de CPUs que puede interconectar, los de conexión bus por una parte no puede recargar mucho el uso del canal; y los de conexión mediante conmutadores permiten mas CPUs, pero son mas lentos y caros.

Multicomputadores:

Cuando se habla de multicomputadores, es para referirse a sistemas de cómputo con memoria distribuida, en este caso cada computador posee su propia memoria local. Por lo tanto, se comunican con los otros computadores mediante algún canal que los conecte usando primitivas de comunicación.

Multicomputadores con conexión de bus

Este esquema es similar al caso de los multiprocesadores con conexión tipo bus, sólo que no se requiere que el canal de comunicación sea tan alto como es en el caso de los multiprocesadores. Por lo general este esquema corresponde a una LAN conectada con velocidades de 10M, 100Mb/seg. Claro que si el canal ofrece mayor velocidad mejor es el desempeño del sistema.

Aspectos del Software:

El software, por su naturaleza es difícil de clasificarlo, a diferencia del hardware. Sin embargo, es posible hacer una clasificación mas bien cualitativa. Se dice que el software puede ser fuertemente o débilmente acoplado, de acuerdo al grado de interacción distribuida que permita. Desde el punto de vista del software los siguientes son los mas conocidos.

- ✓ Sistemas operativos de redes
- ✓ Sistemas distribuidos reales
- ✓ Sistemas de multiprocesadores con tiempo compartido
- ✓ Sistemas operativos de redes

Este caso de software se dice que es débilmente acoplado en un hardware también débilmente acoplado. Este esquema corresponde a una LAN en donde

cada computador opera en forma autónoma posee su propio sistema operativo. Sin embargo, este esquema permite la comunicación e interacción entre los distintos computadores mediante algunos comandos de red. Un ejemplo de esto, es el comando rlogin de Unix, el cual permite abrir una sesión en una máquina remota, antes de hacerse efectiva la conexión remota se realiza la desconexión de la máquina original, este comando permite usar la máquina original como terminal de la máquina remota. Otro comando de Unix similar al rlogin es el rcp, el cual permite copiar un archivo de una máquina a otra.

Sistemas distribuidos reales:

Este esquema corresponde a un sistema de software fuertemente acoplado en un sistema de hardware débilmente acoplado. En este caso, el software utiliza los recursos del hardware distribuido de la misma manera como un sistema operativo convencional utiliza los recursos del computador donde es montado. Esta perspectiva es percibida por los usuarios como si la red de computadores fuera un solo sistema de tiempo compartido. Es en este sentido que se habla de sistemas operativos distribuidos. Un sistema operativo distribuido contempla varias funciones, entre las que se encuentran:

- ✓ Mecanismos de comunicación global entre procesos
- ✓ Mecanismos de protección global
- ✓ Interfaz de llamadas a sistema homogénea
- ✓ Distribución adecuada de los procesos en el sistema
- ✓ Sistemas de multiprocesadores con tiempo compartido

En este caso se tiene un software fuertemente acoplado con un hardware fuertemente acoplado. Este esquema contempla un sistema operativo para una máquina que contienen múltiples procesadores trabajando con memoria compartida. Por lo general, cada CPU tiene una memoria caché. Una de las principales características de estos sistemas es en cuanto a la ejecución de proceso existe una cola de procesos listos, que pueden ser atendidos por cualquier CPU, respecto a este punto es necesario garantizar el acceso adecuado a dicha cola. Además el uso de cachés, puede influir en la decisión del planificador respecto a la CPU que sería mejor ejecutarse un proceso determinado. El sistema operativo, en este caso, puede considerar la posibilidad de realizar espera ocupada cuando se efectúa una operación de E/S, a diferencia de un sistema operativo que opera sobre un solo procesador en donde no se realiza espera ocupada en el caso de una operación de E/S.

Aspectos de diseño:

Dentro de los principales aspectos que se deben considerar a la hora de diseñar un sistema operativo distribuido se encuentran:

- ✓ Transparencia.
- ✓ Flexibilidad.
- ✓ Confiabilidad.
- ✓ Desempeño.
- ✓ Escalabilidad.

Transparencia: La transparencia es un concepto que esta directamente relacionado con las acciones que realiza el sistema para ocultar al usuario y al programador que dicho sistema consiste en un conjunto de procesadores (o máquinas) que operan en tiempo compartido. Es decir, los usuarios, por ejemplo no necesitan saber si están conectados en la máquina X o en la Y, luego un comando del tipo rlogin no corresponde a un esquema de transparencia. El concepto de transparencia puede clasificarse en varios tipos diferentes, los cuales se presentan a continuación:

- ✓ Transparencia de localización.
- ✓ Transparencia de migración.
- ✓ Transparencia de copia o réplica.
- ✓ Transparencia de concurrencia.
- ✓ Transparencia de paralelismo.

La transparencia, también puede no ser requerida siempre, por ejemplo, un usuario puede en algunos casos preferir imprimir en una impresora particularmente (porque le queda mas cerca, por ejemplo).

Flexibilidad: Este aspecto tiene relación con que tan flexible puede ser un sistema. Dentro de este aspecto existen dos grandes distintas líneas de diseño. Entre ellas se tiene:

- ✓ Sistemas Monolíticos.
- ✓ Sistemas de Microkernel.

UNIX como sistema operativo no distribuido es un sistema monolítico, y los sistemas operativos distribuidos que se han construido en base a él han adoptado la misma filosofía. En cambio los sistemas operativos diseñados a partir de cero, de preferencia, han sido diseñados como sistemas de microkernel. La tendencia en el diseño de sistemas operativos distribuidos es la utilización de la visión de microkernel. El sistema Amoeba, es de tipo microkernel y el Sprite es monolítico.

Confiabilidad: La confiabilidad es un aspecto muy importante que debe considerarse en el diseño de sistemas distribuidos. Se supone que si el sistema distribuido esta compuesto por un conjunto de recursos que cooperan entre sí, entonces si un componente del sistema se descompone otro componente podría reemplazarlo. Un aspecto muy relacionado con la confiabilidad respecto a los descrito tiene que ver con la Tolerancia a fallas. Este aspecto suena mucho en el ambiente de investigación. Otro aspecto importante de mencionar respecto a la confiabilidad es la disponibilidad, este aspecto tiene que ver con la fracción de tiempo en que se puede utilizar el sistema. Unos de los mecanismos utilizados para aumentar la disponibilidad del sistema es la redundancia. Sin embargo, este mecanismo debe ser controlado muy bien para no caer en inconsistencias. La seguridad es otro aspecto clave relacionado con la confiabilidad, pues deben existir los mecanismos de protección adecuados para el control de la información de los usuarios y del propio sistema. En un sistema distribuido la seguridad es mas vulnerable que un sistema centralizado o de comutadores aislados o en red.

Desempeño: El desempeño es una motivación muy importante que conlleva el uso de sistemas distribuidos. El desempeño puede ser evaluado en base al tiempo de respuesta, al grado de flexibilidad, al grado de confiabilidad, etc. Sin embargo, el desempeño depende mucho de las capacidades de los canales de comunicación que interconectan a los diversos componentes del sistema. En un sistema multiprocesador, las velocidades de comunicación son mas altas respecto a las que se manejan en una LAN. Sin embargo, el desarrollo de tecnologías de comunicación de altas velocidades están apareciendo con gran fuerza en los últimos tiempos, como es el caso de ATM.

Escalabilidad: La escalabilidad tiene relación con la proyección del crecimiento de las redes, en donde día a día se incorporan nuevos servicios y nuevos usuarios que requieren los servicios. La escalabilidad tienen relación con la perspectiva que debe tener el sistema a dichos incrementos.

Ventajas de los Sistemas Distribuidos:

En general, los sistemas distribuidos (no solamente los sistemas operativos) exhiben algunas ventajas sobre los sistemas centralizados que se describen enseguida.

- ✓ **Economía:** El cociente precio/desempeño de la suma del poder de los procesadores separados contra el poder de uno solo centralizado es mejor cuando están distribuidos.
- ✓ **Velocidad:** Relacionado con el punto anterior, la velocidad sumada es muy superior.
- ✓ **Confiabilidad:** Si una sola máquina falla, el sistema total sigue funcionando.
- ✓ **Crecimiento:** El poder total del sistema puede irse incrementando al añadir

- ✓ pequeños sistemas, lo cual es mucho más difícil en un sistema centralizado y caro.
- ✓ **Distribución:** Algunas aplicaciones requieren de por sí una distribución física.

Por otro lado, los sistemas distribuidos también exhiben algunas ventajas sobre sistemas aislados. Estas ventajas son:

- ✓ **Compartir datos:** Un sistema distribuido permite compartir datos más fácilmente que los sistemas aislados, que tendrían que duplicarlos en cada nodo para lograrlo.
- ✓ **Compartir dispositivos:** Un sistema distribuido permite accesar dispositivos desde cualquier nodo en forma transparente, lo cual es imposible con los sistemas aislados. El sistema distribuido logra un efecto sinérgico.
- ✓ **Comunicaciones:** La comunicación persona a persona es factible en los sistemas distribuidos, en los sistemas aislados no. _ Flexibilidad: La distribución de las cargas de trabajo es factible en el sistema distribuidos, se puede incrementar el poder de cómputo.

Desventajas de los Sistemas Distribuidos:

Así como los sistemas distribuidos exhiben grandes ventajas, también se pueden identificar algunas desventajas, algunas de ellas tan serias que han frenado la producción comercial de sistemas operativos en la actualidad. El problema más importante en la creación de sistemas distribuidos es el software: los problemas de compartición de datos y recursos es tan complejo que los mecanismos de solución generan mucha sobrecarga al sistema haciéndolo ineficiente. El checar, por ejemplo, quiénes tienen acceso a algunos recursos y quiénes no, el aplicar los mecanismos de protección y registro de permisos consume demasiados recursos. En general, las soluciones presentes para estos problemas están aún en pañales. Otros problemas de los sistemas operativos distribuidos surgen debido a la concurrencia y al paralelismo. Tradicionalmente las aplicaciones son creadas para computadoras que ejecutan secuencialmente, de manera que el identificar secciones de código 'paralelizable' es un trabajo arduo, pero necesario para dividir un proceso grande en sub-procesos y enviarlos a diferentes unidades de procesamiento para lograr la distribución. Con la concurrencia se deben implantar mecanismos para evitar las condiciones de competencia, las postergaciones indefinidas, el ocupar un recurso y estar esperando otro, las condiciones de espera circulares y , finalmente, los "abrazos mortales" (deadlocks). Estos problemas de por sí se presentan en los sistemas operativos multiusuarios o multitareas, y su tratamiento en los sistemas distribuidos es aún más complejo, y por lo tanto, necesitará de algoritmos más complejos con la inherente sobrecarga esperada.

• Modelos de procesos distribuidos y multihilos (E)

• Sistemas en tiempo real (E)

Un sistema operativo en tiempo real procesa las instrucciones recibidas al instante, y una vez que han sido procesadas muestra el resultado. Este tipo tiene relación con los sistemas operativos monousuarios, ya que existe un solo operador y no necesita compartir el procesador entre varias solicitudes. Su característica principal es dar respuestas rápidas; por ejemplo en un caso de peligro se necesitarían respuestas inmediatas para evitar una catástrofe.

• Sistemas para computadoras paralelas (memoria compartida, memoria distribuida, memoria distribuida/compartida) (E)

Tiempo Compartido.

El tiempo compartido en ordenadores o computadoras consiste en el uso de un sistema por más de una persona al mismo tiempo. El tiempo compartido ejecuta programas separados de forma concurrente, intercambiando porciones de tiempo asignadas a cada programa (usuario). En este aspecto, es similar a la capacidad de multitareas que es común en la mayoría de los microordenadores o las microcomputadoras. Sin embargo el tiempo compartido se asocia generalmente con el acceso de varios usuarios a computadoras más grandes y a organizaciones de servicios, mientras que la multitarea relacionada con las microcomputadoras implica la realización de múltiples tareas por un solo usuario.

2. Ambientes gráficos

• Manejo de ventanas (E)

- Entornos multitarea (E)

C. Utilerías y manejadores

I. Orientados al usuario

1. Sistemas de respaldo y recuperación

- Compactación y descompactación (RP)

Compactación.

Una técnica empleada cuando la memoria está dividida en particiones de tamaño variable. De cuando en cuando, el sistema operativo desplaza las particiones para que queden contiguas y así toda la memoria libre esté reunida en un solo bloque.

La creación de áreas libres (huecos) entre particiones se denomina fragmentación externa, y aparece habitualmente en sistemas con asignación dinámica de bloques de memoria de tamaño variable. Si la memoria resulta seriamente fragmentada, la única salida posible es reubicar algunos o todas las particiones en un extremo de la memoria a así combinar los huecos para formar una única área libre grande, a esta solución se le llama *compactación*.

- Respaldos incrementales, periodicidad y confiabilidad (RP)

Respaldos incrementales

La forma más simple de estos vaciados es mediante un vaciado completo en forma periódica, por ejemplo, una vez al mes o a la semana y hacer un vaciado diario solo de los archivos modificados desde el último vaciado total. Para la implementación de este método, debe mantenerse en el disco una lista de los tiempos de vaciado de cada archivo. El archivo de vaciado verifica entonces cada uno de los archivos del disco. Si un archivo fue modificado desde su último vaciado, se le vacía de nuevo y su tiempo del último vaciado cambia por el tiempo actual. MS-Dos ayuda un poco en la realización de respaldar. A cada archivo se le asocia un bit de atributo llamado bit de biblioteca. Al hacer un respaldo del sistema de archivos, los bit de biblioteca de todos los archivos toman el valor de 0. Después, cuando se modifica un archivo, el sistema activa en forma automática su bit de biblioteca. Cuando es tiempo del siguiente respaldo, el programa de respaldo revisa todos los bits de biblioteca y solo respalda aquellos archivos cuyos bits de biblioteca están activos. También limpia todos esos bits para hacer una revisión posterior del uso de los archivos.

- Herramientas para reparación y recuperación (RP)

Una de las herramientas de recuperación son las bitácoras, esta técnica nos permite garantizar la disponibilidad continua de los datos. Esto es posible creando un archivo de bitácora de todas las transacciones copiándolas en otro disco. Esta redundancia puede resultar costosa, pero en caso de fallos es posible reconstruir todo el trabajo perdido.

Otra herramienta de recuperación son los respaldos incrementales, que mencionamos anteriormente.

2. Tratamiento de virus

Un virus es un programa que puede “infectar” a otros programas modificándolos, la modificación incluye una copia del programa de virus, que puede entonces seguir infectando a otros programas. Un virus informático porta en su código de instrucciones la receta para hacer copias perfectas de sí mismo. Una vez alojado en un computador anfitrión, el típico virus toma el control temporalmente del sistema operativo situado en el disco del computador. Entonces, cuando la computadora infectada entra en contacto con un elemento de software no infectado, se pasa una nueva copia del virus al programa. En un entorno de red, la capacidad de acceder a las aplicaciones y los servicios del sistema de otras computadoras ofrece un cultivo perfecto para la propagación de un virus.

La naturaleza de los virus

Un virus puede hacer cualquier cosa que hagan otros programas. La única diferencia es que se engancha a otro programa y se ejecuta de forma oculta cada vez que se ejecuta el programa anfitrión. Una vez que un virus se ejecuta, puede efectuar cualquier función, como borrar archivos y programas. Esta es la amenaza de los virus.

Durante su vida, un virus típico pasa por las siguientes cuatro etapas:

1. una fase latente, en la que el virus está inactivo. El virus será activado finalmente por algún suceso, como una fecha, la presencia de otro programa o archivo o que la capacidad del disco exceda de cierto límite. No todos los virus pasan por esta etapa.
2. una fase de propagación, durante la cual el virus sitúa una copia idéntica suya con otros programas o en ciertas zonas del sistema en el disco. Cada programa infectado contendrá ahora un clónico del virus, que entrará a su vez en la fase de propagación.
3. la fase de activación, en la que el virus se activa para llevar a cabo la función para la que esta propuesto. Como en la fase latente, la fase de activación puede ser causada por una variedad de sucesos del sistema, incluyendo la cuenta del número de veces que esta copia el del virus ha hecho copias de sí mismo.
4. la fase de ejecución, en la que se lleva a cabo la función. La función puede ser no dañina, como dar un mensaje por la pantalla, o dañina, como la destrucción de archivos de programas y datos.

La mayoría de los archivos llevan a cabo su trabajo de manera específica para un sistema operativo concreto y, en algunos casos, específicamente para una plataforma de hardware en particular. Así pues, están diseñados para sacar partido de los detalles y las debilidades de los sistemas concretos.

Infección Inicial

Una vez que un virus ha tenido acceso a un sistema por la infección de un solo programa, está en posición de infectar algunos o todos los archivos ejecutables del sistema, cuando se ejecute el programa infectado. Así pues, la infección vírica puede ser prevenida por completo impidiendo que el virus entre por primera vez. Por desgracia, la prevención es extremadamente difícil porque un virus puede formar parte de cualquier programa exterior a un sistema. Solo una pequeña parte de las infecciones tienen comienzo a través de conexiones de red. La mayoría de ellas se obtienen de un sistema de tablón de anuncios electrónico. Normalmente, un empleado traerá por la red un juego o una utilidad aparentemente útil para descubrir más tarde que tenía virus.

Métodos antivirus

La solución ideal para la amenaza de los virus es la prevención: en primer lugar, no permitir que los virus entren en el sistema. Esta meta es, en general, imposible de alcanzar, aunque la prevención puede reducir el número de ataques víricos fructuosos. El siguiente mejor método es ser capaz de hacer lo siguiente:

- ✓ Detección: una vez que se ha producido la infección, determinar que ha tenido lugar y localizar el virus.
- ✓ Identificación: una vez que se ha logrado la detección, identificar el virus específico que ha infectado un programa. Eliminar el virus de todos los sistemas infectados, de forma que la plaga no pueda extenderse más.
- ✓ Eliminación: una vez que se ha identificado el virus específico, eliminar todo rastro del virus del programa infectado y reponerlo a su estado original.

Si la detección tiene éxito, pero la identificación o la eliminación no son posibles, la alternativa de descartar el programa infectado y cargar una copia limpia de reserva.

Los avances en la tecnología de virus y antivirus van de la mano. Los primeros virus eran trozos de código relativamente simple y podían identificarse y liquidarse con paquetes antivirus relativamente sencillos. A medida que la carrera de armamentos de los virus ha avanzado, tanto los virus como, necesariamente, los antivirus han crecido en complejidad y sofisticación. Se identifican 4 generaciones de software antivirus:

- ✓ Primera Generación: rastreadores simples
- ✓ Segunda Generación: rastreadores heurísticos
- ✓ Tercera Generación: trampas de actividad
- ✓ Cuarta Generación: protección completa

Un rastreador de primera generación requiere una firma del virus para identificarlo. El virus puede contener "comodines", pero tiene básicamente la misma estructura y patrón de bits en todas las copias. Dichos rastreadores de firmas específicas están limitados a la detección de virus conocidos. Otro tipo de rastreadores de primera generación mantienen un registro de la longitud de los programas y buscan cambios de longitud.

Un rastreador de segunda generación no depende de una firma específica. Mas bien, el rastreador emplea reglas heurísticas para buscar infecciones probables por virus. Un tipo de tales rastreadores buscan trozos de código que suelen estar asociados con virus. Otro método de segunda generación es la prueba de integridad. Se puede añadir

un código de prueba (checksum) a cada programa. Si un virus infecta el programa sin cambiar el código, una prueba de integridad detectará el cambio. Para contrarrestar un virus suficientemente sofisticado, como para cambiar el código de prueba cuando infecta un programa, se puede emplear una función de dispersión (hash) cifrada. La clave de cifrado se almacena por separado del programa, de forma que el virus no pueda generar un nuevo código de dispersión y cifrarlo. Utilizando una función de dispersión en vez de un sencillo código de dispersión que antes.

Los programas de tercera generación son programas residentes en memoria que identifican un virus por sus acciones más que por la estructura de un programa infectado. Dichos programas tienen la ventaja de que no hace falta construir firmas y heurísticas para una amplia muestra de virus. Más bien, sólo es necesario identificar el pequeño conjunto de acciones que indican que se está intentando una infección y, en tal caso, intervenir.

Los productos de cuarta generación son paquetes que constan de una variedad de técnicas antivirus utilizadas en conjunto. Entre estos se incluyen posibilidades de control de acceso, que limitan la capacidad de los virus para penetrar en un sistema y, por tanto, limitan la capacidad de los virus para actualizar los archivos y contagiar la infección.

• **Prevención, detección y erradicación (E, RP)**

Las medidas preventivas más comunes incluyen el filtrado preventivo de todo software de recién adquisición, las copias de respaldo frecuentes y una combinación de utilidades tales como comprobadores de integridad, programas de vigilancia y supresores de virus.

1. Comprobadores de integridad.- Son programas que intentan detectar modificaciones en otros programas, mediante el cálculo y almacenamiento de sumas de chequeo en los programas ejecutables y en los archivos de órdenes. Se pretende que sean ejecutados con bastante frecuencia, por ejemplo en cada arranque. Puesto que la infección del virus modifica el tamaño del programa afectado.
2. Programas supervisores.- Son programas permanentemente residentes (demonios) que comprueban continuamente ciertas operaciones de memoria y de E/S para detectar comportamientos sospechosos que puedan ser atribuibles al virus. Los programas de vigilancia son rutinas adicionales al S.O que tratan de sellar los agujeros de seguridad conocidos o que puedan ser probablemente explorados por virus. Pueden ser capaces de detectar algunas formas de infección vírica antes de que alcancen la etapa de activación.
3. Los supresores de virus.- Son programas que examinan el almacenamiento del sistema buscando firmas y patrones que pertenecen a un virus conocido por el autor del programa. Pueden detectar y eliminar virus conocidos incluso en la etapa de letargo. Los supresores de virus no pueden tratar a virus desconocidos, estos se deben actualizar frecuentemente con el fin de adaptar su trabajo a los nuevos tipos de virus.

• **Reparación de archivos (E, RP)**

II. Orientados al sistema

1. Cargadores y ligadores

• **Manejo de ligaduras estáticas y dinámicas (E, RP)**

Concepto De Cargadores y Ligadores

Un ligador es un programa de sistema que combina dos o más programas objeto separados y permite que se hagan referencias unos a otros, o sea, que cada uno de estos programas puedan hacer referencia a código ó variables de los otros programas con los que está enlazado.

En muchos programas el cargador hace la labor del programa de enlace, por que existe solo un "linking loader" y no existe programa de enlace independiente. Es importante señalar que no se necesita un programa de enlace ni un cargador separado para cada traductor en el sistema, ya que estos programas trabajan con el programa objeto, sin importar el lenguaje fuente. Por otro lado es importante que cada compilador o ensamblador produzca el programa objeto usando el mismo formato. Cada programa objeto es un archivo de récord.

Un cargador es un programa que coloca en la memoria para su ejecución, el programa guardado en algún dispositivo de almacenamiento secundario.

Un cargador es un programa del sistema que realiza la función de carga, pero muchos cargadores también incluyen relocalización y ligado.

Algunos sistemas tienen un ligador para realizar las operaciones de enlaces y un cargador separado para manejar la relocalización y la carga.

Los procesos de ensamblado y carga están íntimamente relacionados.

El cargador consiste en un juego de instrucciones que permiten al dispositivo de entrada (teclado ó unidad de cinta) asignar la dirección de inicio de la memoria y asegurar que el computador leerá el programa y lo cargara byte a byte.

Funciones De Un Cargador y Un Ligador

Las funciones más importantes de un cargador son: colocar un programa objeto en la memoria e iniciar su ejecución. Si tenemos un cargador que no necesita realizar las funciones de ligado y relocalización de programas, su operación es simple pues todas las funciones se realizan en un solo paso. Se revisa el registro de encabezamiento para comprobar se ha presentado el programa correcto para la carga (entrando en la memoria disponible). A medida que lee cada registro de texto, el código objeto que contiene pasa a dirección de la memoria indicada. Cuando se encuentra el registro de fin, el cargador salta a la dirección especificada para iniciar la ejecución del programa cargado. Un programa objeto contiene instrucciones traducidas y valores de datos del programa fuente y específica direcciones en memoria dónde cargarán estos elementos.

Carga que lleva el programa objeto a la memoria para su ejecución.

Relocalización que modifica el programa objeto de forma que puede cargarse en una dirección diferente de la localidad especificada originalmente.

Ligado que combina dos o más programas objeto independientes y proporciona la información necesaria para realizar diferencias entre ellos.

El cargador es un programa del sistema que realiza la función de carga pero muchos cargadores también incluyen relocalización y ligado. Algunos sistemas tienen un ligador (ó editor de ligado) para realizar las operaciones de enlace, y un cargador separado para manejar la relocalización y la carga.

En la mayoría de los casos todos los traductores de programas (esto es, ensambladores y compiladores) de un sistema en particular producen programas objeto en el mismo formato.

De esta misma forma, puede usarse el cargador o ligador del sistema con independencia del lenguaje de programación fuente original, se suele utilizar el término cargador en lugar de cargador y ligador, los procesos de ensamblado y carga están íntimamente relacionados entre sí.

Se han desarrollado herramientas especiales de software, llamadas cargadores para asistir, al programados en la carga del programa. El cargador es normalmente un programa pequeño que permite al usuario entrar directamente las palabras de instrucción y datos a direcciones concretas de la memoria mediante, ó un teclado ó una cinta magnética.

Tipos de cargadores y Ligadores

Hay diferentes tipos de cargadores como:

- ✓ **Cargadores Bootstrap:** el programa cargador una vez, situado en la memoria del computador, cargará el programa de aplicación y los datos. Pero, previamente, se ha debido cargar el cargador en la memoria. Y esto se puede realizar por los siguientes métodos:

1. *Entrada manual:* mediante el teclado el usuario teclea el cargador BOOTSTRAP. Despues de esto, el cargador se carga así mismo en la memoria del computador.
2. *Entrada por ROM:* es posible tener las instrucciones de inicialización almacenados permanentemente en alguna porción de la ROM, en lugar de introducirlas manualmente por teclado o por panel frontal. Cuando se requiere el programa de bootstrapping, el operador simplemente dirige al computador, mediante los interruptores del panel, a ejecutar las instrucciones memorizadas en ROM: al estar el programa almacenado en ROM se elimina también la posibilidad de borrados accidentales.

- ✓ **Cargadores iniciales:** indican a la computadora la forma de poner, dentro de la memoria principal unos datos que están guardados en un periférico de memoria externa (cinta, disco, etc.). Sirven para cargar en la memoria pequeños programas que inician el funcionamiento de una computadora.
Algunas computadoras de carácter general no tienen en memoria ningún programa de forma permanente y cuando se desconectan pierden toda la información de su memoria interna. Al volverlos a conectar no son capaces de controlar ningún periférico. Se hace así para que sea el usuario el que ponga los programas que le interese ejecutar.
- ✓ **Cargadores absolutos:** el programa cargador pone en memoria las instrucciones guardadas en sistemas externos. Independientemente de que sea un cargador inicial, o no sin dichas instrucciones se almacenan siempre en el mismo espacio de memoria (cada vez que se ejecuta el programa cargador) se dice que es un cargador absoluto.
- ✓ **Cargadores con reubicación:** en ocasiones un mismo programa necesita ejecutarse en diferentes posiciones de memoria. Para esto la traducción debe estar realizada en forma adecuada, es decir no utilizando referencias absolutas a direcciones en memoria, sino referencias a una dirección especial llamada de reubicación.
- ✓ **Cargadores ligadores:** conocidos también como linker. Un linker es un término en inglés que significa montar.

Montar un programa consiste en añadir al programa objeto obtenido a la traducción las rutinas externas a las que hace referencia dicho programa. El ensamblador debe permitir dichas referencias y las rutinas deben estar a su vez en lenguaje máquina guardadas en algún elemento accesible por el montador. Generalmente, dichas rutinas se encuentran guardadas en un fichero especial al que suele denominarse librería porque están almacenadas todas las rutinas externas susceptibles de ser utilizadas por los diferentes programas del usuario. Allí va el programa ligador cuando esta realizando el montaje de un programa a buscarlas y las adjunta al programa objeto.

- ✓ **Editores de ligado:** la diferencia fundamental entre un editor de ligado y un cargador ligador es:

Primero.- se ensambla o compila el programa fuente, produciendo un programa objeto (que puede contener varias secciones de control diferentes). Una cargador ligador realiza todas las operaciones de ligado y relocalización incluyendo búsqueda automática en bibliotecas, si se especifica, y carga el programa ligado directamente en la memoria para su ejecución. Por otro lado un editor de ligado produce una versión ligada del programa (llamada a menudo modulo de carga ó imagen ejecutable) que se escribe en un archivo o biblioteca para su ejecución posterior.

Cuando el usuario está listo para ejecutar el programa ligado, se puede utilizar un cargador relocalizador simple para cargar el programa en la memoria. La única modificación necesaria al código objeto es la suma de una dirección de carga real a los valores relativos del programa. El editor de ligado realiza la relocalización de todas las secciones de control al inicio del programa ligado. De esta forma todos los elementos que necesitan modificarse en el momento de la carga tienen valores relativos al inicio del programa ligado.

Los editores de ligado se pueden utilizar para construir paquetes de subrutinas u otras secciones de control que se suelen utilizar juntas. Esto puede ser útil al tratar con bibliotecas de subrutinas que manejan lenguajes de programación de alto nivel. A veces permiten al usuario especificar que las referencias externas no se resuelven por búsqueda automática en biblioteca.

- ✓ **Ligado dinámico:** ofrece algunas ventajas sobre los tipos de ligado. Proporciona la posibilidad de cargar las rutinas sólo cuando si se necesitan. Si las subrutinas son grandes ó tienen muchas referencias externas se pueden conseguir ahorros considerables de tiempo y espacio en memoria.

El ligado dinámico evita la necesidad de cargar la biblioteca completa para cada ejecución. Puede incluso hacer innecesario que el programa conozca el conjunto de subrutinas que se podría utilizar. El nombre de la subrutina se trataría simplemente como otro elemento de entrada. En el método que se utilice aquí las rutinas que se carguen dinámicamente deben llamarse por medio de una solicitud del servicio al sistema operativo. Este método también podría considerarse como una solicitud a una parte del cargador que se mantiene en la memoria durante la ejecución del programa.

Cuando se utiliza ligado dinámico, la asociación de dirección real y el nombre simbólico de la rutina llamada no se hace hasta que se ejecuta la proposición llamada.

- **Resolución de direcciones y referencias externas (RP)**

Un programa llamado cargador realiza las dos funciones de carga y edición de enlace (ligadura). El proceso de carga consiste en tomar el código de maquina relocalizable, modificar las direcciones relocalizables y ubicar las instrucciones y los datos modificados en las posiciones apropiadas de la memoria.

El editor de enlace permite formar un solo programa a partir de varios archivo de código maquina relocalizable. Estos archivos pueden haber sido el resultado de varias compilaciones distintas, y uno o varios de ellos pueden ser archivos de biblioteca de rutinas proporcionadas por el sistema y disponibles para cualquier programa que las necesite.

Si los archivos se van a usar juntos de manera útil, puede haber algunas referencias externas, en las que el código de un archivo hace referencia a una posición de otro archivo. Esta referencia puede ser a una posición de datos definida en un archivo y utilizada en otro, o puede ser el punto de entrada de un procedimiento que aparece en el código de un archivo y se llama de otro. El archivo con el código de maquina relocalizable debe conservar la información de la tabla de símbolos para cada posición de datos o etiqueta de instrucción a la que se hace referencia externamente.

2. Administración y vigilancia

- **Bitácoras (RP)**

- **Detección de errores físicos (E)**

Una fuente frecuente de errores físicos son los dispositivos periféricos electromecánicos, que proveen entrada o salida a la CPU. Los componentes mecánicos de estos dispositivos periféricos son probables de llevar a cabo un error resultado de una presión de uso. Algunos de estos dispositivos son frágiles, consecuentemente una fuente potencial de errores. El debilitamiento de estos medios puede no ser fatal, porque las unidades periféricas pueden ser incapacitadas. Los componentes modernos son diseñados para monitorear su propio desempeño y constantemente realizan un test por ellos mismos, para asegurar que cada operación sea ejecutada correctamente.

- **Manejo de suspensiones de energía e interrupciones de servicio (E)**

- **Herramientas para arranque y manejo de recursos físicos (E)**

- **Herramientas para diagnóstico (RP)**

Se cuenta con dos tipos generales de herramientas de diagnóstico asistidas por computadora para ayudar a detectar y aislar problemas que podrían ocurrir en el sistema:

- 1.- Herramienta de diagnóstico interno que corre en marcha, la puesta en cero o la autoevaluación.
- 2.- Un disquete de prueba de diagnóstico que corre una serie de programas de prueba desde el disquete.

- **Reinicio de tareas (E, RP)**

- **Puntos de reinicio (E, RP)**

Las características de punto de reinicio/verificación disponibles en muchos sistemas facilitan la suspensión/reanudación solo con la pérdida de trabajo desde el punto de verificación (la última grabación del estado del sistema). Pero muchos sistemas de aplicación se diseñan sin aprovechar las ventajas de las funciones de reinicio/verificación.

3. Herramientas para la administración

- **Uso de recursos físicos:** manejadores de memoria, para optimización de espacio en disco, para dispositivos de comunicaciones (RP)
- **Instalación y arranque:** definición de entornos iniciales, sistemas para instalación automática, autoinstalación (RP)
- **Manejadores para supervisión y diagnóstico de dispositivos físicos (E, RP)**

PROGRAMACIÓN E INGENIERÍA DE SOFTWARE

A. Algorítmica

I. Representación y manipulación de datos

1. Taxonomía de las estructuras de datos

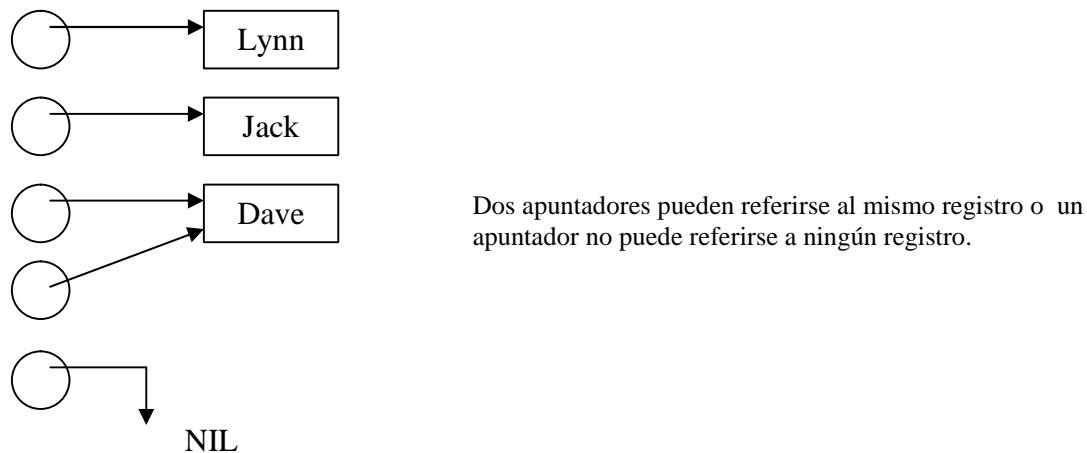
• Apuntadores (R, E, RP)

Un apuntador (también llamado liga) se define como una variable que indica la localización de alguna otra variable; por ejemplo: alguna sección de datos. Así, un apuntador de pila indica la localización del elemento en el tope superior de la pila y los apuntadores de cola dan los sitios de la cabeza y el extremo de la cola.

Estructuras dinámicas de datos.

Este tipo de estructura es generado a partir de un tipo de dato conocido con el nombre de puntero o de referencia. Un dato puntero almacena una dirección o referencia a un dato. Por tanto, debe distinguirse entre un dato puntero y el dato al cual se apunta. Se usará la notación $P = ^D$ para indicar que P es un puntero a datos de tipo D . Se genera un valor para una variable de tipo puntero cuando se asigna dinámicamente memoria a un dato apuntado por ella. La principal ventaja de los punteros o apuntadores es que se puede adquirir posiciones de memoria que se necesitan, y liberarlas cuando ya no se requieran. De esta manera se pueden crear estructuras dinámicas que se expandan o se contraigan, según se les agregue o elimine elementos. El dinamismo de estas estructuras soluciona el problema de decidir cuál es la cantidad óptima de memoria que debe reservarse para un problema dado. Al proceso que asigna y reasigna la memoria en esta forma se le conoce como asignación dinámica de memoria.

Estructura con apuntadores Nil



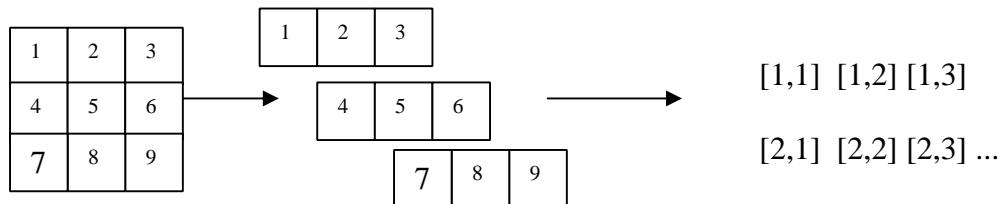
• Vectores, matrices, cubos, hipercubos (R, E, RP)

Vectores o Arreglos

Un arreglo o vector contiene un número constante de posiciones, es decir su tamaño se fija cuando el programa se compila. Arreglo o vector el cual contiene un número constante de posiciones. Los arreglos también se corresponden directamente con los vectores, término matemático utilizado para las listas indexadas de objetos. Análogamente, los arreglos bidimensionales se corresponden con las matrices.

Arreglos Rectangulares

Casi todos los lenguajes de alto nivel proporcionan medios convenientes y eficaces para almacenarlos y acceder a ellos. El almacenamiento en la computadora se arregla en una secuencia contigua, es decir, en una línea recta y cada entrada sigue a la otra. La forma de leer un arreglo rectangular, es leer las entradas del primer renglón de izquierda a derecha, después las entradas del segundo renglón así hasta terminar de leer el último renglón.



Este es el orden en que la mayoría de los compiladores almacenan un arreglo rectangular, y se denomina ordenamiento por renglón mayor.

Matrices

Arreglos Triangulares

Una matriz triangular inferior puede definirse formalmente como un arreglo cuadrado en el cual la entrada es 0 en cada posición donde el índice de columna sea mas grande que el índice del renglón.

$\begin{pmatrix} x & & \\ . & x & 0 \\ . & & x \\ . & & & x \\ xx & \dots & \dots & x \end{pmatrix}$	$\begin{pmatrix} xx & & \\ xxx & 0 & \\ xxx & & \\ xxx & & \\ 0 & xxx & \\ & & xx \end{pmatrix}$	$\begin{pmatrix} x & \dots & x & \\ x & . & & \\ x & . & . & \\ 0 & x & . & \\ & & x & \end{pmatrix}$	$\begin{pmatrix} x & & \\ xx & 0 & \\ xx & & \\ 0 & xx & \\ & & x \end{pmatrix}$
Matriz Triangular Inferior	Matriz Tridiagonal	Matriz Triangular Superior	Matriz Diagonal

La diagonal principal de una matriz cuadrada se compone de las entradas para las cuales los índices de renglón y de columna son iguales. Una matriz Diagonal es una matriz cuadrada en la que todas las entradas que no están sobre la diagonal principal son ceros. Una matriz tridiagonal es una matriz cuadrada en la que todas las entradas son cero excepto posiblemente aquellas sobre la diagonal principal y sobre las diagonales inmediatamente arriba y debajo de ella. Es decir, T es una matriz tridiagonal si $T_{ij} = 0$ a menos que $|i - j| \leq 1$. La transpuesta de una matriz es la matriz obtenida al intercambiar sus renglones por columnas correspondientes. Es decir, que la matriz B sea la transpuesta de la matriz A significa que $B_{j,i} = A_{i,j}$ para todos los índices (permitidos) i y j. Una matriz triangular superior es aquella en que todas las entradas bajo la diagonal principal son cero. La transpuesta de una matriz triangular inferior será una matriz triangular superior.

Matrices Simétricas y Antisimétricas

Una matriz A de $n \times n$ elementos es simétrica si $A_{i,j} = A_{j,i}$, y esto se cumple para todo i y para todo j.

$$\begin{pmatrix} 0 & 4 & 0 & 0 \\ 4 & 0 & 0 & 6 \\ 0 & 0 & 8 & 9 \\ 0 & 6 & 9 & 0 \end{pmatrix} \quad \text{Matriz simétrica}$$

una matriz A de $n \times n$ elementos es antisimétrica si $A_{i,j} = -A_{j,i}$, y esto se cumple para todo i y para todo j; considerando a i \neq j.

$$\begin{pmatrix} 0 & 4 & 0 & 0 \\ -4 & 0 & 0 & 6 \\ 0 & 0 & 8 & 9 \\ 0 & -6 & -9 & 0 \end{pmatrix} \quad \text{Matriz antisimétrica}$$

para almacenar en un arreglo unidimensional una matriz simétrica, se puede hacer almacenando solamente los elementos de una matriz triangular inferior o superior; por ejemplo: almacenando la matriz triangular inferior .

A [1,1] A [2,1] A [2,2] A [3,1] A [3,2] A [3,3] A [4,1] A [4,2] A [4,3] A [4,4]

0	4	0	0	0	8	0	6	9	0
---	---	---	---	---	---	---	---	---	---

1 2 3 4 5 6 7 8 9 10

por otra parte, para almacenar una matriz antisimétrica, se procede de la misma forma almacenando solamente los elementos de la matriz triangular inferior o superior.

A [1,1] A [1,2] A [1,3] A [1,4] A [2,2] A [2,3] A [2,4] A [3,3] A [3,4] A [4,4]

0	4	0	0	0	0	6	8	9	0
---	---	---	---	---	---	---	---	---	---

1 2 3 4 5 6 7 8 9 10 i >= j

- **Listas: simples, ligadas, doblemente ligadas, colas, pilas (stacks), colas de prioridades (R, E, RP)**

Listas

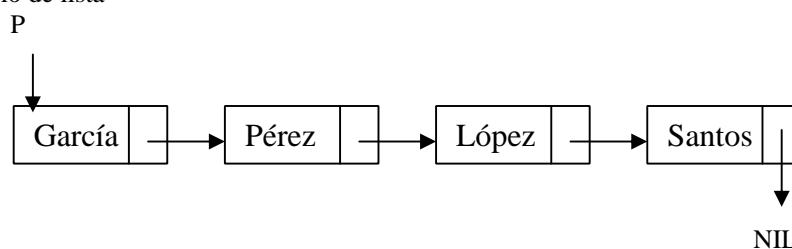
Una lista es una colección de elementos llamados generalmente nodos. El orden entre los nodos se establece por medio de punteros, es decir, direcciones o referencias a otros nodos. Un nodo consta de dos partes:

- ✓ Un campo información que será del tipo de datos que se quiera almacenar en una lista.
- ✓ Un campo liga de tipo puntero que se utiliza para establecer la liga o el enlace con otro nodo de la lista.

Si el nodo fuera el ultimo de la lista este campo tendrá como valor: nil (vacío).

Al emplearse el campo liga para relacionar dos nodos no es necesario almacenar físicamente a los nodos en espacios contiguos.

Ejemplo de lista

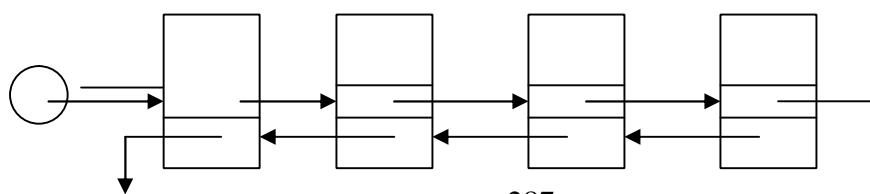


El primer nodo de la lista esta apuntado por una variable P, de tipo puntero (P almacena la dirección del primer nodo). El campo liga del ultimo nodo de la lista tiene un valor nil que indica que dicho nodo no apunta a ningún otro. Las operaciones que pueden llevase acabo en una lista son:

- ✓ Recorrido de la lista.
- ✓ Inserción de un elemento.
- ✓ Borrado de un elemento.
- ✓ Búsqueda de un elemento.

Lista doblemente ligada

Una lista doblemente ligada es aquella en que cada nodo tiene dos ligas, una para el siguiente nodo de la lista y otra para el nodo precedente. De este modo es posible moverse en cualquier dirección atraves de la lista conservando solo un apuntador.



Lista doblemente ligada con apuntadores.

Colas

La cola se define como una lista en la que todas las adiciones a la lista se hacen a un extremo y todas las eliminaciones se efectúan en el otro extremo. Las colas se llaman también primero en entrar, primero en salir (first-in-first-out) o PEPS (FIFO). Al elemento en una cola lista para manejarse, es decir, el primer elemento que se removerá de la cola, lo llamamos cabeza de cola o frente de la cola, y el último elemento, es decir al más reciente agregado, extremo o fondo de la cola. Las operaciones de una cola son:

- ✓ Determinar si la cola esta vacía o no.
- ✓ Recobrar el primer nodo de la cola, si no esta vacía.
- ✓ Insertar un nuevo nodo después del último nodo de la cola.
- ✓ Eliminar el primer nodo de la cola, si no esta vacía.

Pilas

La pila se define como una lista en la que todas las inserciones y eliminaciones se hacen por un extremo, llamado tope de la pila. Cuando agregamos un elemento a la pila decimos que se le inserta “empujándolo” en la pila y cuando sacamos un elemento decimos que se le extrae de la pila. Otro nombre que se usa para en ocasiones para designar pilas es el de listas UEPS (LIFO), basado en la propiedad último en entrar primero en salir (last in, first out).



Maxpila es una constante que da el máximo tamaño permitido de la pila.

Type pila = record

```
Tope: 0 ....maxpila;
Entrada: array [ 1 . . . maxpila] of elemento
```

End;

Cuando queremos insertar un elemento en una pila llena ocurre un desbordamiento y cuando queremos extraer un elemento de una pila vacía es insuficiencia. Las operaciones de una pila son:

- ✓ Determinar si la pila esta vacía o no.
- ✓ Recobrar el ultimo nodo de la pila, si no esta vacía.
- ✓ Insertar un nuevo nodo después del ultimo nodo de la pila.
- ✓ Eliminar el ultimo nodo de la pila, si no esta vacía.

Colas de prioridad

Una cola de prioridad es una estructura de datos con solo dos operaciones:

1. insertar un elemento.
2. suprimir el elemento que tenga la llave más grande (o la más pequeña).

Si los elementos tienen llaves iguales, la regla habitual establece que el primer elemento introducido deberá ser el primero en eliminarse. Por ejemplo: en un sistema de cómputo de tiempo compartido, un extenso número de tareas puede estar esperando para hacer uso de la CPU. Una de ellas tendrá mayor prioridad que otras. De ahí que el conjunto de las que esperan formen una cola de prioridad.

• Arboles: binarios, n-arios (R, E, RP)

Árboles

Árboles Balanceados

Se define un árbol balanceado como un árbol binario de búsqueda en el cual se debe cumplir la siguiente condición:

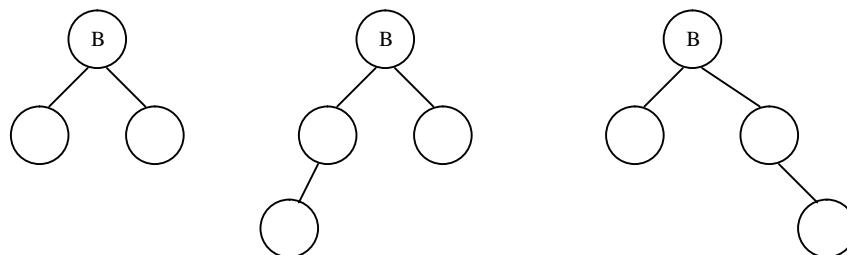
Para todo nodo T del árbol, la altura de los subárboles izquierdo y derecho no debe diferir en más de una unidad.

Estos árboles también reciben el nombre de AVL en honor a sus inventores, dos matemáticos rusos, G.M.Adelson-Velski y E.M.Landis. La idea de estos árboles es realizar reacomodos o balanceos después de inserciones o eliminaciones de elementos.

Inserción en árboles平衡ados

Al insertar un elemento en un árbol balanceado se deben seguir los casos:

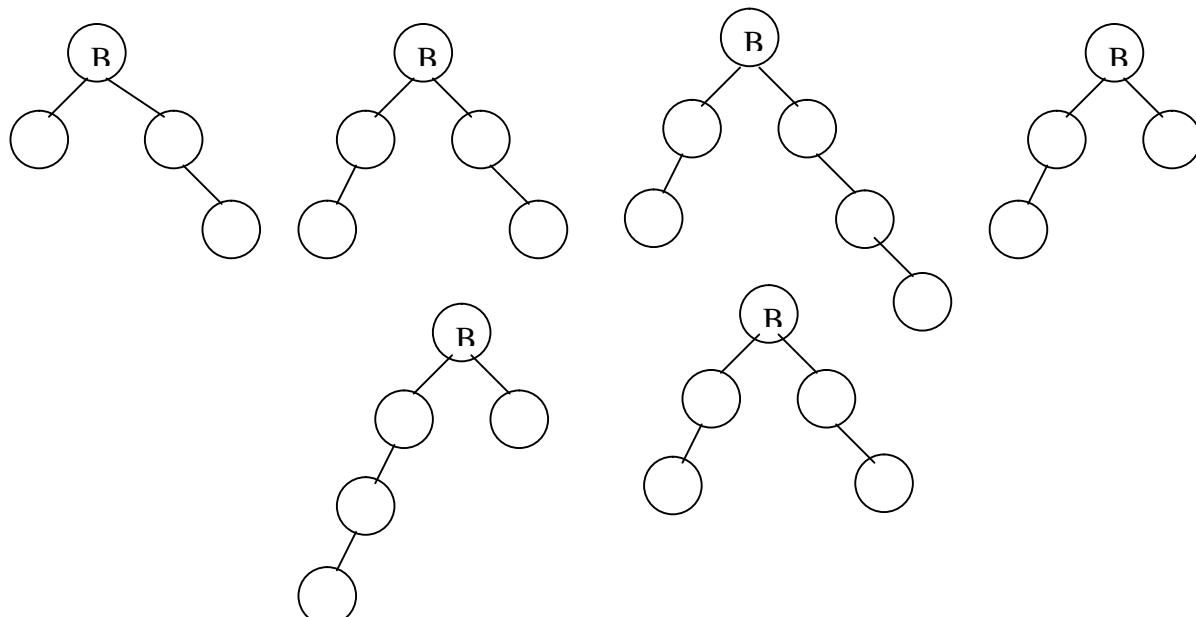
1. Las ramas izquierdo (RI) y derecho (RD) del árbol tienen la misma altura ($HRI = HRD$), por tanto:
 - ✓ si se inserta un elemento en RI entonces HRI será mayor a HRD
 - ✓ si se inserta un elemento en RD entonces HRD será mayor a HRI



2. Las ramas izquierdas (RI) y derechas (RD) del árbol tienen diferente altura ($HRI \neq HRD$). Supóngase que $HRI < HRD$
 - ✓ Si se inserta un elemento en la rama izquierda RI entonces $HRI = HRD$ {las alturas de ambas ramas son iguales por lo que se mejora el equilibrio}
 - ✓ Si se inserta en RD, entonces se rompe el equilibrio y será necesario reestructurarlo.

Supongase $HRI > HRD$:

- ✓ Si se inserta en RI, entonces se rompe el criterio de equilibrio y será necesario reestructurar el árbol.
- ✓ Si se inserta un elemento en RD, entonces $HRD = HRI$. Por lo que las ramas tendrán la misma altura mejorando el equilibrio del árbol.



Borrado en árboles平衡ados

Consiste en quitar un nodo sin violar los principios del árbol balanceado. La operación de borrado debe distinguir los siguientes casos:

1. Si el elemento a borrar es terminal u hoja, simplemente se suprime.

2. Si el elemento a borrar tiene un solo descendiente entonces, tiene que sustituirse por ese descendiente.
 3. Si el elemento a borrar tiene dos descendientes, entonces tiene que sustituirse por el nodo que se encuentra mas a la izquierda en el subárbol o por el nodo que se encuentra mas a la derecha en el subárbol izquierdo.
- Para poder determinar si un árbol esta balanceado se debe calcular su factor de equilibrio de un nodo (FE) que se define como:

La altura del subárbol derecho menos la altura del subárbol izquierdo.

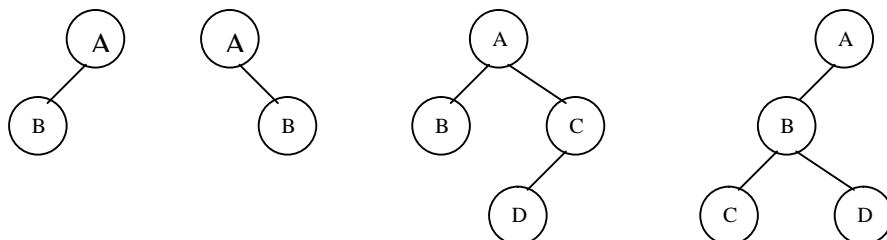
$$FE = HRD - HRI$$

Árboles Binarios

En un árbol binario cada nodo puede tener como máximo dos subárboles; y siempre es necesario distinguir entre el subárbol izquierdo y el derecho. Se define un árbol binario de tipo T como una estructura homogénea que es la concatenación de un elemento de tipo T llamado raíz, con dos árboles binarios disjuntos, llamados subárbol izquierdo y derecho. Los árboles binarios también se conocen como árboles ordenados de grado dos. Un árbol ordenado es aquel en el las ramas de los nodos del árbol están ordenadas.

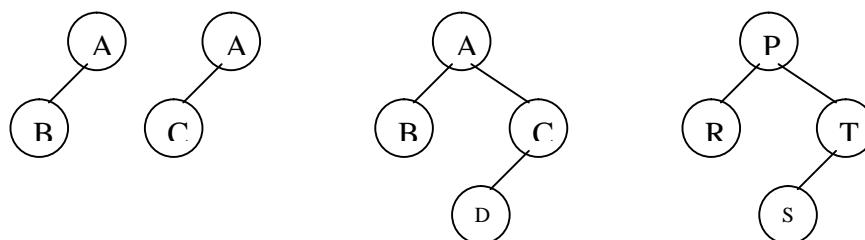
Árboles Binarios Distintos

Dos árboles binarios son distintos cuando sus estructuras son diferentes.



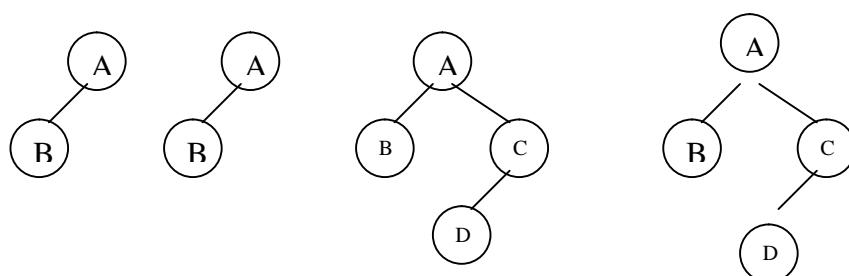
Árboles Binarios Similares

Dos árboles binarios son similares cuando sus estructuras son idénticas, pero la información que contienen en sus nodos difiere entre sí.



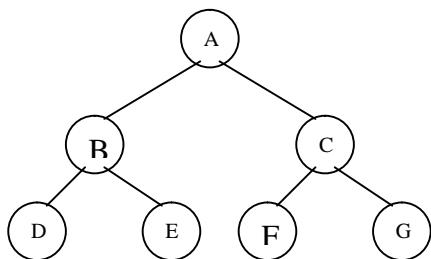
Árboles Binarios Equivalentes

Los árboles binarios equivalentes se definen como aquellos que son similares y además los nodos contienen la misma información.



Árboles Binarios Completos

Se define un árbol binario completo como un árbol en el que todos sus nodos, excepto los del ultimo nivel, tienen dos hijos; el subárbol izquierdo y el derecho.



Recorridos en Árboles Binarios

Recorrer significa visitar los nodos del árbol en forma sistemática; de tal manera que todos los nodos sean visitados una sola vez. Existen tres formas diferentes de efectuar un recorrido.

En preorden.

- ✓ Visitar la raíz.
- ✓ Recorrer el subárbol izquierdo.
- ✓ Recorrer el subárbol derecho.

En inorden.

- ✓ Recorrer el subárbol izquierdo.
- ✓ Visitar la raíz.
- ✓ Recorrer el subárbol derecho.

En postorden.

- ✓ Recorrer el subárbol izquierdo.
- ✓ Recorrer el subárbol derecho.
- ✓ Visitar la raíz.

Algoritmos.

Preorden (nodo)

{Recorrido preorden en un árbol binario. Nodo es un dato de tipo puntero} {INFO, IZQ., DER. Son campos del registro nodo. INFO es una variable de tipo carácter IZQ. Y DER. Son variables de tipo puntero}

1. si NODO != NIL entonces
 - visitar el nodo { NODO^.INFO }
 - regresar a preorden con NODO^.IZQ.
 - {llamada recursiva a preorden con la rama izquierda del nodo en cuestión}
 - regresar a preorden con NODO^.DER.
 - {llamada recursiva a preorden con la rama derecha del nodo en cuestión}
2. {fin del condicional del paso 1}

Arboles m-arios

W.I. Landauer sugirió la construcción de un árbol m-ario obligando que el nivel L estuviera casi completamente lleno antes de que apareciera nada en el nivel L +1; Landauer supuso que teníamos que buscar elementos en el árbol mas a menudo de lo que necesitábamos insertarlos o eliminarlos. Un árbol B de orden m es un árbol que satisface las siguientes propiedades:

- i) Cada nodo tiene $\leq m$ hijos
- ii) Cada nodo, excepto la raíz y las hojas, tienen $\geq m/2$ hijos
- iii) La raíz tiene por lo menos dos hijos
- iv) Todas las hojas aparecen en el mismo nivel y no llevan información
- v) Un nodo no hoja con k hijos contiene k-1 claves.

• Montículos (**heaps**), balanceados, árboles-AVL (R, E, RP)

Montículos binarios (heaps)

Veamos otro tipo especial de árbol binario, los llamados heaps (montículos), que se pueden representar eficazmente con un vector. Definición: un montículo de máximos (mínimos), o max heap (min heap), es un árbol binario completo tal que, el valor de la clave de cada nodo es mayor (menor) o igual que las claves de sus nodos hijos (si los tiene). De la definición se deduce que la clave contenida en el nodo raíz de un montículo de máximos es la mayor clave del árbol, pero esto no quiere decir que sea la única con ese valor. Análogamente sucede con los montículos de mínimos y la menor clave del árbol. La estructura del montículo tiene interesantes aplicaciones, por ejemplo, la ordenación de arrays (algoritmo heapsort) o el mantenimiento de las llamadas colas de prioridad. Las colas de prioridad son un tipo especial de colas donde todo elemento que se inserta en la cola lleva asociado una prioridad, de forma que el elemento que se borra de la cola es el primero entre los que tengan la máxima prioridad (en montículos de máximos). Los montículos, como árboles binarios completos que son, se pueden representar de forma eficaz mediante una estructura secuencial (un array). A cada nodo del árbol se le asigna una posición dentro de la secuencia en función del nivel del árbol en el que se encuentra y dentro de ese nivel de la posición ocupada de izquierda a derecha. De forma que para representado por el elemento $A[i]$ se cumplen las propiedades que vimos al principio del tema para árboles completos, es decir, el nodo padre estará localizado en $A[i \text{ div } 2]$, si $i > 1$, y los hijos estarán localizados en $A[2i]$ y $A[2i+1]$. Una característica fundamental de esta estructura de datos es que la propiedad que caracteriza a un montículo puede ser restaurada eficazmente tras cualquier modificación de un nodo (cambio de su valor, inserción, borrado, etc). Supongamos que se trata de un montículo de máximos (cualquier comentario sobre este tipo de montículos se puede extender fácilmente a los montículos de mínimos). En este caso, se debe cumplir que la clave de cualquier nodo sea mayor o igual que las claves de sus hijos. Si se modifica el valor de un nodo incrementando su valor, entonces es posible que la propiedad del montículo no se cumpla con respecto a sus nodos antecesores, no respecto a sus hijos. Esto implicaría ir ascendiendo por el árbol comprobando si el nodo modificado es mayor que el padre, si es así intercambiarlos y repetir el proceso en un nivel superior hasta que se encuentre que no es necesario realizar ningún intercambio o que se llegue al nodo raíz. En cualquier caso, tras ese proceso se habrá restaurado la propiedad del montículo. Si por el contrario, modificamos un nodo disminuyendo su valor, el problema consistirá en comprobar la propiedad respecto a sus descendientes. Si la clave del nodo es mayor que las claves de sus hijos, entonces la propiedad se sigue cumpliendo, si no es así, hay que intercambiar la clave de este nodo con la del hijo que tenga la clave máxima y repetir todo el proceso desde el principio con este nodo hijo hasta que no se realice el intercambio. Estas dos posibilidades de restauración del montículo se pueden expresar en forma de los dos siguientes algoritmos:

Algoritmo Subir

```

Entradas
p: arbol[1..n]
i: indice
Inicio
  si (i > 1) Y (A[i] > A[i div 2]) entonces
    intercambiar(A[i], A[i div 2])
    Subir(A, i div 2)
  fin_si
Fin

```

Algoritmo Bajar

```

Entradas
p: arbol[1..n]
i: indice
Variable
max: indice
Inicio
  max <-- i
  si (2i <= n) Y (A[2i] > A[max]) entonces
    max <-- (2i)
  si (2i + 1 <= n) Y (A[2i + 1] > A[max]) entonces
    max <-- (2i + 1)
  si (max <> i) entonces
    intercambiar(A[i], A[max])

```

```

Bajar(A, max)
fin_si
Fin

```

A partir de estos dos algoritmos básicos que permiten restaurar la propiedad del montículo, se pueden definir las operaciones de inserción y borrado de elementos en esta estructura de datos.

Inserción

Al intentar añadir un nuevo elemento al montículo, no se sabe cuál será la posición que ocupará el nuevo elemento de la estructura, pero lo que si se sabe es que, por tratarse de un árbol completo, dónde se debe enlazar el nuevo nodo, siempre será en la última posición de la estructura. Por ejemplo, en el siguiente montículo formado por cinco elementos, si se intenta añadir un nuevo elemento, sea el que sea, se conoce cuál será la estructura final del árbol: Si se almacena en el nuevo nodo la información que se desea insertar, lo más probable será que la propiedad del montículo no se conserve. Como el nuevo nodo siempre es terminal, para mantener la propiedad del montículo bastará con aplicar el algoritmo Subir a partir del nodo insertado. Con esto, el algoritmo de inserción resulta verdaderamente simple:

Algoritmo Insertar

```

Entradas
A: arbol[1..n]
x: Valor
Inicio
si (A.num < MAX_NODOS) entonces
    A.num <-> A.num + 1
    A[n] <-> x
    Subir(A, n)
fin_si
Fin

```

Borrado

Al igual que en la operación de inserción, hay que tener en cuenta que el montículo es un árbol completo y que, para que se mantenga esa propiedad, el nodo que debe desaparecer realmente es el último. Por lo tanto, la estrategia a seguir para borrar cualquier nodo del montículo podría ser: sustituir la información del nodo a borrar por la del último nodo del montículo y considerar que el árbol tiene un nodo menos; como esta modificación seguramente habrá alterado la propiedad del montículo, entonces aplicar el algoritmo Subir o Bajar, lo que sea preciso, para restaurar esa propiedad. La aplicación de cualquiera de los algoritmos dependerá de que la modificación de la información del nodo haya implicado un incremento o disminución de la clave.

Algoritmo Borrar

```

Entradas
p: arbol[1..n]
i: indice
Variable
x: Valor
Inicio
x <-> A[i]
A[i] <-> A[n]
n <-> n - 1
si (A[i] <> x) entonces
    si (A[i] > x) entonces Bajar(A, i)
    sino Subir(A, i)
fin_si
Fin

```

Aplicación de los montículos

Los montículos tienen como aplicación fundamental el mantenimiento de las llamadas colas de prioridad. Esta estructura de datos es un tipo especial de cola, donde cada elemento lleva asociado un valor de prioridad, de forma que cuando se borra un elemento de la estructura éste será el primero de los elementos con la mayor prioridad. En cualquier instante se puede insertar un elemento con prioridad arbitraria. Si la utilización de la cola de prioridad

requiere borrar el elemento con la mayor prioridad, entonces se utiliza para su representación un montículo de máximos, donde resulta inmediata la localización de ese elemento. De forma análoga se haría si necesitamos borrar el elemento con la menor prioridad, utilizaríamos un montículo de mínimos. En un montículo de máximos resulta inmediato obtener el valor máximo de las claves del árbol. Como se ha visto anteriormente, este valor está siempre situado en la raíz del árbol, por lo tanto bastaría con acceder al primer elemento del array A[1].

Algoritmo EliminarMaximo

```

Entradas
p: arbol[1..n]
Inicio
    si (p.num = 0) entonces "Error: arbol vacío."
    A[i] <- A[n]
    n <- n - 1
    si (A[i] <> x) entonces
        si (A[i] > x) entonces Bajar(A, i)
        sino Subir(A, i)
    fin_si
Fin

```

• Gráficas: no dirigidas, dirigidas, multigráficas, pesadas (R, E, RP)

Grafos

Fundamentos y terminología básica

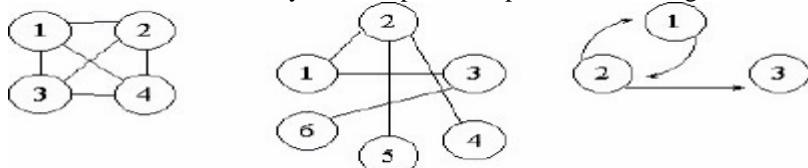
Un grafo, G, es un par, compuesto por dos conjuntos V y A. Al conjunto V se le llama conjunto de vértices o nodos del grafo. A es un conjunto de pares de vértices, estos pares se conocen habitualmente con el nombre de arcos o ejes del grafo. Se suele utilizar la notación $G = (V, A)$ para identificar un grafo. Los grafos representan un conjunto de objetos donde no hay restricción a la relación entre ellos. Son estructuras más generales y menos restrictivas. Podemos clasificar los grafos en dos grupos: dirigidos y no dirigidos. En un grafo no dirigido el par de vértices que representa un arco no está ordenado. Por lo tanto, los pares (v_1, v_2) y (v_2, v_1) representan el mismo arco. En un grafo dirigido cada arco está representado por un par ordenado de vértices, de forma que y representan dos arcos diferentes.

Ejemplos de grafos (dirigidos y no dirigidos):

$$\begin{aligned} G1 &= (V1, A1) \\ V1 &= \{1, 2, 3, 4\} & A1 &= \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\} \\ G2 &= (V2, A2) \\ V2 &= \{1, 2, 3, 4, 5, 6\} & A2 &= \{(1, 2), (1, 3), (2, 4), (2, 5), (3, 6)\} \end{aligned}$$

$$\begin{aligned} G3 &= (V3, A3) \\ V3 &= \{1, 2, 3\} & A3 &= \{<1, 2>, <2, 1>, <2, 3>\} \end{aligned}$$

Gráficamente estas tres estructuras de vértices y arcos se pueden representar de la siguiente manera:



Los grafos permiten representar conjuntos de objetos arbitrariamente relacionados. Se puede asociar el conjunto de vértices con el conjunto de objetos y el conjunto de arcos con las relaciones que se establecen entre ellos. Los grafos son modelos matemáticos de numerosas situaciones reales: un mapa de carreteras, la red de ferrocarriles, el plano de un circuito eléctrico, el esquema de la red telefónica de una compañía, etc. El número de distintos pares de vértices $(v(i), v(j))$, con $v(i) <> v(j)$, en un grafo con n vértices es $n^*(n-1)/2$. Este es el número máximo de arcos en un grafo no dirigido de n vértices. Un grafo no dirigido que tenga exactamente $n^*(n-1)/2$ arcos se dice que es un grafo completo. En el caso de un grafo dirigido de n vértices el número máximo de arcos es $n^*(n-1)$.

Algunas definiciones básicas en grafos:

- ✓ Orden de un grafo: es el número de nodos (vértices) del grafo.
- ✓ Grado de un nodo: es el número de ejes (arcos) que inciden sobre el nodo
- ✓ Grafo simétrico: es un grafo dirigido tal que si existe la relación entonces existe , con u, v pertenecientes a V.
- ✓ Grafo no simétrico: es un grafo que no cumple la propiedad anterior.
- ✓ Grafo reflexivo: es el grafo que cumple que para todo nodo u de V existe la relación (u, u) de A.
- ✓ Grafo transitivo: es aquél que cumple que si existen las relaciones (u, v) y (v, z) de A entonces existe (u, z) de A.
- ✓ Grafo completo: es el grafo que contiene todos los posibles pares de relaciones, es decir, para cualquier par de nodos u, v de V, ($u <> v$), existe (u, v) de A.
- ✓ Camino: un camino en el grafo G es una sucesión de vértices y arcos: $v(0), a(1), v(1), a(2), v(2), \dots, a(k), v(k)$; tal que los extremos del arco $a(i)$ son los vértices $v(i-1)$ y $v(i)$.
- ✓ Longitud de un camino: es el número de arcos que componen el camino.
- ✓ Camino cerrado (circuito): camino en el que coinciden los vértices extremos ($v(0) = v(k)$).
- ✓ Camino simple: camino donde sus vértices son distintos dos a dos, salvo a lo sumo los extremos.
- ✓ Camino elemental: camino donde sus arcos son distintos dos a dos.
- ✓ Camino euleriano: camino simple que contiene todos los arcos del grafo.
- ✓ Grafo euleriano: es un grafo que tiene un camino euleriano cerrado.
- ✓ Grafo conexo: es un grafo no dirigido tal que para cualquier par de nodos existe al menos un camino que los une.
- ✓ Grafo fuertemente conexo: es un grafo dirigido tal que para cualquier par de nodos existe un camino que los une.
- ✓ Punto de articulación: es un nodo que si desaparece provoca que se cree un grafo no conexo.
- ✓ Componente conexa: subgrafo conexo maximal de un grafo no dirigido (parte más grande de un grafo que sea conexa).

Representación de grafos

Existen tres maneras básicas de representar los grafos: mediante matrices, mediante listas y mediante matrices dispersas. Cada representación tiene unas ciertas ventajas e inconvenientes respecto de las demás, que comentaremos más adelante.

Representación mediante matrices: matrices de adyacencia

Un grafo es un par compuesto por dos conjuntos: un conjunto de nodos y un conjunto de relaciones entre los nodos. La representación tendrá que ser capaz de guardar esta información en memoria. La forma más fácil de guardar la información de los nodos es mediante la utilización de un vector que indexe los nodos, de manera que los arcos entre los nodos se pueden ver como relaciones entre los índices. Esta relación entre índices se puede guardar en una matriz, que llamaremos de adyacencia.

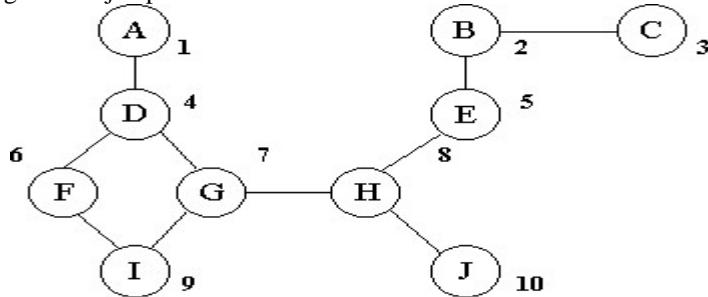
```

Const
  MAX_NODOS = ??;
Type
  Indice = 1..MAX_NODOS;
  Valor_Nodo = ??;
  Valor_Arco = ??;
  Arco = Record
    Info: Valor_Arco;      (* información asociada a cada arco *)
    Existe: Boolean;
  end;
  Nodo = Record
    Info: Valor_Nodo;    (* información asociada a cada nodo *)
    Existe: Boolean;
  end;
  Grafo = Record
    Nodos: Array[Indice] of Nodo;
    Arcos: Array[Indice, Indice] of Arco;
  end;

```

Con esta representación tendremos que reservar al menos del orden de (n^2) espacios de memoria para la información de los arcos, y las operaciones relacionadas con el grafo implicarán, habitualmente, recorrer toda la

matriz, con lo que el orden de las operaciones será, en general, cuadrático, aunque tengamos un número de relaciones entre los nodos mucho menor que (n^2). En cambio, con esta representación es muy fácil determinar, a partir de dos nodos, si están o no relacionados: sólo hay que acceder al elemento adecuado de la matriz y comprobar el valor que guarda. Ejemplo:



Supongamos el grafo representado en la figura anterior. A partir de ese grafo la información que guardariamos, con esta representación, sería:

GRAFO		1	2	3	4	5	6	7	8	9	10	11	12
Nodos	Existe	T	T	T	T	T	T	T	T	T	T	F	F
Arcos	Existe	F	F	F	V	F	F	F	F	F	F		
1													
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													

Representación mediante punteros: listas de adyacencia

En las listas de adyacencia se intenta evitar justamente el reservar espacio para aquellos arcos que no contienen ningún tipo de información. El sustituto obvio a los vectores con huecos son las listas. En las listas de adyacencia lo que haremos será guardar por cada nodo, además de la información que pueda contener el propio nodo, una lista dinámica con los nodos a los que se puede acceder desde él. La información de los nodos se puede guardar en un vector, al igual que antes, o en otra lista dinámica. Si elegimos la representación en un vector para los nodos, tendríamos la siguiente definición de grafo en Pascal:

```

Const
  MAX_NODOS = ??;
Type
  Indice = 1..MAX_NODOS;
  Valor_Nodo = ??;
  Valor_Arco = ??;
  Punt_Arco = ^Arco;
  Arco = Record
    Info: Valor_Arco;      (* información asociada a cada arco *)
    Destino: Indice;
    Sig_Arco: Punt_Arco;
  end;
  Nodo = Record
    Info: Valor_Nodo;    (* información asociada a cada nodo *)
    Existe: Boolean;
    Lista_Arcos: Punt_Arco;
  end;
  Grafo = Record
    Nodos: Array[Indice] of Nodo;
  end;

```

end;

En general se está guardando menor cantidad de elementos, sólo se reservará memoria para aquellos arcos que efectivamente existan, pero como contrapartida estamos guardando más espacio para cada uno de los arcos (estamos añadiendo el índice destino del arco y el puntero al siguiente elemento de la lista de arcos). Las tareas relacionadas con el recorrido del grafo supondrán sólo trabajar con los vértices existentes en el grafo, que puede ser mucho menor que (n^2). Pero comprobar las relaciones entre nodos no es tan directo como lo era en la matriz, sino que supone recorrer la lista de elementos adyacentes perteneciente al nodo analizado. Además, sólo estamos guardando realmente la mitad de la información que guardábamos en el caso anterior, ya que las relaciones inversas (las relaciones que llegan a un cierto nodo) en este caso no se guardan, y averiguarlas supone recorrer todas las listas de todos los nodos.

Representación mediante punteros: matrices dispersas

Para evitar uno de los problemas que teníamos con las listas de adyacencia, que era la dificultad de obtener las relaciones inversas, podemos utilizar las matrices dispersas, que contienen tanta información como las matrices de adyacencia, pero, en principio, no ocupan tanta memoria como las matrices, ya que al igual que en las listas de adyacencia, sólo representaremos aquellos enlaces que existen en el grafo.

```

Const
  MAX_NODOS = ??;
Type
  Indice = 1..MAX_NODOS;
  Valor_Nodo = ??;
  Valor_Arco = ??;
  Punt_Arco = ^Arco;
  Arco = Record
    Info: Valor_Arco;      (* información asociada a cada arco *)
    Origen: Indice;
    Destino: Indice;
    Sig_Arco_Salida: Punt_Arco;
    Sig_Arco_Entrada: Punt_Arco;
  end;
  Nodo = Record
    Info: Valor_Nodo;    (* información asociada a cada nodo *)
    Existe: Boolean;
    Lista_Arcos_Salida: Punt_Arco;
    Lista_Arcos_Entrada: Punt_Arco;
  end;
  Grafo = Record
    Nodos: Array[Indice] of Nodo;
    N_Nod: 0..MAX_NODOS;
    Existe_Nodo: Array[Indice] of Boolean;
  end;

```

Recorrido de grafos

Recorrer un grafo supone intentar alcanzar todos los nodos que estén relacionados con uno dado que tomaremos como nodo de salida. Existen básicamente dos técnicas para recorrer un grafo: el recorrido en anchura; y el recorrido en profundidad.

Recorrido en anchura o BFS (Breadth First Search)

El recorrido en anchura supone recorrer el grafo, a partir de un nodo dado, en niveles, es decir, primero los que están a una distancia de un arco del nodo de salida, después los que están a dos arcos de distancia, y así sucesivamente hasta alcanzar todos los nodos a los que se pudiese llegar desde el nodo salida.

El algoritmo general de recorrido es el siguiente:

Algoritmo Recorrido_en_Anchura (BFS)

Entradas

gr: Grafo (* grafo a recorrer *)

```

nodo_salida: Indice      (* origen del recorrido *)
Variables
queue: Cola de Indice
aux_nod1, aux_nod2: Indice
Inicio
Iniciar_Cola(queue)
Procesar(nodo_salida)
Visitado(nodo_salida) <- CIERTO
Encolar(queue, nodo_salida)
mientras NO Cola_Vacia(queue) hacer
    aux_nod1 <- Desencolar(queue)
    para (todos los nodos), aux_nod2, adyacentes a aux_nod1 hacer
        si NO Visitado[aux_nod2] entonces
            Procesar(aux_nod2)
                Visitado[aux_nod2] <- CIERTO
                Encolar(queue, aux_nod2)
        fin_si
    fin_para
fin_mientras
Fin

```

La diferencia a la hora de implementar el algoritmo general para cada una de las implementaciones de la estructura de datos grafo, residirá en la manera de averiguar los diferentes nodos adyacentes a uno dado. En el caso de las matrices de adyacencia se tendrán que comprobar si los enlaces entre los nodos existen en la matriz. En los casos de las listas de adyacencia y de las matrices dispersas sólo habrá que recorrer las listas de enlaces que parten del nodo en cuestión para averiguar qué nodos son adyacentes al estudiado.

Recorrido en profundidad o DFS (Depth First Search)

A diferencia del algoritmo anterior, el recorrido en profundidad trata de buscar los caminos que parten desde el nodo de salida hasta que ya no es posible avanzar más. Cuando ya no puede avanzarse más sobre el camino elegido, se vuelve atrás en busca de caminos alternativos, que no se estudiaron previamente. El algoritmo es similar al anterior, pero utilizando, para guardar los nodos accesibles desde uno dado, una pila en lugar de una cola.

Algoritmo Recorrido_en_Profundidad (DFS)

```

Entradas
gr: Grafo          (* grafo a recorrer *)
nodo_salida: Indice      (* origen del recorrido *)
Variables
stack: Pila de Indice
aux_nod1, aux_nod2: Indice
Inicio
Iniciar_Pila(stack)
Procesar(nodo_salida)
Visitado(nodo_salida) <- CIERTO
Apilar(stack, nodo_salida)
mientras NO Pila_Vacia(stack) hacer
    aux_nod1 <- Desapilar(stack)
    para (todos los nodos), aux_nod2, adyacentes a aux_nod1 hacer
        si NO Visitado[aux_nod2] entonces
            Procesar(aux_nod2)
                Visitado[aux_nod2] <- CIERTO
                Apilar(stack, aux_nod2)
        fin_si
    fin_para
fin_mientras
Fin

```

La utilización de la pila se puede sustituir por la utilización de la recurrencia, de manera que el algoritmo quedaría como sigue:

Algoritmo Recorrido en Profundidad (DFS)

```
Entradas
gr: Grafo          (* grafo a recorrer *)
nodo_salida: Indice    (* origen del recorrido *)
Variables
aux_nod2: Indice
Inicio
Procesar(nodo_salida)
Visitado(nodo_salida) <-- CIERTO
para (todos los nodos), aux_nod2, adyacentes a aux_nod1 hacer
    si NO Visitado[aux_nod2] entonces
        Recorrido_en_Profundidad(gr, aux_nod2)
    fin_si
fin_para
Fin
```

II. Archivos

1. Archivos de almacenamiento

• Dispositivos de almacenamiento (R)

Debido a la cantidad de información que manejamos actualmente, los dispositivos de almacenamiento se han vuelto casi tan importantes como el mismísimo computador. Aunque actualmente existen dispositivos para almacenar que superan las 650 MB de memoria, aún seguimos quejándonos por la falta de capacidad para transportar nuestros documentos y para hacer Backups de nuestra información más importante. Todo esto sucede debido al aumento de software utilitario que nos permite, por dar un pequeño ejemplo, convertir nuestros Cds en archivos de Mp3. El espacio en nuestro Disco duro ya no es suficiente para guardar tal cantidad de información; por lo que se nos es de urgencia conseguir un medio alternativo de almacenamiento para guardar nuestros Cds en Mp3 o los programas que descargamos de Internet.

La tecnología óptica

La tecnología óptica de almacenamiento por láser es bastante más reciente. Su primera aplicación comercial masiva fue el superexitoso CD de música, que data de comienzos de la década de 1.980. Los fundamentos técnicos que se utilizan son relativamente sencillos de entender: un haz láser va leyendo (o escribiendo) microscópicos agujeros en la superficie de un disco de material plástico, recubiertos a su vez por una capa transparente para su protección del polvo. Realmente, el método es muy similar al usado en los antiguos discos de vinilo, excepto porque la información está guardada en formato digital (unos y ceros como valles y cumbres en la superficie del CD) en vez de analógico y por usar un láser como lector. El sistema no ha experimentado variaciones importantes hasta la aparición del DVD, que tan sólo ha cambiado la longitud de onda del láser, reducido el tamaño de los agujeros y apretado los surcos para que quepa más información en el mismo espacio.

Disco de vídeo digital, también conocido en la actualidad como disco versátil digital (DVD), un dispositivo de almacenamiento masivo de datos cuyo aspecto es idéntico al de un disco compacto, aunque contiene hasta 25 veces más información y puede transmitirla al ordenador o computadora unas 20 veces más rápido que un CD-ROM. Su mayor capacidad de almacenamiento se debe, entre otras cosas, a que puede utilizar ambas caras del disco y, en algunos casos, hasta dos capas por cada cara, mientras que el CD sólo utiliza una cara y una capa. Las unidades lectoras de DVD permiten leer la mayoría de los CDs, ya que ambos son discos ópticos; no obstante, los lectores de CD no permiten leer DVDs. En un principio se utilizaban para reproducir películas, de ahí su denominación original de disco de vídeo digital. Hoy, los DVD-Vídeo son sólo un tipo de DVD que almacenan hasta 133 minutos de película por cada cara, con una calidad de vídeo LaserDisc y que soportan sonido digital Dolby surround; son la base de las instalaciones de cine en casa que existen desde 1996. Además de éstos, hay formatos específicos para la computadora que almacenan datos y material interactivo en forma de texto, audio o vídeo, como los DVD-R, unidades en las que se puede grabar la información una vez y leerla muchas, DVD-RW, en los que la información

se puede grabar y borrar muchas veces, y los DVD-RAM, también de lectura y escritura. En 1999 aparecieron los DVD-Audio, que emplean un formato de almacenamiento de sonido digital de segunda generación con el que se pueden recoger zonas del espectro sonoro que eran inaccesibles al CD-Audio. Todos los discos DVD tienen la misma forma física y el mismo tamaño, pero difieren en el formato de almacenamiento de los datos y, en consecuencia, en su capacidad. Así, los DVD-Vídeo de una cara y una capa almacenan 4,7 GB, y los DVD-ROM de dos caras y dos capas almacenan hasta 17 GB. Del mismo modo, no todos los DVDs se pueden reproducir en cualquier unidad lectora; por ejemplo, un DVD-ROM no se puede leer en un DVD-Vídeo, aunque sí a la inversa. Por su parte, los lectores de disco compacto, CD, y las unidades de DVD, disponen de un láser, ya que la lectura de la información se hace por procedimientos ópticos. En algunos casos, estas unidades son de sólo lectura y en otros, de lectura y escritura.

Disco duro

Disco duro, en los ordenadores o computadoras, unidad de almacenamiento permanente de gran capacidad. Está formado por varios discos apilados —dos o más—, normalmente de aluminio o vidrio, recubiertos de un material ferromagnético. Como en los disquetes, una cabeza de lectura/escritura permite grabar la información, modificando las propiedades magnéticas del material de la superficie, y leerla posteriormente (La tecnología magnética, consiste en la aplicación de campos magnéticos a ciertos materiales cuyas partículas reaccionan a esa influencia, generalmente orientándose en unas determinadas posiciones que conservan tras dejar de aplicarse el campo magnético. Esas posiciones representan los datos, bien sean una canción, bien los bits que forman una imagen o un documento importante.); esta operación se puede hacer un gran número de veces. La mayor parte de los discos duros son fijos, es decir, están alojados en el ordenador de forma permanente. Existen también discos duros removibles, como los discos Jaz de Iomega, que se utilizan generalmente para hacer backup —copias de seguridad de los discos duros— o para transferir grandes cantidades de información de un ordenador a otro. El primer disco duro se instaló en un ordenador personal en 1979; era un Seagate con una capacidad de almacenamiento de 5 MB. Hoy día, la capacidad de almacenamiento de un disco duro puede superar los 50 MB. A la vez que aumentaba la capacidad de almacenamiento, los discos duros reducían su tamaño; así se pasó de las 12 pulgadas de diámetro de los primeros, a las 3,5 pulgadas de los discos duros de los ordenadores portátiles o las 2,5 pulgadas de los discos de los *notebooks* (ordenadores de mano). Modernamente, sólo se usan en el mundo del PC dos tipos de disco duro: el IDE y el SCSI (leído "escasí"). La diferencia entre estos Discos duros radica en la manera de conectarlos a la MainBoard.

IDE

Los discos IDE son los más habituales; ofrecen un rendimiento razonablemente elevado a un precio económico y son más o menos fáciles de instalar. Sin embargo, se ven limitados a un número máximo de 4 dispositivos (y esto con las controladoras EIDE, las IDE originales sólo pueden manejar 2).

Su conexión se realiza mediante un cable plano con conectores con 40 pines colocados en dos hileras (aparte del cable de alimentación, que es común para todos los tipos de disco duro). Así pues, para identificar correctamente un disco IDE basta con observar la presencia de este conector, aunque para estar seguros al 100% deberemos buscar unos microinterruptores ("jumpers") que, en número de 2 a 4, permiten elegir el orden de los dispositivos (es decir, si se comportan como "Maestro" o como "Esclavo").

SCSI

Esta tecnología es mucho menos utilizada, pero no por ser mala, sino por ser relativamente cara. Estos discos suelen ser más rápidos a la hora de transmitir datos, a la vez que usan menos al procesador para hacerlo, lo que se traduce en un aumento de prestaciones. Es típica y casi exclusiva de ordenadores caros, servidores de red y muchos Apple Macintosh.

Los conectores SCSI son múltiples, como lo son las variantes de la norma: SCSI-1, SCSI-2, Wide SCSI, Ultra SCSI... Pueden ser planos de 50 contactos en 2 hileras, o de 68 contactos, o no planos con conector de 36 contactos, con mini-conector de 50 contactos...

Una pista para identificarlos puede ser que, en una cadena de dispositivos SCSI (hasta 7 ó 15 dispositivos que van intercalados a lo largo de un cable o cables, como las bombillas de un árbol de Navidad), cada aparato tiene un número que lo identifica, que en general se puede seleccionar. Para ello habrá una hilera de jumpers, o bien una rueda giratoria, que es lo que deberemos buscar.

MFM, ESDI

Muy similares, especialmente por el hecho de que están descatalogados. Su velocidad resulta insufrible, más parecida a la de un disquete que a la de un disco duro moderno. Se trata de cacharros pesados, de formato casi siempre 5,25 pulgadas, con capacidades de 10, 20, 40 o hasta 80 megas máximo.

Dispositivos Periféricos.

Jaz (Iomega) - 1 GB ó 2 GB

- ✓ **Pros:** capacidad muy elevada, velocidad, portabilidad
- ✓ **Contras:** inversión inicial, no tan resistente como un magneto-óptico, cartuchos relativamente caros

Las cifras de velocidad del Jaz son absolutamente alucinantes, casi indistinguibles de las de un disco duro moderno: poco más de 5 MB/s y menos de 15 ms. La razón de esto es fácil de explicar: cada cartucho Jaz es internamente, a casi todos los efectos, un disco duro al que sólo le falta el elemento lector-grabador, que se encuentra en la unidad. Por ello, atesora las ventajas de los discos duros: gran capacidad a bajo precio y velocidad, junto con sus inconvenientes: información sensible a campos magnéticos, durabilidad limitada en el tiempo, relativa fragilidad. De cualquier forma, y sin llegar a la extrema resistencia de los discos Zip, podemos calificar este soporte de *duro* y fiable, aunque la información nunca estará tan a salvo como si estuviera guardada en un soporte óptico o magneto-óptico.

Aplicaciones

Almacenamiento masivo de datos que deben guardarse y recuperarse con la mayor velocidad posible, lo cual lo hace ideal para la edición de vídeo digital (casi una hora en formato MPEG); en general, sirve para lo mismo que los discos duros, pero con la ventaja de su portabilidad y fácil almacenaje. En cuanto a defectos y críticas, aparte de que los datos no duren "para siempre", sólo tiene un inconveniente: el precio. La unidad lectora-grabadora de 1 GB vale una respetable cantidad de dinero, unos \$650.000, y los discos unos \$180.000 c/u.

Zip (Iomega) - 100 MB

- ✓ **Pros:** portabilidad, reducido formato, precio global, muy extendido
- ✓ **Contras:** capacidad reducida, incompatible con disquetes de 3,5"

Las unidades Zip se caracterizan externamente por ser de un color azul oscuro, al igual que los disquetes habituales (los hay de todos los colores). Estos discos son dispositivos magnéticos un poco mayores que los clásicos disquetes de 3,5 pulgadas, aunque mucho más robustos y fiables, con una capacidad sin compresión de 100 MB una vez formateados. Su capacidad los hace inapropiados para hacer copias de seguridad del disco duro completo, aunque perfectos para archivar todos los archivos referentes a un mismo tema o proyecto en un único disco. Su velocidad de transferencia de datos no resulta comparable a la de un disco duro actual, aunque son decenas de veces más rápidos que una disquetera tradicional (alrededor de 1 MB/s). Existen en diversos formatos, tanto internos como externos. Los internos pueden tener interfaz IDE, como la de un disco duro o CD-ROM, o bien SCSI; ambas son bastante rápidas, la SCSI un poco más, aunque su precio es también superior. Las versiones externas aparecen con interfaz SCSI (con un rendimiento idéntico a la versión interna) o bien conectable al puerto paralelo, sin tener que prescindir de la impresora conectada a éste. El modelo para puerto paralelo pone el acento en la portabilidad absoluta entre ordenadores (Sólo se necesita que tengan el puerto Lpt1) aunque su velocidad es la más reducida de las tres versiones. Muy resistente, puede ser el acompañante ideal de un portátil. Ha tenido gran aceptación, siendo el estándar en su segmento, pese a no poder prescindir de la disquetera de 3,5" con la que no son en absoluto compatibles, aunque sus ventajas puede que suplan este inconveniente. El precio de la versión interna ronda los \$262.500 (más IVA) y los Discos alrededor de \$35.000 (más IVA). Muchas de las primeras unidades Zip sufrieron el denominado "mal del click", que consistía en un defecto en la unidad

lectora-grabadora que, tras hacer unos ruiditos o "clicks", destrozaba el disco introducido; afortunadamente, este defecto está corregido en las unidades actuales. En todo caso, los discos son bastante resistentes, pero evidentemente no llegan a durar lo que un CD-ROM.

- **Parámetros de almacenamiento: número de cilindros, pistas, tiempo de acceso, posicionamiento, densidad de grabación (R, E, RP)**

Tipos de discos compactos

SOPORTE	CAPACIDAD DE ALMACENAMIENTO	DURACIÓN MÁXIMA DE AUDIO	DURACIÓN MÁXIMA DE VÍDEO	NÚMERO DE CDs A LOS QUE EQUIVALE
Disco compacto (CD)	650 Mb	1 h 18 min.	15 min.	1
DVD una cara / una capa	4,7 Gb	9 h 30 min.	2 h 15 min.	7
DVD una cara / doble capa	8,5 Gb	17 h 30 min.	4 h	13
DVD doble cara / una capa	9,4 Gb	19 h	4 h 30 min.	14
DVD doble cara / doble capa	17 Gb	35 h	8 h	26

- **Formas de almacenaje: modo texto y modo binario (R, E)**

Modo texto

En informática, también denominado modo alfanumérico o modo carácter. El modo de operación con el cual algunos equipos informáticos muestran letras y otros caracteres de texto, pero no imágenes gráficas como punteros de *mouse* (ratón) o formato de caracteres de WYSIWYG, es decir, cursivas, números superíndices y otros. Los equipos IBM y compatibles pueden operar tanto en modo texto como en modo gráfico. Los equipos Apple Macintosh, al ser máquinas que se basan en gráficos, operan en modo gráfico.

2. Accesos

- **Acceso secuencial (R, E, RP)**

Dependiendo de la manera en que se accesen los registros de un archivo, se le clasifica como SECUENCIAL o como DIRECTO. En el caso de los archivos de ACCESO SECUENCIAL, para tener acceso al registro localizado en la posición N, se deben haber accedido los N-1 registros previos, en un orden secuencial. Cuando se tienen pocos registros en un archivo, o que los registros son pequeños, la diferencia entre los tiempos de acceso de forma secuencial y directa puede no ser perceptible para el usuario; sin embargo, la diferencia viene a ser significativa cuando se manejan archivos con grandes cantidades de información. La forma de manejar los archivos de acceso secuencial es más sencilla en la mayoría de los lenguajes de programación, por lo que su estudio se antepone al de los archivos de acceso directo. El manejo secuencial de un archivo es recomendable cuando se deben procesar todos o la mayoría de los registros, como por ejemplo en los casos de una nómina de empleados o en la elaboración de reportes contables.

- **Acceso directo y cálculo de posición (R, E, RP)**
- **Acceso por índices: índice secuencial ordenado (R, E)**
- **Acceso por índices: tablas de dispersión (R, E)**
- **Acceso por índices: árboles B y B+ (R, E)**

Árboles B y árboles B+

Los árboles B y los árboles B+ son casos especiales de árboles de búsqueda. Un árbol de búsqueda es un tipo de árbol que sirve para guiar la búsqueda de un registro, dado el valor de uno de sus campos. Los índices multinivel de la sección anterior pueden considerarse como variaciones de los árboles de búsqueda. Cada bloque o nodo del índice multinivel puede tener hasta p valores del campo de indexación y p punteros. Los valores del campo de indexación de cada nodo guían al siguiente nodo (que se encuentra en otro nivel), hasta llegar al bloque del fichero de datos que contiene el registro deseado. Al seguir un puntero, se va restringiendo la búsqueda en cada nivel a un subárbol del árbol de búsqueda, y se ignoran todos los nodos que no estén en dicho subárbol. Los árboles de búsqueda difieren un poco de los índices multinivel. Un árbol de búsqueda de orden p es un árbol tal que cada nodo contiene como mucho $p - 1$ valores del campo de indexación y p punteros:

$$(P_1, K_1, P_2, K_2, \dots, P_{q-1}, K_{q-1}, P_q)$$

donde $q \leq p$, cada P_i es un puntero a un nodo hijo y cada K_i es un valor de búsqueda proveniente de algún conjunto ordenado de valores. Se supone que todos los valores de búsqueda son únicos. Un árbol de búsqueda debe cumplir, en todo momento, las siguientes restricciones:

1. Dentro de cada nodo: $K_1 < K_2 < \dots < K_{q-1}$
2. Para todos los valores X del campo de indexación del subárbol al que apunta P_i , se tiene: $K_{i-1} < X < K_i$ para $1 < i < q$, $X < K_i$ para $i = 1$ y $K_{i-1} < X$ para $i = q$.

Al buscar un valor X siempre se sigue el puntero P_i apropiado de acuerdo con las condiciones de la segunda restricción. Para insertar valores de búsqueda en el árbol y eliminarlos, sin violar las restricciones anteriores, se utilizan algoritmos que no garantizan que el árbol de búsqueda esté equilibrado (que todas las hojas estén al mismo nivel). Es importante mantener equilibrados los árboles de búsqueda porque esto garantiza que no habrá nodos en niveles muy profundos que requieran muchos accesos a bloques durante una búsqueda. Además, las eliminaciones de registros pueden hacer que queden nodos casi vacíos, con lo que hay un desperdicio de espacio importante que también provoca un aumento en el número de niveles. El árbol B es un árbol de búsqueda, con algunas restricciones adicionales, que resuelve hasta cierto punto los dos problemas anteriores. Estas restricciones adicionales garantizan que el árbol siempre estará equilibrado y que el espacio desperdiciado por la eliminación, si lo hay, nunca será excesivo. Los algoritmos para insertar y eliminar se hacen más complejos para poder mantener estas restricciones. No obstante, la mayor parte de las inserciones y eliminaciones son procesos simples, se complican sólo en circunstancias especiales: cuando se intenta insertar en un nodo que está lleno o cuando se intenta borrar en un nodo que está ocupado hasta la mitad. Un árbol B de orden p se define del siguiente modo:

1. La estructura de cada nodo interno tiene la forma:

$$(P_1, (K_1, Pr_1), P_2, (K_2, Pr_2), P_3, (K_3, Pr_3), \dots, P_{q-1}, (K_{q-1}, Pr_{q-1}), P_q)$$

donde $q \leq p$. Cada P_i es un puntero a un nodo interno del árbol y cada Pr_i es un puntero al registro del fichero de datos que tiene el valor K_i en el campo de búsqueda o de indexación.

2. Dentro de cada nodo se cumple:
3. Para todos los valores X del campo de indexación del subárbol al que apunta P_i , se cumple: $K_{i-1} < X < K_i$ para $1 < i < q$, $X < K_i$ para $i = 1$ y $K_{i-1} < X$ para $i = q$.
4. Cada nodo tiene, como mucho, p punteros a nodos del árbol.
5. Cada nodo, excepto la raíz y las hojas, tiene, al menos, $\lceil p/2 \rceil$ punteros a nodos del árbol. El nodo raíz tiene, como mínimo, dos punteros a nodos del árbol, a menos que sea el único nodo del árbol.
6. Un nodo con q punteros a nodos, $q \leq p$, tiene $q - 1$ valores del campo de indexación.
7. Todos los nodos hoja están al mismo nivel. Los nodos hoja tienen la misma estructura que los nodos internos, pero los punteros a nodos del árbol son nulos.

Como se puede observar, en los árboles B todos los valores del campo de indexación aparecen alguna vez en algún nivel del árbol, junto con un puntero al fichero de datos. En un árbol B+ los punteros a datos se almacenan sólo en los nodos hoja del árbol, por lo cual, la estructura de los nodos hoja difiere de la de los nodos internos. Los nodos

hoja tienen una entrada por cada valor del campo de indexación, junto con un puntero al registro del fichero de datos. Estos nodos están enlazados para ofrecer un acceso ordenado a los registros a través del campo de indexación. Los nodos hoja de un árbol B+ son similares al primer nivel (nivel base) de un índice. Los nodos internos del árbol B+ corresponden a los demás niveles del índice. Algunos valores del campo de indexación se repiten en los nodos internos del árbol B+ con el fin de guiar la búsqueda. En un árbol B+ de orden p la estructura de un nodo interno es la siguiente:

1. Todo nodo interno es de la forma:

$$(P_1, K_1, P_2, K_2, P_3, K_3, \dots, P_{q-1}, K_{q-1}, P_q)$$

donde $q \leq p$. Cada P_i es un puntero a un nodo interno del árbol.

$$K_1 < K_2 < \dots < K_{q-1}$$

2. Dentro de cada nodo interno se cumple:

3. Para todos los valores X del campo de indexación del subárbol al que apunta P_i , se cumple:

$$K_{i-1} < X \leq K_i \text{ para } 1 < i < q, \quad X \leq K_i \text{ para } i = 1 \text{ y } K_{i-1} < X \text{ para } i = q$$

4. Cada nodo interno tiene, como mucho, p punteros a nodos del árbol.

5. Cada nodo interno, excepto la raíz, tiene, al menos, $\lfloor p/2 \rfloor$ punteros a nodos del árbol. El nodo raíz tiene, como mínimo, dos punteros a nodos del árbol si es un nodo interno.

6. Un nodo interno con q punteros a nodos, $q \leq p$, tiene $q - 1$ valores del campo de indexación.

La estructura de los nodos hoja de un árbol B+ de orden p es la siguiente:

1. Todo nodo hoja es de la forma:

$$((K_1, Pr_1), (K_2, Pr_2), (K_3, Pr_3), \dots, (K_{q-1}, Pr_{q-1}), P_{siguiente})$$

donde $q \leq p$. Cada Pr_i es un puntero al registro de datos que tiene el valor K_i en el campo de indexación, y $P_{siguiente}$ es un puntero al siguiente nodo hoja del árbol.

2. Dentro de cada nodo hoja se cumple:

$$\lfloor p/2 \rfloor$$

3. Cada nodo hoja tiene, al menos, $\lfloor p/2 \rfloor$ valores.

4. Todos los nodos hoja están al mismo nivel.

Como las entradas en los nodos internos de los árboles B+ contienen valores del campo de indexación y punteros a nodos del árbol, pero no contienen punteros a los registros del fichero de datos, es posible ``empaquetar'' más entradas en un nodo interno de un árbol B+ que en un nodo similar de un árbol B. Por tanto, si el tamaño de bloque (nodo) es el mismo, el orden p será mayor para el árbol B+ que para el árbol B. Esto puede reducir el número de niveles del árbol B+, mejorándose así el tiempo de acceso. Como las estructuras de los nodos internos y los nodos hoja de los árboles B+ son diferentes, su orden p puede ser diferente. Se ha demostrado por análisis y simulación que después de un gran número de inserciones y eliminaciones aleatorias en un árbol B, los nodos están ocupados en un 69% cuando se estabiliza el número de valores del árbol. Esto también es verdadero en el caso de los árboles B+. Si llega a suceder esto, la división y combinación de nodos ocurrirá con muy poca frecuencia, de modo que la inserción y la eliminación se volverán muy eficientes. Cuando los árboles se definen sobre un campo no clave, los punteros a datos pasan a ser punteros a bloques de punteros a datos (se añade un nivel de indirección).

III. Fundamentos de algorítmica

1. Interpretación de algoritmos

- Resultado de algoritmos que involucren ejecución secuencial (asignación, condición e iteración) (R, E)
- Resultado de algoritmos que involucren ejecución recursiva (R, E)
- Resultado de algoritmos que involucren ejecución concurrente (R, E)

2. Aplicación y reconocimiento de algoritmos de búsqueda y ordenamiento interno

- Burbuja, de inserción y selección: rangos de n para los cuales son recomendados, estructuras que requieren del lenguaje de implementación, con qué tipos de llaves son convenientes (R, E, RP)

Ordenamiento Burbuja (Bubblesort)

1. Descripción. Este es el algoritmo más sencillo probablemente. Ideal para empezar. Consiste en ciclar repetidamente a través de la lista, comparando elementos adyacentes de dos en dos. Si un elemento es mayor que el que está en la siguiente posición se intercambian. ¿Sencillo no?

2. Pseudocódigo en C.

Tabla de variables		
Nombre	Tipo	Uso
lista	Cualquiera	Lista a ordenar
TAM	Constante entera	Tamaño de la lista
i	Entero	Contador
j	Entero	Contador
temp	El mismo que los elementos de la lista	Para realizar los intercambios

```

1. for (i=1; i<TAM; i++)
2.   for j=0 ; j<TAM - 1; j++)
3.     if (lista[j] > lista[j+1])
4.       temp = lista[j];
5.       lista[j] = lista[j+1];
6.       lista[j+1] = temp;

```

3. Un ejemplo. Vamos a ver un ejemplo. Esta es nuestra lista:

4 - 3 - 5 - 2 - 1

Tenemos 5 elementos. Es decir, TAM toma el valor 5. Comenzamos comparando el primero con el segundo elemento. 4 es mayor que 3, así que intercambiamos. Ahora tenemos:

3 - 4 - 5 - 2 - 1

Ahora comparamos el segundo con el tercero: 4 es menor que 5, así que no hacemos nada. Continuamos con el tercero y el cuarto: 5 es mayor que 2. Intercambiamos y obtenemos:

3 - 4 - 2 - 5 - 1

Comparamos el cuarto y el quinto: 5 es mayor que 1. Intercambiamos nuevamente:

3 - 4 - 2 - 1 - 5

Repetiendo este proceso vamos obteniendo los siguientes resultados:

3 - 2 - 1 - 4 - 5
2 - 1 - 3 - 4 - 5
1 - 2 - 3 - 4 - 5

4. Optimizando. Se pueden realizar algunos cambios en este algoritmo que pueden mejorar su rendimiento.

Si observas bien, te darás cuenta que en cada pasada a través de la lista un elemento va quedando en su posición final. Si no te queda claro mira el ejemplo de arriba. En la primera pasada el 5 (elemento mayor) quedó en la última posición, en la segunda el 4 (el segundo mayor elemento) quedó en la penúltima posición. Podemos evitar hacer comparaciones innecesarias si disminuimos el número de éstas en cada pasada. Tan sólo hay que cambiar el ciclo interno de esta manera:

for (j=0; j<TAM - i; j++)

Puede ser que los datos queden ordenados antes de completar el ciclo externo. Podemos modificar el algoritmo para que verifique si se han realizado intercambios. Si no se han hecho entonces terminamos con la ejecución, pues eso significa que los datos ya están ordenados.

Otra forma es ir guardando la última posición en que se hizo un intercambio, y en la siguiente pasada sólo comparar hasta antes de esa posición.

Análisis del algoritmo. Éste es el análisis para la versión no optimizada del algoritmo:

Estabilidad: Este algoritmo nunca intercambia registros con claves iguales. Por lo tanto es estable.

Requerimientos de Memoria: Este algoritmo sólo requiere de una variable adicional para realizar los intercambios.

Tiempo de Ejecución: El ciclo interno se ejecuta n veces para una lista de n elementos. El ciclo externo también se ejecuta n veces. Es decir, la complejidad es $n * n = O(n^2)$. El comportamiento del caso promedio depende del orden de entrada de los datos, pero es sólo un poco mejor que el del peor caso, y sigue siendo $O(n^2)$.

Ventajas:

- ✓ Fácil implementación.
- ✓ No requiere memoria adicional.

Desventajas:

- ✓ Muy lento.
- ✓ Realiza numerosas comparaciones.
- ✓ Realiza numerosos intercambios.

Este algoritmo es uno de los más pobres en rendimiento. Si miras la demostración te darás cuenta de ello. No es recomendable usarlo. Tan sólo está aquí para que lo conozcas, y porque su sencillez lo hace bueno para empezar. Ya veremos otros mucho mejores. Ahora te recomiendo que hagas un programa y lo pruebes. Si tienes dudas mira el programa de ejemplo.

Ordenamiento por Inserción

1. Descripción. Este algoritmo también es bastante sencillo. ¿Has jugado cartas?. ¿Cómo las vas ordenando cuando las recibes? Yo lo hago de esta manera: tomo la primera y la coloco en mi mano. Luego tomo la segunda y la comparo con la que tengo: si es mayor, la pongo a la derecha, y si es menor a la izquierda (también me fijo en el color, pero omitiré esa parte para concentrarme en la idea principal). Después tomo la tercera y la comparo con las que tengo en la mano, desplazándola hasta que quede en su posición final. Continúo haciendo esto, insertando cada carta en la posición que le corresponde, hasta que las tengo todas en orden. ¿Lo haces así tu también? Bueno, pues si es así entonces comprenderás fácilmente este algoritmo, porque es el mismo concepto.

Para simular esto en un programa necesitamos tener en cuenta algo: no podemos desplazar los elementos así como así o se perderá un elemento. Lo que hacemos es guardar una copia del elemento actual (que sería como la carta que tomamos) y desplazar todos los elementos mayores hacia la derecha. Luego copiamos el elemento guardado en la posición del último elemento que se desplazó.

2. Pseudocódigo en C.

Tabla de variables		
Nombre	Tipo	Uso
lista	Cualquiera	Lista a ordenar
TAM	Constante Entera	Tamaño de la lista
i	Entero	Contador
j	Entero	Contador

Tabla de variables		
Nombre	Tipo	Uso
temp	El mismo que los elementos de la lista	Para realizar los intercambios

1. for (i=1; i<TAM; i++)
2. temp = lista[i];
3. j = i - 1;
4. while ((lista[j] > temp) && (j >= 0))
5. lista[j+1] = lista[j];
6. j--;
7. lista[j+1] = temp;

Nota: Observa que en cada iteración del ciclo externo los elementos 0 a i forman una lista ordenada.

3. Un ejemplo. ¿Te acuerdas de nuestra famosa lista?

4 - 3 - 5 - 2 - 1

temp toma el valor del segundo elemento, 3. La primera carta es el 4. Ahora comparamos: 3 es menor que 4. Luego desplazamos el 4 una posición a la derecha y después copiamos el 3 en su lugar.

4 - 4 - 5 - 2 - 1
3 - 4 - 5 - 2 - 1

El siguiente elemento es 5. Comparamos con 4. Es mayor que 4, así que no ocurren intercambios. Continuamos con el 2. Es menor que cinco: desplazamos el 5 una posición a la derecha:

3 - 4 - 5 - 5 - 1

Comparamos con 4: es menor, así que desplazamos el 4 una posición a la derecha:

3 - 4 - 4 - 5 - 1

Comparamos con 3. Desplazamos el 3 una posición a la derecha:

3 - 3 - 4 - 5 - 1

Finalmente copiamos el 2 en su posición final:

2 - 3 - 4 - 5 - 1

El último elemento a ordenar es el 1. Cinco es menor que 1, así que lo desplazamos una posición a la derecha:

2 - 3 - 4 - 5 - 5

Continuando con el procedimiento la lista va quedando así:

2 - 3 - 4 - 4 - 5
2 - 3 - 3 - 4 - 5
2 - 2 - 3 - 4 - 5
1 - 2 - 3 - 4 - 5

4. Análisis del algoritmo.

Estabilidad: Este algoritmo nunca intercambia registros con claves iguales. Por lo tanto es estable.

Requerimientos de Memoria: Una variable adicional para realizar los intercambios.

Tiempo de Ejecución: Para una lista de n elementos el ciclo externo se ejecuta n-1 veces. El ciclo interno se ejecuta como máximo una vez en la primera iteración, 2 veces en la segunda, 3 veces en la tercera, etc. Esto produce una complejidad O(n²).

Ventajas:

- ✓ Fácil implementación.
- ✓ Requerimientos mínimos de memoria.

Desventajas:

- ✓ Lento.
- ✓ Realiza numerosas comparaciones.

Este también es un algoritmo lento, pero puede ser de utilidad para listas que están ordenadas o semiordenadas, porque en ese caso realiza muy pocos desplazamientos.

Ordenamiento por Selección.

1. Descripción. Este algoritmo también es sencillo. Consiste en lo siguiente:

- Buscas el elemento más pequeño de la lista.
- Lo intercambias con el elemento ubicado en la primera posición de la lista.
- Buscas el segundo elemento más pequeño de la lista.
- Lo intercambias con el elemento que ocupa la segunda posición en la lista.
- Repite este proceso hasta que hayas ordenado toda la lista.

2. Pseudocódigo en C.

Tabla de variables	Nombre	Tipo	Uso
lista	Cualquiera	Lista	a ordenar
TAM	Constante entera	Tamaño	de la lista
i	Entero	Contador	
pos_men	Entero	Posición del menor elemento de la lista	
temp	El mismo que los elementos de la lista	Para realizar los intercambios	

```
1. for (i=0; i<TAM - 1; i++)  
2.   pos_men = Menor(lista, TAM, i);  
3.   temp = lista[i];  
4.   lista[i] = lista [pos_men];  
5.   lista [pos_men] = temp;
```

Nota: Menor(lista, TAM, i) es una función que busca el menor elemento entre las posiciones i y TAM-1. La búsqueda es lineal (elemento por elemento).

3. Un ejemplo. Vamos a ordenar la siguiente lista (la misma del ejemplo anterior)

4 - 3 - 5 - 2 - 1

Comenzamos buscando el elemento menor entre la primera y última posición. Es el 1. Lo intercambiamos con el 4 y la lista queda así:

1 - 3 - 5 - 2 - 4

Ahora buscamos el menor elemento entre la segunda y la última posición. Es el 2. Lo intercambiamos con el elemento en la segunda posición, es decir el 3. La lista queda así:

1 - 2 - 5 - 3 - 4

Buscamos el menor elemento entre la tercera posición (sí, adivinaste :-D) y la última. Es el 3, que intercambiamos con el 5:

1 - 2 - 3 - 5 - 4

El menor elemento entre la cuarta y quinta posición es el 4, que intercambiamos con el 5:

1 - 2 - 3 - 4 - 5

4. Análisis del algoritmo.

Estabilidad: si tengo dos registros con claves iguales, el que ocupe la posición más baja será el primero que sea identificado como menor. Es decir que será el primero en ser desplazado. El segundo registro será el menor en el siguiente ciclo y quedará en la posición adyacente. Por lo tanto se mantendrá el orden relativo. Lo que podría hacerlo inestable sería que el ciclo que busca el elemento menor revisara la lista desde la última posición hacia atrás.

Requerimientos de Memoria: Al igual que el ordenamiento burbuja, este algoritmo sólo necesita una variable adicional para realizar los intercambios. Tiempo de Ejecución: El ciclo externo se ejecuta n veces para una lista de n elementos. Cada búsqueda requiere comparar todos los elementos no clasificados. Luego la complejidad es $O(n^2)$. Este algoritmo presenta un comportamiento constante independiente del orden de los datos. Luego la complejidad promedio es también $O(n^2)$.

Ventajas:

- ✓ Fácil implementación.
- ✓ No requiere memoria adicional.
- ✓ Realiza pocos intercambios.
- ✓ Rendimiento constante: poca diferencia entre el peor y el mejor caso.

Desventajas:

- ✓ Lento.
- ✓ Realiza numerosas comparaciones.

Este es un algoritmo lento. No obstante, ya que sólo realiza un intercambio en cada ejecución del ciclo externo, puede ser una buena opción para listas con registros grandes y claves pequeñas. Si miras el programa de demostración notarás que es el más rápido en la parte gráfica (por lo menos en un PC lento y con una tarjeta gráfica mala como el mío x-)). La razón es que es mucho más lento dibujar las barras que comparar sus largos (el desplazamiento es más costoso que la comparación), por lo que en este caso especial puede vencer a algoritmos como Quicksort. Bien, ya terminamos con éste. Otra vez te recomiendo que hagas un programa y trates de implementar este algoritmo, de preferencia sin mirar el código ni el pseudocódigo otra vez.

- Rápido (*quicksort*), mezcla (*mergesort*), ordenamiento de Shell (*shellsort*), montículo (*heapsort*): rangos de n para los cuales son recomendados, estructuras que requieren del lenguaje de implementación, con qué tipos de llaves son convenientes (R, E, RP)

Ordenamiento Rápido (Quicksort)

1. Descripción. Esta es probablemente la técnica más rápida conocida. Fue desarrollada por C.A.R. Hoare en 1960. El algoritmo original es recursivo, pero se utilizan versiones iterativas para mejorar su rendimiento (los algoritmos recursivos son en general más lentos que los iterativos, y consumen más recursos). El algoritmo fundamental es el siguiente: Eliges un elemento de la lista. Puede ser cualquiera (en Optimizando veremos una forma más efectiva).

Lo llamaremos elemento de división. Buscas la posición que le corresponde en la lista ordenada (explicado más abajo).

Acomodas los elementos de la lista a cada lado del elemento de división, de manera que a un lado queden todos los menores que él y al otro los mayores (explicado más abajo también). En este momento el elemento de división separa la lista en dos sublistas (de ahí su nombre). Realizas esto de forma recursiva para cada sublista mientras éstas tengan un largo mayor que 1. Una vez terminado este proceso todos los elementos estarán ordenados. Una idea preliminar para ubicar el elemento de división en su posición final sería contar la cantidad de elementos menores y colocarlo un lugar más arriba. Pero luego habría que mover todos estos elementos a la izquierda del elemento, para que se cumpla la condición y pueda aplicarse la recursividad. Reflexionando un poco más se obtiene un procedimiento mucho más efectivo. Se utilizan dos índices: i , al que llamaremos contador por la izquierda, y j , al que llamaremos contador por la derecha. El algoritmo es éste: Recorres la lista simultáneamente con i y j : por la izquierda con i (desde el primer elemento), y por la derecha con j (desde el último elemento). Cuando $\text{lista}[i]$ sea mayor que el elemento de división y $\text{lista}[j]$ sea menor los intercambias. Repites esto hasta que se crucen los índices. El punto en que se cruzan los índices es la posición adecuada para colocar el elemento de división, porque sabemos que a un lado los elementos son todos menores y al otro son todos mayores (o habrían sido intercambiados). Al finalizar este

procedimiento el elemento de división queda en una posición en que todos los elementos a su izquierda son menores que él, y los que están a su derecha son mayores.

2. Pseudocódigo en C.

Tabla de variables		
Nombre	Tipo	Uso
lista	Cualquiera	Lista a ordenar
inf	Entero	Elemento inferior de la lista
sup	Entero	Elemento superior de la lista
elem_div	El mismo que los elementos de la lista	El elemento divisor
temp	El mismo que los elementos de la lista	Para realizar los intercambios
i	Entero	Contador por la izquierda
j	Entero	Contador por la derecha
cont	Entero	El ciclo continua mientras cont tenga el valor 1

Nombre Procedimiento: OrdRap

Parámetros:

```

lista a ordenar (lista)
índice inferior (inf)
índice superior (sup)
// Inicialización de variables
1. elem_div = lista[sup];
2. i = inf - 1;
3. j = sup;
4. cont = 1;
// Verificamos que no se crucen los límites
5. if (inf >= sup)
6.   retornar;
// Clasificamos la sublista
7. while (cont)
8.   while (lista[++i] < elem_div);
9.   while (lista[--j] > elem_div);
10.  if (i < j)
11.    temp = lista[i];
12.    lista[i] = lista[j];
13.    lista[j] = temp;
14.  else
15.    cont = 0;
// Copiamos el elemento de división en su posición final
16. temp = lista[i];
17. lista[i] = lista[sup];
18. lista[sup] = temp;
// Aplicamos el procedimiento recursivamente a cada sublista
19. OrdRap (lista, inf, i - 1);
20. OrdRap (lista, i + 1, sup);

```

Nota:

La primera llamada debería ser con la lista, cero (0) y el tamaño de la lista menos 1 como parámetros.

3. Un ejemplo

Esta vez voy a cambiar de lista ;-D

5 - 3 - 7 - 6 - 2 - 1 - 4

Comenzamos con la lista completa. El elemento divisor será el 4:

5 - 3 - 7 - 6 - 2 - 1 - 4

Comparamos con el 5 por la izquierda y el 1 por la derecha.

5 - 3 - 7 - 6 - 2 - 1 - 4

5 es mayor que cuatro y 1 es menor. Intercambiamos:

1 - 3 - 7 - 6 - 2 - 5 - 4

Avanzamos por la izquierda y la derecha:

1 - 3 - 7 - 6 - 2 - 5 - 4

3 es menor que 4: avanzamos por la izquierda. 2 es menor que 4: nos mantenemos ahí.

1 - 3 - 7 - 6 - 2 - 5 - 4

7 es mayor que 4 y 2 es menor: intercambiamos.

1 - 3 - 2 - 6 - 7 - 5 - 4

Avanzamos por ambos lados:

1 - 3 - 2 - 6 - 7 - 5 - 4

En este momento termina el ciclo principal, porque los índices se cruzaron. Ahora intercambiamos lista[i] con lista[sup] (pasos 16-18):

1 - 3 - 2 - 4 - 7 - 5 - 6

Aplicamos recursivamente a la sublistas de la izquierda (índices 0 - 2). Tenemos lo siguiente:

1 - 3 - 2

1 es menor que 2: avanzamos por la izquierda. 3 es mayor: avanzamos por la derecha. Como se intercambiaron los índices termina el ciclo. Se intercambia lista[i] con lista[sup]:

1 - 2 - 3

Al llamar recursivamente para cada nueva sublistas (lista[0]-lista[0] y lista[2]-lista[2]) se retorna sin hacer cambios (condición 5.). Para resumir te muestro cómo va quedando la lista:

Segunda sublistas: lista[4]-lista[6]

7 - 5 - 6

5 - 7 - 6

5 - 6 - 7

Para cada nueva sublistas se retorna sin hacer cambios (se cruzan los índices). Finalmente, al retornar de la primera llamada se tiene el arreglo ordenado:

1 - 2 - 3 - 4 - 5 - 6 - 7

4. Optimizando. Sólo voy a mencionar algunas optimizaciones que pueden mejorar bastante el rendimiento de quicksort: Hacer una versión iterativa: Para ello se utiliza una pila en que se van guardando los límites superior e inferior de cada sublistas. No clasificar todas las sublistas: Cuando el largo de las sublistas va disminuyendo, el proceso se va encareciendo. Para solucionarlo sólo se clasifican las listas que tengan un largo menor que n. Al terminar la clasificación se llama a otro algoritmo de ordenamiento que termine la labor. El indicado es uno que se comporte bien con listas casi ordenadas, como el ordenamiento por inserción por ejemplo. La elección de n depende de varios factores, pero un valor entre 10 y 25 es adecuado. Elección del elemento de división: Se elige desde un conjunto de tres elementos: lista[inferior], lista[mitad] y lista[superior]. El elemento elegido es el que tenga el valor medio según el criterio de comparación. Esto evita el comportamiento degenerado cuando la lista está prácticamente ordenada.

5. Análisis del algoritmo.

Estabilidad: No es estable.

Requerimientos de Memoria: No requiere memoria adicional en su forma recursiva. En su forma iterativa la necesita para la pila.

Tiempo de Ejecución: Caso promedio. La complejidad para dividir una lista de n es $O(n)$. Cada sublistas genera en promedio dos sublistas más de largo $n/2$. Por lo tanto la complejidad se define en forma recurrente como:

$$\begin{aligned}f(1) &= 1 \\f(n) &= n + 2 f(n/2)\end{aligned}$$

La forma cerrada de esta expresión es:

$$f(n) = n \log_2 n$$

Es decir, la complejidad es $O(n \log_2 n)$. El peor caso ocurre cuando la lista ya está ordenada, porque cada llamada genera sólo una sublistas (todos los elementos son menores que el elemento de división). En este caso el rendimiento se degrada a $O(n^2)$. Con las optimizaciones mencionadas arriba puede evitarse este comportamiento.

Ventajas:

- ✓ Muy rápido
- ✓ No requiere memoria adicional.

Desventajas:

- ✓ Implementación un poco más complicada.
- ✓ Recursividad (utiliza muchos recursos).
- ✓ Mucha diferencia entre el peor y el mejor caso.

La mayoría de los problemas de rendimiento se pueden solucionar con las optimizaciones mencionadas arriba (al costo de complicar mucho más la implementación). Este es un algoritmo que puedes utilizar en la vida real. Es muy eficiente. En general será la mejor opción.

• Búsqueda: secuencial y binaria (R, E, RP)

Existen diferentes tipos de búsqueda, de las cuales mencionaremos tres de las más importantes:

- 1.-) Búsqueda secuencial.
- 2.-) Búsqueda binaria.
- 3.-) Búsqueda por transformación de claves (Hash).

Secuencial

La más conocida es la búsqueda secuencial y la más utilizada, sin embargo dependerá del método para saber cuál será la más eficiente y la más rápida para poder buscar un elemento deseado. La búsqueda de tipo secuencial se realiza de una forma ordenada, la cual se va ir buscando posición por posición el número que se desea buscar.

Ejemplo:

pos	0	1	2	3	4
	50	15	56	14	35

Por lo tanto se va ir buscando posición por posición, recorriendo primero la pos. 0, después la pos. 1, así sucesivamente. Si se llega a la pos. 4 (en este ejemplo) y no se encontró el número buscado, se deduce que el número no se encuentra. De lo contrario si el número se encuentra en la posición por ejemplo la pos. 2, entonces se detiene hasta ahí la búsqueda (sin continuar hacia las posiciones 3 y 4) y se deduce que el número se ha encontrado.

Binaria

La búsqueda de tipo binaria se aplica a una serie que ya se encuentra previamente ordenada. Este tipo de búsqueda consiste en posicionar a la mitad de la serie y comparar el número que se desea buscar con el que se tiene al obtener de la mitad de la serie. Aquí se van a tener tres posibles resultados:

- a) Si el número a buscar es igual al tomado, entonces ya se encontró el número deseado.
- b) Si el número a buscar es menor al tomado en la mitad de la serie, se descarta la mitad derecha y se repite de nuevo el procedimiento sólo con la mitad izquierda.
- c) Si el número a buscar es mayor al tomado en la mitad de la serie, se descarta la mitad izquierda y se repite de nuevo el procedimiento sólo con la mitad derecha.

Ejemplo: numero a buscar: 11

pos	0	1	2	3	4	5	6	7	8
	2	7	9	11	20	27	35	49	55

Se toma la mitad de la serie, aquí sería la pos. 4 y se compara el número a buscar con el 20, por lo tanto como no son iguales, checamos si el numero es mayor o menor al de la pos. 4, como es menor se descarta la mitad derecha.

pos	0	1	2	3
	2	7	9	11

Se repite el nuevo procedimiento y posiciona a la mitad de la serie (pos 1), y ahora se va a descartar la mitad izquierda por ser mayor al numero en la pos. 1.

pos	2	3
	9	11

Ahora se toma en la posición 2 y como es mayor a 9 se descarta el lado izquierdo y sólo queda el lado derecho. Cuando sólo queda un elemento de la serie y no es igual al número buscado se deduce que no existe, de lo contrario (como es en éste ejemplo) el elemento sí existe

Hash

La búsqueda por transformación de claves es también conocida como técnica Hash. Se tiene que generar una transformación de clave con el fin de obtener la dirección del número donde va a ser colocado. Por ejemplo:

Para un número 31, su transformación de clave por medio de sumas (véase en técnicas de transformación) nos genera una dirección 4, por lo tanto se va a colocar el número 31 en la pos. 4.

Ejemplo:

pos	0	1	2	3	4	5	6
					31		

Ahora bien, cuando se tienen dos número que generan la misma dirección se tiene una colisión. Por lo tanto se necesita solucionar la colisión, y tenemos dos tipos de direccionamiento:

1) Direccionamiento Abierto.

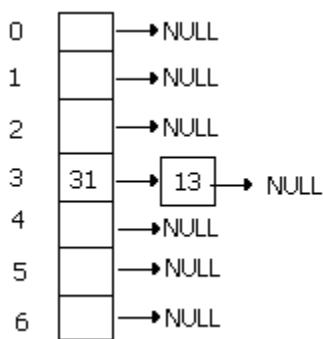
Este tipo de solución de colisiones consiste en colocar en la siguiente posición vacía el número que genero la colisión.

pos	0	1	2	3	4	5	6
					31	41	13

Aquí se tienen que hacer validaciones, como tener un cierto número de posiciones, si al realizar el direccionamiento abierto no se tienen posiciones, números repetidos, etc.

2) Direccionamiento cerrado.

Este tipo de solución de colisiones no se tiene ningún problema con que se repita la misma dirección, utilizando el uso de listas (ver dinámica). Ejemplo:



Las búsquedas de ambos casos requieren primeramente la generación de claves y después realizar la búsqueda. En la abierta se busca desde la dirección hasta una posición vacía y en el cerrado hasta que se encuentra NUL

3. Aplicación y reconocimiento de algoritmos de ordenamiento externo

• Polifase (R)

El método de mezcla equilibrada utiliza 2M archivos; esto supone utilizar muchos archivos temporales y es un inconveniente. La mezcla polifásica obtiene mejor partido de cada archivo y se utiliza un número constante de archivos de entrada. Supongamos, por ejemplo, que se dispone de tres archivos, F1, F2 y F3. F1 contiene N1 secuencias, F2 contiene N2 secuencias ($N_2 < N_1$). Se realiza la fusión de N2 secuencias de F1 con N2 secuencias de F2, produciendo sobre F3, N2 secuencias.

• Intercalación de archivos (R, E, RP)

Suponiendo que un archivo desordenado de n registros en memoria externa (disco o cinta) y que se dispone de cuatro áreas contiguas disponibles en la misma. Estas áreas pueden estar en un disco, sobre superficies diferentes de un paquete de discos o en cintas separadas. Dos de ellas son para archivos de entrada; las otras dos son para archivos de salida. Las secuencias se escriben alternativamente sobre archivos de entrada hasta que no quede ningún registro en el archivo original. El segundo paso comienza leyendo las primeras secuencias de los dos archivos de entrada, mezclándolos y escribiendo el más largo sobre uno de los archivos de salida. Las segundas secuencias de los archivos de entrada se mezclan sobre el segundo archivo de salida, etc. Cuando se utilizan cuatro archivos, éste método se conoce como mezcla equilibrada de dos caminos. En general, una mezcla equilibrada de n-caminos utiliza 2^n archivos externos, n cada una para entrada y salida y mezclas de n-caminos. La mezcla equilibrada no requiere de redistribución de registros de salida a n archivos de entrada en cada paso; sin embargo requiere aproximadamente dos veces el número de archivos de salida.

• Cascada (R)

Se disponen las secuencias sobre N archivos F1, F2, ..., FN de tal modo que el número de secuencias sobre F1 sea superior al de secuencias sobre F2, etcétera.

$$\text{numero}(F1) > \text{numero}(F2) > \text{numero}(F3) > \dots > \text{numero}(FN)$$

El método prevé un archivo vacío suplementario de numero NI+1.

• Mergesort (R, E, RP)

El uso de los métodos de ordenación externa es parte importante en el rendimiento global de una computadora y de un sistema de información. Los algoritmos de ordenación externa consideran que los datos se encuentran fuera de la memoria principal. La ordenación interna supone que todos los registros caben completamente en la memoria principal, mientras que en la ordenación externa la mayoría de los datos residirán en un dispositivo de almacenamiento como un disco o una cinta. Los algoritmos de ordenación externa se basan esencialmente en el método conocido como ordenación por mezcla (sort-merge). Estos algoritmos ordenan un archivo con la técnica de particiones (divide y vencerás).

“Mezclas equilibradas”

Suponiendo que un archivo desordenado de n registros en memoria externa (disco o cinta) y que se dispone de cuatro áreas contiguas disponibles en la misma. Estas áreas pueden estar en un disco, sobre superficies diferentes de un paquete de discos o en cintas separadas. Dos de ellas son para archivos de entrada; las otras dos son para archivos de salida. Las secuencias se escriben alternativamente sobre archivos de entrada hasta que no quede ningún registro en el archivo original. El segundo paso comienza leyendo las primeras secuencias de los dos archivos de entrada, mezclándolos y escribiendo el más largo sobre uno de los archivos de salida. Las segundas secuencias de los archivos de entrada se mezclan sobre el segundo archivo de salida, etc. Cuando se utilizan cuatro archivos, éste método se conoce como mezcla equilibrada de dos caminos. En general, una mezcla equilibrada de n-caminos utiliza 2^n archivos externos, n cada una para entrada y salida y mezclas de n-caminos. La mezcla equilibrada no requiere de redistribución de registros de salida a n archivos de entrada en cada paso; sin embargo requiere aproximadamente dos veces el número de archivos de salida.

4. Análisis de algoritmos

- "Ordenar" las distintas complejidades en algoritmos y las relaciones que existen entre distintas clases de complejidad (R)

Diseño del algoritmo

Que describe la secuencia ordenada de pasos – sin ambigüedades – que conduzcan a la solución de un problema dado. Expresar el algoritmo como un programa en un lenguaje de programación adecuado.

Ejecución y validación del programa por la computadora.

Para llegar a la realización de un programa es necesario el diseño previo de un algoritmo, de modo que sin algoritmo no puede existir un programa. Los algoritmos son independientes tanto del lenguaje de programación en que se expresan como de la computadora que los ejecuta. Sus características fundamentales son:

- a) Debe ser preciso e indicar el orden de realización de cada paso.
- b) Debe estar definido. Si se sigue un algoritmo dos veces, se debe obtener el mismo resultado cada vez.
- c) Debe ser finito. Si se sigue un algoritmo, se debe terminar en algún momento; o sea debe tener un número finito de pasos.
- d) Ha de cumplir tres partes: Entrada – Proceso – Salida.

- Dadas las funciones de complejidad, calcular el tiempo y/o espacio requerido para problemas concretos que utilicen estructuras de datos con representaciones distintas (R, E, RP)
- Problemas tratables e intratables: P, NP y NP-completos (R)

PROGRAMACIÓN E INGENIERÍA DE SOFTWARE

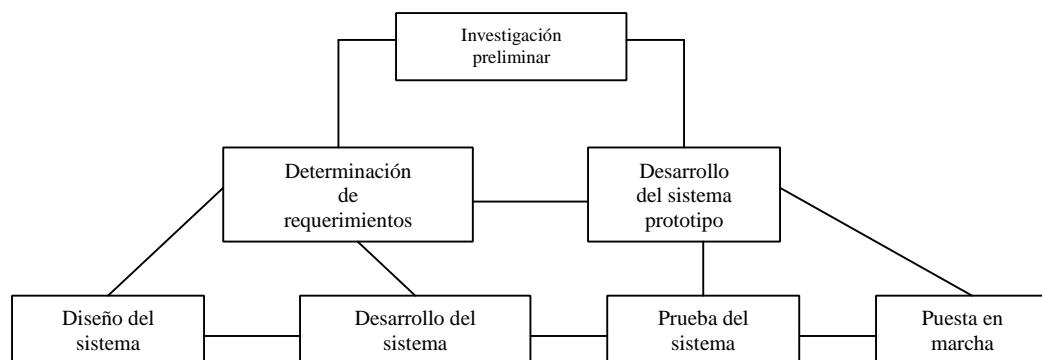
C. Sistemas e industria del software

I. Ciclo de vida de sistemas

1. Las fases del desarrollo de software

El desarrollo de sistemas es un proceso que consiste en dos etapas principales de análisis y diseño de sistemas; comienza cuando la gerencia, o en algunas ocasiones el personal de desarrollo de sistemas, se da cuenta de que cierto sistema del negocio necesita mejorarse. El ciclo de vida del desarrollo de sistemas es el conjunto de actividades de los analistas, diseñadores y usuarios, que necesitan llevarse a cabo para desarrollar y poner en marcha un sistema de información. Se debe tener presente que en la mayoría de las situaciones del negocio, las actividades están íntimamente relacionadas y son inseparables. El ciclo de vida del desarrollo de sistemas consiste en las siguientes actividades:

1. Investigación preliminar
2. Determinación de requerimientos
3. Desarrollo de sistema prototipo
4. Diseño de sistema
5. Desarrollo de software
6. Prueba de los sistemas
7. Puesta en marcha



- **Análisis preliminar (enunciado general del problema, propósito del sistema, requerimientos, principales entradas y salidas, diagrama de bloques del sistema, estimación de costos y beneficios y plan preliminar de desarrollo) (R, E)**

Investigaciones preliminares:

¿Cuántas veces se está en situaciones en donde se pregunta si no existe una mejor manera de hacer algo? Por ejemplo, abrir una tienda departamental adicional que creará una necesidad para nuevos procedimientos de facturación, cuando un alto porcentaje de clientes utiliza la cuenta de crédito de esta compañía y compra en todas las tiendas. Se puede iniciar una petición por muchas razones, pero la clave es que alguien, ya sea gerente, un empleado o un especialista de sistemas, inicie un requerimiento para recibir ayuda de un sistema de información. Cuando ese requerimiento se realiza, la primera actividad de sistemas, es decir, la investigación preliminar, se inicia. Esta actividad tiene tres partes: clarificación de requerimiento, estudio de la factibilidad y aprobación del requerimiento. El resultado será aprobar el requerimiento para la atención posterior o rechazarlo como no factible para un desarrollo futuro.

Estudio de Factibilidad:

Un resultado importante de la investigación preliminar es la determinación de que el sistema requerido es factible. Existen tres aspectos en el estudio de factibilidad de la investigación preliminar:

1. *Factibilidad técnica.* ¿Puede realizarse el trabajo para el proyecto con el equipo actual, tecnología de software y el personal disponible? Si se requiere nueva tecnología, ¿qué probabilidades hay de que pueda desarrollarse?
2. *Factibilidad económica.* ¿Existen suficientes beneficios en la creación del sistema para hacer que los costos sean aceptables? O, en forma inversa, ¿son tan altos los costos como para que el proyecto no deba llevarse a cabo?
3. *Factibilidad operativa.* ¿Se utilizará el sistema si se desarrolla y pone en marcha? Habrá resistencia de los usuarios, que los posibles beneficios reducirán del sistema.

El estudio de factibilidad se lleva a cabo con un pequeño grupo de gente, familiarizada con las técnicas de los sistemas de información, que entienden la parte de la empresa que será afectada por el proyecto y tienen los conocimientos suficientes del proceso de análisis y diseño de sistemas.

- **Análisis detallado (técnicas de recolección de información, determinación de los requerimientos detallados, formalización de los requerimientos, especificación de alcances y limitaciones del sistema) (R, E)**

Clarificación del Requerimiento:

En las empresas muchos requerimientos de los empleados y usuarios no están establecidos claramente; por lo tanto, antes de que pueda considerarse la investigación del sistema, el proyecto requerido debe examinarse para determinar precisamente lo que desea la empresa. Una simple llamada telefónica puede ser suficiente si la persona que requiere el servicio tiene una idea clara, pero no sabe cómo establecerla. Por otro lado, la persona que hace el requerimiento puede estar simplemente pidiendo ayuda sin saber qué es lo que está mal o por qué existe un problema. La clarificación del problema es este caso, antes de poder llagar a otro paso, el requerimiento de proyecto debe estar claramente establecido.

Aprobación del Requerimiento:

No todos los proyectos requeridos son deseables o factibles. Sin embargo, aquellos que son tanto factibles como deseables deben anotarse para tomarlos en cuenta. En algunos casos, el desarrollo puede comenzar inmediatamente, pero en la mayor parte, los miembros del departamento de sistemas están ocupados en otros proyectos que se encuentran en marcha. Cuando esto sucede, la gerencia decide que proyectos son más importantes y entonces los programa. Despues de que se aprueba la requisición de un proyecto, se estima su costo, la prioridad, el tiempo de terminación y los requerimientos del personal que se utilizan. Posteriormente, cuando se terminan algunos proyectos anteriores, puede iniciarse el desarrollo de la aplicación propuesta. En este momento, comienza la recabación de datos y la determinación de los requerimientos.

Determinación de Requerimientos:

El punto clave de análisis de sistemas se consigue al adquirir un conocimiento detallado de todas las facetas importantes dentro del área de negocios que se investiga. (Por esta razón, a menudo esta actividad se conoce como *investigación detallada*.) Los analistas, al trabajar con los empleados y gerentes, deben estudiar el proceso que actualmente se efectúa para contestar estas preguntas clave:

1. ¿Qué se está haciendo?
2. ¿Cómo se está haciendo?
3. ¿Qué tan frecuentemente ocurre?
4. ¿Qué tan grande es la cantidad de transacciones o decisiones?
5. ¿Qué tan bien se lleva acabo la tarea?
6. ¿Existe algún problema?
7. Si el problema existe, qué tan serio es?
8. ¿Si el problema existe, cuál es la causa principal?

Para contestar estas preguntas, los analistas de sistemas hablarán con diferentes personas para recabar los detalles en relación con el proceso, así como sus opiniones sobre las causas por las cuales suceden las cosas de esa manera y algunas ideas en relación a modificarlas. Se utilizan cuestionarios para recopilar esta información, aplicándolos a grandes que no pueden entrevistarse en forma individual. Las investigaciones detalladas también requieren el estudio de manuales y reportes, la observación real de las actividades de trabajo y algunas veces la recabación de formas y documentos para entender completamente el proceso. Conforme se recopilan los elementos, los analistas

estudian los requerimientos de datos para identificar las características que tendrá el nuevo sistema, incluyendo la información que el sistema debe producir y las características operativas, como son controles de procesamiento, tiempos de respuesta y métodos de entrada y salida.

- **Diseño preliminar (definición de restricciones, diseño de la arquitectura del sistema, definición de interfaces entre módulos) (R, E)**

Desarrollo del Sistema Prototípico.

La **preparación de prototipos** es el proceso de crear, desarrollar y refinar un modelo funcional del sistema final. Se puede crear un **modelo prototípico preliminar** durante la etapa de definición del problema. Un miembro del equipo de reconocimiento -suponga que se trata de un especialista en el procesamiento de datos- puede construir un modelo de este tipo que muestre la composición de las pantallas y los formatos de los informes. Durante una sesión de requerimientos, otros miembros del equipo y usuarios del futuro sistema examinan esta muestra en la forma con el constructor del modelo entiende en principio el problema y los resultados que debe producir el sistema. En este momento puede iniciarse un proceso de refinación si los usuarios señalan omisiones y equivocas. Durante este proceso de refinación, cuyo objetivo es definir la necesidad que existe, uno o más miembros del equipo pueden utilizar una computadora personal y un **paquete de programas de prototipos** a fin de crear una serie de pantallas en la computadora personal. Estas pantallas no son las salidas que producen los programas ya terminados, pero pueden parecerse mucho a esos resultados. Es posible exhibir en el monitor de la computadora, como una secuencia de diapositivas, menús de captura de datos, la interfaz con el usuario debe servir para buscar, consultar y manipular datos y el formato de los informes de salida. Por ejemplo, se pueden simular los resultados de una serie de selecciones hechas en menús para que los usuarios tengan una idea más clara de la forma como el constructor o los constructores del sistema están interpretando el problema. Si los usuarios no están convencidos de lo que se exhibe define con precisión sus necesidades, pueden modificar fácilmente las plantillas prototípico hasta que estén satisfechos. La creación de un modelo preliminar de prototípico en este punto produce varios beneficios: los usuarios pueden ver que se está avanzado, se les motiva para que participen activamente en la definición del problema, se mejora la comunicación entre todas las partes interesadas y se aclaran los equívocos en una etapa temprana del estudio de sistemas, antes de que se conviertan en costosos errores. Como se acaba de ver, puede ser necesario un proceso repetitivo (o *interactivo*) para terminar el paso de definición del problema. No existe un procedimiento definido que se deba seguir antes de que se pueda iniciar el análisis detallado del sistema. Un alto ejecutivo puede creer que existen diferencias de información. Puede preparar una declaración general de los objetivos y nombrar a un gerente para que realice un reconocimiento. Pueden realizarse varias sesiones de requerimientos para traducir los deseos generales a objetivos más específicos. Asimismo, pueden crearse y refinarse modelos preliminares de prototípico; se puede ampliar o reducir el alcance del estudio y es posible también que cambien los objetivos conforme se reúnan los datos. Una vez que parezca haberse logrado la aprobación en cuanto a la definición del problema, el equipo de reconocimiento deberá poner la definición detallada *por escrito* y enviarla a todas las personas interesadas, las cuales deberán aprobarla también por escrito. Si persisten diferencias, deberán resolverse en sesiones adicionales de requerimientos. Hay quienes se impacientan con los "retrasos" en el desarrollo del sistema causados por estas sesiones adicionales. Sin embargo, las personas más prudentes saben que los retrasos verdaderamente largos y costosos se presentan cuando los usuarios descubren, ya muy avanzados el proceso del desarrollo, que el sistema diseñado no es satisfactorio por haberse pasado por alto algunos requerimientos.

Diseño del Sistema:

El diseño de un sistema de información produce los elementos que establecen cómo el sistema cumplirá los requerimientos indicados durante el análisis de sistemas. A menudo los especialistas de sistemas se refieren a esta etapa como en *diseño lógico*, en contraste con desarrollo del software de programas, que se conoce como *diseño físico*. Los analistas de sistemas comienzan por identificar los informes y otras salidas que el sistema producirá. A continuación los datos específicos con éstos se señalan, incluyendo su localización exacta sobre el papel, la pantalla de despliegue u otro medio. Usualmente, los diseñadores dibujan la forma o la visualización como la esperan cuando el sistema esté terminado. El diseño del sistema también describe los datos calculados o almacenados que se introducirán. Los grupos de datos individuales y los procedimientos de cálculo se describen con detalle. Los diseñadores seleccionan las estructuras de los archivos y los dispositivos de almacenamiento, como son discos magnéticos, cintas magnéticas o incluso archivos en papel. Los procedimientos que ellos escriben muestran cómo se van a procesar los datos y a producir la salida. Los documentos que contienen las especificaciones de diseño utilizan muchas formas para representar los diseños, diagramas, tablas y símbolos especiales, algunos de los cuales el lector puede haber utilizado ya y otros que pudieran ser totalmente nuevos. La información del diseño detallado se pasa al grupo de programación para que pueda comenzar el desarrollo del software. Los diseñadores son responsables de proporcionar a los programadores las especificaciones completas y escritas con claridad, que establezcan lo que debe hacer el software. Conforme comienza la programación, los diseñadores están pendientes

para contestar preguntas, esclarecer ideas confusas y manejar los problemas que confronten los programadores cuando utilicen las especificaciones de diseño.

- **Diseño detallado (descomposición de los módulos en componentes elementales, especificación de los componentes elementales, diseño detallado de entradas y salidas, diseño detallado de la base de datos) (R, E)**

- **Construcción (programación estructurada u orientada a objetos, pruebas de componentes) (R, E)**

Desarrollo del Software:

Los desarrolladores del software pueden instalar o modificar; por ejemplo, software comercial que se haya comprado, o pueden escribir programas nuevos diseñados a la medida. La decisión de qué se va a hacer depende del costo de cada una de las opciones, el tiempo disponible para describir el software y la disponibilidad de programadores. En forma usual, en las grandes empresas los programadores de computadoras (o la combinación de analistas-programadores) son parte del grupo profesional permanente. Las compañías más pequeñas en donde los programadores permanentes no se han contratado, pueden obtener servicios externos de programación con base en un contrato. Los programadores también son responsables de documentar el programa e incluir los comentarios que expliquen tanto cómo y por qué se utilizó cierto procedimiento conforme se codificó de cierta forma. La documentación es esencial para probar el programa y darle mantenimiento una vez que la aplicación se ha puesto en marcha.

- **Pruebas (elaboración del plan de pruebas, preparación de casos, establecimiento de criterios de revisión, elaboración de las pruebas, evaluación de resultados) (R, E)**

Prueba de los Sistemas:

Durante la prueba, el sistema se utiliza en forma experimental para asegurar que el software no falle; es decir, que corra de acuerdo a sus especificaciones y a la manera que los usuarios esperan que lo haga. Se examinan datos especiales de prueba en la entrada del procesamiento y los resultados para localizar algunos problemas inesperados. Puede permitirse también a un grupo limitado de usuarios que utilice el sistema, de manera que los analistas puedan captar si tratan de utilizarlo en forma no planeadas. Es preferible detectar cualquier anomalía antes de que la empresa ponga en marcha el sistema y dependa de él. En muchas compañías la prueba se lleva a cabo por personas diferentes a aquellos que los escriben en forma original; es decir si se utilizan personas que no conocen como se diseñaron ciertas partes de los programas, se asegura una mayor y más completa prueba, además de ser imparcial, lo que da a un software más confiable.

- **Instalación (manual de usuario, ayudas interactivas, capacitación, corridas de prueba) (R, E)**

Puesta en Marcha:

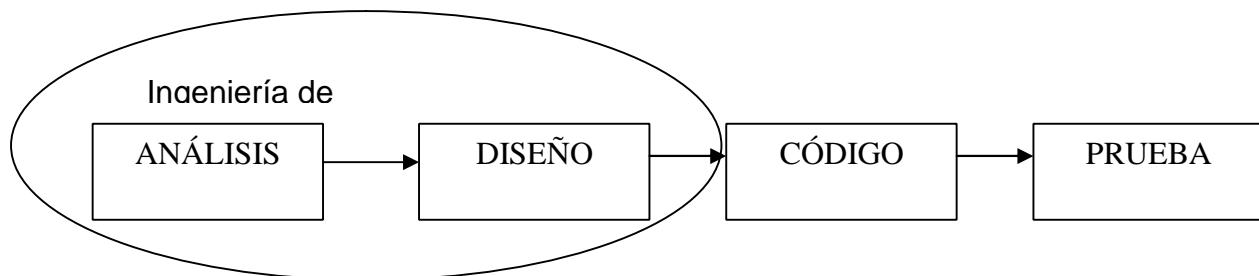
Cuando el personal de sistemas verifica y pone en uso el nuevo equipo, entrena al personal (usuario); instala la nueva aplicación y constituye los archivos de datos que se necesiten, entonces el sistema está puesto en marcha. De acuerdo con el tamaño de la empresa que empleará la aplicación y el riesgo asociado con su uso, los desarrolladores del sistema pueden escoger una prueba piloto para la operación del sistema solamente en un área de la compañía; por ejemplo, en un departamento o sólo con una o dos personas. A veces correrán en forma paralela tanto el sistema anterior como el nuevo para comparar los resultados de ambos; en otras situaciones, los desarrolladores pararán por completo el sistema anterior un día y al siguiente empezarán a utilizar el nuevo. Como se puede apreciar, cada estrategia para la puesta en marcha tiene sus méritos, que dependen de la situación del negocio considerado. Sin importar la estrategia para la puesta en marcha que se haya utilizado, los desarrolladores tendrán que asegurarse que el uso inicial del sistema esté libre de problemas. Una vez instalada, con frecuencia la aplicación se utiliza por muchos años; sin embargo, tanto la empresa como los usuarios cambiarán, y el medio ambiente será diferente también a través del tiempo. Por lo tanto, la aplicación indudablemente necesitará mantenimiento; es decir, se harán cambios y modificaciones al software, y a los archivos o procedimientos para cubrir los requerimientos nuevos de los usuarios. Los sistemas de la empresa y el medio ambiente de los negocios están en continuo cambio. Los sistemas de información deben mantenerse de la misma forma; es este sentido, la propuesta en marcha es un proceso continuo.

- **Mantenimiento (correctivo, adaptativo, perfectivo, pruebas de regresión, técnicas de mantenimiento) (R, E)**

2. Ciclos de vida de desarrollo de sistemas

• En cascada (R, E)

El modelo lineal secuencial para la ingeniería del software, llamado algunas veces ciclo de vida básico o modelo en cascada, sugiere un enfoque sistemático secuencial del desarrollo del software que comienza en un nivel de sistemas y progresar con el análisis, diseño, codificación, pruebas y mantenimiento. El modelo lineal secuencial acompaña a las actividades siguientes:



Ingeniería y modelado de sistemas/información:

El trabajo comienza estableciendo requisitos de todos los elementos del sistema y asignando al software algún subgrupo de estos requisitos. Esta visión del sistema es esencial cuando el software se debe interconectar con otros elementos como hardware, personas y bases de datos. La ingeniería y el análisis de sistemas acompaña a los requisitos que se recogen en el nivel del sistema con una pequeña parte de análisis y de diseño.

Análisis de los requisitos del software:

El proceso de reunión de requisitos se intensifica y se centra en el software. El ingeniero (analista) del software debe comprender el dominio de información del software, así como la función requerida, comportamiento, rendimiento e interconexión. El cliente documenta y repasa los requisitos del sistema.

Diseño:

El diseño del software es un proceso de mucho pasos que se centra en cuatro atributos distintos de un programa: estructura de datos, arquitectura del software, representaciones de interfaz y detalle procedimental (algoritmo). El proceso de diseño traduce requisitos en una representación del software que se pueda evaluar por calidad antes de que comience la generación del código. El diseño se documenta y se hace parte de la configuración del software.

Generación de código:

El diseño se debe traducir en una forma legible por la máquina. El paso de generación de código lleva a cabo esta tarea. Si se lleva a cabo un diseño detallado, la generación de código se realiza mecánicamente.

Pruebas:

Una vez que se ha generado un código, comienzan las pruebas del programa. El proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado y el sentirse seguro de que la entrada definida produzca resultados reales de acuerdo con los resultados requeridos.

Mantenimiento:

El software indudablemente sufrirá cambios después de ser entregado al cliente (una excepción posible es el software empotrado). Se producirán cambios porque se han encontrado errores o porque el software debe adaptarse para acoplarse a los cambios de su entorno externo o porque el cliente requiere mejoras funcionales o de rendimiento. El mantenimiento vuelve a aplicar cada una de las fases precedentes a un programa ya existente y no a uno nuevo. El modelo lineal secuencial es el paradigma más antiguo y más extensamente utilizado en la ingeniería del software. Sin embargo, la crítica ha puesto en duda su eficacia. Entre los problemas que se encuentran algunas veces en el modelo lineal secuencial están:

- ✓ Los proyectos reales raras veces siguen el modelo secuencial que propone el modelo.
- ✓ A menudo es difícil que el cliente exponga explícitamente todos los requisitos.
- ✓ El cliente debe tener paciencia. Una versión del programa no estarán disponibles hasta que el proyecto esté muy avanzado.
- ✓ Los responsables del desarrollo del software siempre se retrasan innecesariamente.

- **Espiral (R, E)**

El modelo en espiral es un modelo de proceso de software evolutivo que acompaña la naturaleza interactiva de construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal secuencial. Proporciona el potencial para el desarrollo rápido de versiones incrementales del software. Durante las primeras iteraciones, la versión incremental podría ser un modelo en papel o un prototipo. Durante las ultimas iteraciones se producen versiones cada vez más completas de ingeniería del sistema. El modelo en espiral se divide en un numero de actividades estructurales, también llamadas regiones de tareas. Generalmente, existen entre tres y seis regiones de tareas:

- ✓ **Comunicación con el cliente:** Las tareas requeridas para establecer comunicación entre el desarrollador y el cliente.
- ✓ **Planificación:** Las tareas requeridas para definir recursos, el tiempo y otras informaciones relacionadas con el proyecto.
- ✓ **Ánalisis de riesgos:** Las tareas requeridas para evaluar riesgos técnicos y de gestión.
- ✓ **Ingeniería:** Las tareas requeridas para construir una o más representaciones de la aplicación.
- ✓ **Construcción y adaptación:** Las tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario.
- ✓ **Evaluación del cliente:** Las tareas requeridas para obtener la reacción del cliente según la evaluación de las representaciones del software creadas durante la etapa de ingeniería e implementada durante la etapa de instalación. A diferencia del modelo de proceso clásico que termina cuando se entrega el software, el modelo en espiral puede adaptarse y aplicarse a lo largo de la vida del software de computadora. Un proyecto de desarrollo de conceptos comienza en el centro de la espiral y continuará hasta que se completa el desarrollo del concepto. El modelo en espiral utiliza la construcción de prototipos como mecanismos de reducción de riesgos, pero lo que es más importante, permite a quien lo desarrolla aplicar el enfoque de construcción de prototipos en cualquier etapa de evolución del producto.

II. Metodologías para el desarrollo de sistemas

1. Enfoque funcional (Yourdon)

- **Análisis estructurado (diagrama de flujo de datos, diccionario de datos, miniespecificaciones, diagramas de transición de estado) (R, E)**

Diagramas de flujos de datos:

Conforme con la información se mueve a través del software, se modifica mediante una serie de transformaciones. Un diagrama de flujos de datos (DFD), es una técnica grafica que describe el flujo de información y las transformaciones que se aplican a los datos, conforme se mueven de la entrada a la salida. El diagrama es similar en la forma a otros diagramas de flujo de actividades, y ha sido incorporado en técnicas de análisis y diseños propuesto por Yourdon y Constantine, DeMarco y Gane y Sarson. También se le conoce como un grafo de flujo de datos o un diagrama de burbujas.

Diccionario de datos:

Un análisis del dominio de la información puede ser incompleto si solo se considera el flujo de datos. Cada flecha de un diagrama de flujo de datos representa uno o más elementos de información. Por tanto, el analista debe disponer de algún otro método para representar el contenido de cada flecha de un DFD.

Se ha propuesto el diccionario de datos como una gramática casi formal para describir el contenido de los elementos de información y ha sido definido da la siguiente forma:

El diccionario de datos contiene las definiciones de todos los datos mencionados en el DFD, en una especificación del proceso y en el propio diccionario de datos. Los datos compuestos (datos que pueden ser además divididos) se definen en términos de sus componentes; los datos elementales (datos que no pueden ser divididos) se definen en términos del significado de cada uno de los valores que puede asumir. Por tanto, el diccionario de datos esta compuesto de definiciones de flujo de datos, archivos [datos almacenados] y datos usados en los procesos [transformaciones]...

Descripciones funcionales:

Una vez que ha sido representado el dominio de la información (usando un DFD y un diccionario de datos), el analista describe cada función (transformación) representada, usando el lenguaje natural o alguna otra notación

estilizada. Una de tales notaciones se llama inglés estructurado (también llamado lenguaje de diseño del programa o proceso(LDP)). El inglés estructurado incorpora construcciones procedimentales básicas –secuencia, selección y repetición- junto con frases del lenguaje natural, de forma que pueden desarrollarse descripciones procedimentales precisas de las funciones representadas dentro de un DFD.

Métodos orientados a la estructura de datos.

Hemos ya observado que el dominio de la información para un problema de software comprende el flujo de datos, el contenido de datos y la estructura de datos. Los métodos de análisis orientados a la estructura de datos representan los requerimientos del software enfocándose hacia la estructura de datos en vez de al flujo de datos. Aunque cada método orientado a la estructura de datos tiene un enfoque y notación distinta, todos tienen algunas características en común: 1) todos asisten al analista en la identificación de los objetos de información clave (también llamados entidades o ítems) y operaciones (también llamadas acciones o procesos); 2) todos suponen que la estructura de la información es jerárquica; 3) todos requieren que la estructura de datos se represente usando la secuencia, selección y repetición; y 4) todos dan una conjunto de pasos para transformar una estructura de datos jerárquica en una estructura de programa.

Como los métodos orientados al flujo de datos, los métodos de análisis orientados a la estructura de datos proporcionan la base para el diseño de software. Siempre puede extenderse un método de análisis para que abarque el diseño arquitectural y procedimental del software.

- **Diseño estructurado (diagramas de estructura, acoplamiento, cohesión, análisis de transformaciones, análisis de transacciones) (R, E)**

2. Enfoque de datos

- **Modelo entidad-relación (R)**

Entidades.

Se puede definir como entidad a cualquier objeto, real o abstracto, que existe en un contexto determinado o puede llegar a existir y del cual deseamos guardar información, por ejemplo: "PROFESOR", "CURSO", "ALUMNO". Las entidades las podemos clasificar en:

1. **Regulares:** aquellas que existen por sí mismas y que la existencia de un ejemplar en la entidad no depende de la existencia de otros ejemplares en otra entidad. Por ejemplo "EMPLEADO", "PROFESOR".
2. **Débiles:** son aquellas entidades en las que se hace necesaria la existencia de ejemplares de otras entidades distintas para que puedan existir ejemplares en esta entidad. Un ejemplo sería la entidad "ALBARÁN" que sólo existe si previamente existe el correspondiente pedido.

Atributos:

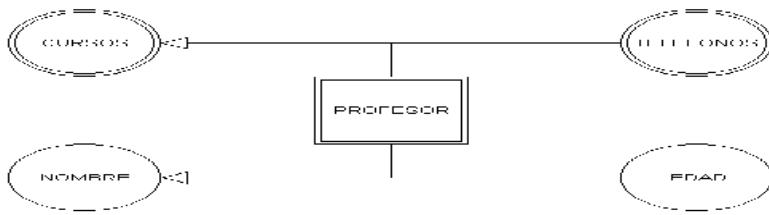
Las entidades se componen de atributos que son cada una de las propiedades o características que tienen las entidades. Cada ejemplar de una misma entidad posee los mismos atributos, tanto en nombre como en número, diferenciándose cada uno de los ejemplares por los valores que toman dichos atributos. Si consideramos la entidad "PROFESOR" y definimos los atributos Nombre, Teléfono y Salario, podríamos obtener los siguientes ejemplares:

{Luis García, 91.555.55.55, 80.500}
{Juan Antonio Alvarez, 91.666.66.66, 92.479}
{Marta López, 91.777.77.77, 85.396}

Existen cuatro tipos de atributos:

1. *Obligatorios:* aquellos que deben tomar un valor y no se permite ningún ejemplar no tenga un valor determinado en el atributo.
2. *Opcional:* aquellos atributos que pueden tener valores o no tenerlo.
3. *Monoevaluado:* aquel atributo que sólo puede tener un único valor.
4. *Multievaluado:* aquellos atributos que pueden tener varios valores.

Dentro del diagrama la entidad "PROFESOR" y sus atributos quedaría de la siguiente forma:



Existen atributos, llamados derivados, cuyo valor se obtiene a partir de los valores de otros atributos. Pongamos como ejemplo la entidad "PROFESOR" que tiene los atributos "NOMBRE", "FECHA DE NACIMIENTO", "EDAD"; el atributo "EDAD" es un atributo derivado por que se calcula a partir del valor del atributo "FECHA DE NACIMIENTO". Su representación gráfica es la siguiente:



En determinadas ocasiones es necesaria la descomposición de un atributo para definirlos en más de un dominio, podría ser el caso del atributo "TELÉFONO" que toma valores del dominio "PREFIJOS" y del dominio "NÚMEROS DE TELÉFONO". Estos atributos se representan de la siguiente forma:

Como complemento al diagrama de entidades del modelo de datos, podemos utilizar la siguiente plantilla para definir los diferentes atributos:

Nombre del atributo	FECHA DE NACIMIENTO
Tipo de dato	Número largo
Formato interno	aaaammdd
Longitud	8
Formato externo	dd/MM/aaaa
Descripción	Fecha de nacimiento del profesor
Dato requerido	SI
Permitir valor vacío	NO
Valor único	NO
Indexado	SI
Dominio	Calendario Gregoriano
Validaciones	La fecha debe ser superior a 01/01/1900
Confidencial	NO
Derechos de acceso	NO
Observaciones	...

Dominios:

Se define dominio como un conjunto de valores que puede tomar un determinado atributo dentro de una entidad. Por ejemplo:

Atributo	Dominio
Fecha de Alta	Calendario Gregoriano
Teléfono	Conjunto de números de teléfonos
Cobro de Incentivos	SI / NO
Edad	16 - 65

De forma casi inherente al término dominio aparece el concepto restricción para un atributo. Cada atributo puede adoptar una serie de valores de un dominio restringiendo determinados valores. El atributo "EDAD" toma sus valores del dominio N (números naturales) pero se puede poner como restricción aquellos que estén en el intervalo (0-120), pero dentro de la entidad "PROFESOR" se podría restringir aun más el intervalo, puesto que la edad mínima para trabajar es de 16 años y la máxima de 65, por lo tanto el intervalo sería (16-65).

Claves:

El modelo entidad - relación exige que cada entidad tenga un identificador, se trata de un atributo o conjunto de atributos que identifican de forma única a cada uno de los ejemplares de la entidad. De tal forma que ningún par de ejemplares de la entidad puedan tener el mismo valor en ese identificador. Un ejemplo de identificador es el atributo "DNI" que, en la entidad "ESPAÑOLES", identifica de forma única a cada uno de los españoles. Estos identificadores reciben en nombre de Identificador Principal (IP) o Clave Primaria (PK - Primary Key-). Se puede dar el caso de existir algún identificador más en la entidad, a estos identificadores se les denomina Identificadores Candidatos (IC). Los atributos identificadores de una entidad se representan en los diagramas de la siguiente forma:



Interrelaciones.

Se entiende por interrelación a la asociación, vinculación o correspondencia entre entidades. Por ejemplo, entre la entidad "PROFESOR" y la entidad "CURSO" podemos establecer la relación "IMPARTE" por que el profesor imparte cursos. Al igual que las entidades, las interrelaciones se pueden clasificar en regulares y débiles, según estén asociando dos tipos de entidades regulares o una entidad débil con otra de cualquier tipo. Las interrelaciones débiles se subdividen en dos grupos:

1. **En existencia:** cuando los ejemplares de la entidad débil no pueden existir si desaparece el ejemplar de la entidad regular del cual dependen.
2. **En identificación:** cuando, además de ser una relación en existencia, los ejemplares de la entidad débil no se pueden identificar por sí mismos y exigen añadir el identificador principal de la entidad regular del cual dependen para ser identificados.

Las interrelaciones, dentro de los diagramas, se representan de la siguiente forma:

- ✓ *Interrelación Regular:*
- ✓ *Interrelación en identidad*
- ✓ *Interrelación en existencia*

Restricciones en las Interrelaciones

- ✓ **Restricción de Exclusividad.** Dos o más interrelaciones son de exclusividad cuando cada ejemplar de la entidad presente en todas sólo puede combinarse con ejemplares de una sola de las entidades restantes.
- ✓ **Restricción de Exclusión.** Se produce una restricción de exclusión cuando los ejemplares de las entidades sólo pueden combinarse utilizando una interrelación.
- ✓ **Restricción de Inclusividad.** Se dice que una relación es de inclusividad cuando todo ejemplar de una entidad que participa en una interrelación ha tenido que participar en la otra.

Restricción de Inclusión. Se establece una restricción de inclusión cuando todo ejemplar de una entidad, para participar en la asociación con otro elemento de otra entidad mediante una interrelación, es necesario que ambos elementos estén asociados por una segunda interrelación.

• **Jackson System Development (R)**

• **Metodología Warnier-Orr (R)**

Diagramas de Warnier/Orr:

Los diagramas de Warnier/Orr son un tipo de diagramas jerárquicos que se utilizan para describir tanto la organización de datos como de procedimientos. Hay cuatro construcciones básicas utilizadas en los diagramas de W/O: jerarquía, secuencia, repetición, y selección. También hay dos conceptos avanzados que ocasionalmente son necesarios: concurrencia y recursión.

- ✓ **Jerarquía:** La jerarquía es la construcción más importante. Consiste simplemente en un grupo anidado de conjuntos y subconjuntos representados por un conjunto de llaves anidadas.

- ✓ Secuencia: La secuencia es la estructura más simple de un diagrama W/O. Dentro de un nivel de una jerarquía, las características listadas son presentadas en el orden en que ocurren.
- ✓ Repetición: La repetición es la representación del clásico loop en términos de programación. Para una estructura de datos significa que el mismo conjunto de datos se repite muchas veces. Para una estructura de proceso significa que el mismo conjunto de acciones se repiten muchas veces. La repetición es indicada colocando un para ordenado de números entre paréntesis debajo del conjunto repetitivo. Típicamente el par de números representa el mínimo y máximo número de veces que ocurre la repetición, aunque puede representar valores exactos.
- ✓ Selección: La selección representa una decisión u or exclusivo entre los conjuntos implicados.
- ✓ Concurrencia: Es usado siempre que la secuencia no es importante, y ocurren ambas cosas.
- ✓ Recursión: La recursión es la menos usada de las construcciones. Se utiliza cuando un conjunto contiene una versión de si mismo. Se simboliza con una doble llave.

Desarrollo de Sistemas Estructurados en Datos DSED (metodología Warnier-Orr):

A partir de las especificaciones de requisitos se realiza el diseño lógico y físico. El diseño lógico se centra en las salidas, en las interfaces, y en el diseño procedimental del software. El diseño físico surge del diseño lógico y se centra en el “empaquetamiento” del sw para conseguir lo mejor posible el rendimiento deseado, la facilidad de mantenimiento, y otras restricciones impuestas al diseño. El proceso de diseño lógico puede dividirse en dos actividades:

Derivación de la ELS:

1. Se evalúa la descripción del problema o la información relativa a los requisitos y se listan todos los elementos de datos diferentes, denominados átomos, que no puedan subdividirse más.
2. Se especifica la frecuencia de ocurrencia de cada átomo.
3. Se evalúan los elementos de datos compuestos llamados universales. Los universales son elementos de datos que están compuestos de otros universales y átomos.
4. Se desarrolla la representación diagramática de la ELS.

Derivación de la ELP:

1. Se quitan todos los átomos del diagrama de Warnier.
2. Se añaden los delimitadores BEGIN y END a todos los universales (repeticiones).
3. Se definen todas las instrucciones o procesos de inicialización y terminación (los begin-end).
4. Se especifican todos los cálculos o procesamientos no numéricos.
5. Se especifican todas las instrucciones y procesos de salida.
6. Se especifican todas las instrucciones y procesos de entrada.

3. Enfoque orientado a objetos

• Conceptos generales (características y conceptos del modelo de objetos, clases y objetos, relaciones entre objetos, relaciones entre clases) (R, E)

Teoría de objeto:

Para entender la idea fundamental que subyace la tecnología orientada a objetos comenzaremos por definir lo que es un objeto:

El término objeto surgió casi independientemente en varios campos de la informática, simultáneamente a principios de los sesenta, para referirse a nociones que eran diferentes en su apariencia, pero relacionados entre sí. Todas estas nociones se inventaron para manejar la complejidad de sistemas software de tal forma que los objetos representaban componentes de un sistema descompuesto modularmente o bien unidades modulares de representación del conocimiento. Desde temprana edad las personas nos formamos conceptos, cada concepto es una idea particular o una comprensión de nuestro mundo, los conceptos adquiridos nos permiten sentir y razonar a cerca de las cosas en el mundo. A estas cosas a las que se aplican nuestros conceptos se llaman objetos. Un objeto puede ser real o abstracto, como: Una factura, una organización, una figura de un programa, un avión, etc.

“Un objeto es cualquier cosa, real o abstracta, a cerca de la cual almacenamos datos y los métodos que controlan dichos datos.”

Un objeto puede estar compuesto por otros objetos, estos últimos a su vez, pueden estar compuestos de objetos, del mismo modo que una maquina esta formada por partes y estas, también, están formadas por otras partes. Esta

estructura intrincada de los objetos permite definir objetos muy complejos. Los objetos tienen una cierta integridad que no debería -de echo no puede- ser violada. Un objeto solo puede cambiar de estado, actuar, ser manipulado o permanecer en relación con otros objetos de maneras apropiadas para ser objeto. Las técnicas orientadas a objetos permiten que el software se construya a partir de objetos de comportamiento específico. Como lo mencionamos antes, los objetos se pueden construir a partir de otros, que a su vez pueden estar formados por otros objetos. Esto nos recuerda una maquinaria compleja, construida por partes, subpartes, sub-subpartes, etc. El análisis de sistemas en el mundo orientado a objetos se realiza al estudiar los objetos en un ambiente.

Los conceptos que poseemos se aplican a tipos determinados de objetos. Un tipo de objeto es una categoría de objeto. Un objeto es una instancia de un tipo de objeto.

Objeto se refiere a los datos y los métodos mediante los cuales se controlan a los propios datos. En el mundo OO, la estructura de datos y los métodos de cada tipo de objeto se manejan juntos. No se puede tener acceso o control de la estructura de datos excepto mediante los métodos que forman parte del tipo de objeto. Por ejemplo actividades como contratar, promover, retirar y despedir están ligadas íntimamente con el tipo de objeto empleado, puesto que cambian su estado. En otras palabras un objeto solo debe ser controlado por medio de las funciones asociadas con su tipo. Así, las operaciones no se pueden definir de manera adecuada sin los tipos de objetos.

Elementos fundamentales del análisis orientado a objetos

El modelo de objetos: El modelo de objetos describe la estructura de los objetos de un sistema: su identidad, sus relaciones con otros objetos, sus atributos y sus operaciones. Los cambios y las transformaciones no tienen sentido a menos que haya algo que cambiar o transformar. El modelo de objetos se representa gráficamente con diagramas de objetos y diagramas de instancias, que contienen clases de objetos e instancias respectivamente. Las clases se disponen en jerarquías que comparten una estructura de datos y un comportamiento común, y se relacionan con otras clases. Cada clase define los atributos que contiene cada uno de los objetos o instancias y las operaciones que realizan o sufren. La tecnología orientada a objetos se apoya en sólidos fundamentos de la ingeniería, cuyos elementos reciben el nombre global de modelo de objetos. El modelo de objetos abarca principios fundamentales que los detallaremos más adelante. Quede claro que el diseño orientado a objetos es fundamentalmente diferente a los enfoques de diseño estructurado tradicionales; requiere un modo distinto de pensar acerca de la descomposición y produce arquitecturas software muy alejadas del dominio de la cultura del diseño estructurado. Además, conviene mencionar que se observa que la mayoría de los programadores trabajan en un lenguaje y utilizan solo un estilo de programación. Programan bajo un paradigma apoyado por el lenguaje que usan. Frecuentemente, no se les han mostrado vías alternativas para pensar sobre un problema, y por lo tanto tiene dificultades para apreciar las ventajas de elegir un estilo, más apropiado para el problema que tienen entre manos. No hay un estilo de programación que sea el mejor para todo tipo de aplicaciones. Por ejemplo, la programación orientada a reglas sería la mejor para el diseño de una base de conocimiento, y la programación orientada a procedimientos sería la más indicada para el diseño de operaciones de cálculo intensivo. Por experiencia de algunos estudios de la materia, el estilo orientado a objetos es el más adecuado para el más amplio conjunto de aplicaciones; realmente, este paradigma de programación sirve con frecuencia como el marco de referencia arquitectónico en el que se emplean otros paradigmas. Cada uno de estos estilos de programación se basa en su propio marco de referencia conceptual. Cada uno requiere una actitud mental diferente, una forma distinta de pensar en el problema. Para todas las cosas orientadas a objetos, el marco de referencia conceptual es el modelo de objetos.

Fundamentos del modelo de objetos:

En realidad, el modelo de objetos ha recibido la influencia de una serie de factores, no solo de la programación orientada a objetos. El modelo de objetos ha demostrado ser un concepto u unificador de en la informática. La razón de este gran atractivo es simplemente que una orientación a objetos ayuda a combatir la complejidad inherente a muchos tipos de sistemas diferentes. El diseño orientado a objetos representa así un desarrollo evolutivo, no revolucionario; no rompe con los avances del pasado, sino que se basa en avances ya probados. Los métodos de diseño estructurado surgieron para guiar a los desarrolladores que intentaban construir sistemas complejos utilizando los algoritmos como bloques fundamentales para su construcción. Análogamente los métodos de diseño orientados a objetos han surgido para ayudar a los desarrolladores a explotar la potencia expresiva de los lenguajes de programación basados en objetos y orientados a objetos, utilizando las clases como bloques básicos de construcción. Desgraciadamente, hoy en día la mayoría de los programadores han sido educados formal e informalmente solo en los principios del diseño estructurado. En el modelo de objetos es necesario estudiar los principios fundamentales en los que se basa el análisis orientado a objetos, es decir;

- ✓ Abstracción

- ✓ Encapsulación
- ✓ Modularidad
- ✓ Jerarquía
- ✓ Concurrencia

Ninguno de estos principios es nuevo por sí mismo. Lo importante del modelo de objetos es el hecho de conjugar todos estos elementos en forma sinérgica. Al decir fundamentales, quiere decir que un modelo que carezca de cualquiera de estos elementos no estará orientado a objetos. Otros elementos que podrían llamarse secundarios, que quiere decir, que cada uno de ellos es una parte útil del modelo de objetos, pero no esenciales son:

- ✓ Tipos
- ✓ Persistencia

A partir de los elementos antes mencionados, trataremos de mostrar, mediante una definición en primer lugar de cada uno de ellos, cual es la asociación con el sistema solar y el sistema celular, esto quiere decir que asociaremos estos elementos bases de la llamada teoría de los objetos con el macrocosmo y el microcosmo, dictando sus características fundamentales para el entendimiento claro de este. A la vez, también definiremos las desigualdades u oposiciones de estos elementos con los sistemas antes mencionados. En otras palabras veremos como pasamos de un análisis estructurado a un análisis orientado a objetos, y viceversa. Macrocosmo v/s Microcosmo, esto incorporado al sistema de la teoría de los objetos.

Conceptos bases y elementales:

Las técnicas orientadas a objetos se basan en organizar el software como una colección de objetos discretos que incorporan tanto estructuras de datos como comportamiento. Esto contrasta con la programación convencional, en la que las estructuras de datos y el comportamiento estaban escasamente relacionadas. Las características principales del enfoque orientado a objetos son, en primer lugar:

- ✓ Objeto
- ✓ Clase
- ✓ Método
- ✓ Polimorfismo
- ✓ Encapsulación
- ✓ Herencia
- ✓ Mensajes
- ✓ Identidad
- ✓ Reusabilidad
- ✓ Jerarquía
- ✓ Abstracción
- ✓ Concurrencia
- ✓ Modularidad

A partir de estos elementos fundamentales, trataremos de dar un enfoque tanto estructurado como también un enfoque orientado a objetos, principal motivo de nuestro trabajo de investigación encomendado, debemos destacar que los elementos antes mencionados son un conjunto de información y bases que se investigó de libros o manuales alusivos al tema tratado, que de forma de organizar y abarcar distintas opiniones de especialistas en esta rama, formamos un todo para explicar y dar un buen entendimiento a la teoría del análisis orientado a los objetos. Para comenzar, daremos un enfoque básico y entendible como también, singular del material investigado, que es la definición según enciclopedias y diccionarios que nada tienen que ver con el análisis directamente orientado a objetos, esto con motivo de hacerlo práctico y de modo que el lector de esto se pueda familiarizar con estos componentes, básicos en nuestro trabajo de investigación, para luego y de una forma más complicada o a la vez, más asociada al trabajo en si, podamos asociarlos y compararlos con el mundo de la computación netamente orientado a objetos, como también con el sistema celular y solar.

Definiciones básicas:

- ✓ **OBJETO:** Fin, intento, propósito. Materia y sujeto de una ciencia.
- ✓ **CLASE:** Orden de cosas de una misma especie. Conjunto de órdenes.
- ✓ **POLIMORFISMO:** Propiedad de los cuerpos que cambian de forma sin cambiar su naturaleza. Presencia de distintas formas individuales en una sola especie.
- ✓ **ENCAPSULACION:** Proceso de constitución de una cápsula.

- ✓ HERENCIA: Derecho de suceder a otro la posesión de bienes o acciones.
- ✓ MENSAJE: Información que se le envía a alguien.
- ✓ METODO: Modo de hacer en orden una cosa, modo habitual de procede.
- ✓ IDENTIDAD: Cualidad de ser lo mismo que otra cosa con que se compara.
- ✓ REUSABILIDAD: Acción de volver a utilizar. Que puede volver a ser utilizado.
- ✓ JERARQUIA: Orden o grados de una especie.
- ✓ ABSTRACCION: Separación, apartamiento, aislamiento, prescindir.
- ✓ CONCURRENCIA: Asistencia, reunión simultanea de personas o cosas.
- ✓ MODULARIDAD: Acción de pasar de un termino a otro.

• Análisis orientado a objetos (el modelo de objetos, el modelo dinámico, el modelo funcional) (R, E)

Análisis estructurado de objetos:

Esta es una alternativa para que el análisis orientado a objetos clásico, utilice los productos del análisis estructurado como vía de entrada al diseño orientado a objetos. En esta aproximación, se comienza con un modelo esencial del sistema, tal como lo describen los diagramas de flujo de datos y otros productos del análisis estructurado. El análisis de la estructura de objetos (AOE) define las categorías de los objetos que percibimos y las formas en que los asociamos. Durante el análisis de las estructuras de objetos, el equipo de análisis se preocupa mas por identificar los tipos de objetos que por identificar los objetos individuales en unos sistemas. En el análisis de la estructura de objetos se identifica lo siguiente:

¿Qué son los tipos de objetos y como se asocian?

La identificación de los objetos y sus asociaciones se representan mediante esquemas de objetos. Esta información guía al diseñador en la definición de clases y estructuras de datos.

¿Cómo se organizan los tipos de objetos en superfinos y subtipos?

La jerarquía de generalizaciones puede organizar en diagramas e indicar al diseñador las direcciones de herencia.

¿Cuál es la composición de los objetos complejos?

Se pueden elaborar diagramas de jerarquías compuestas. La composición guía al diseñador en la definición de mecanismos que controlen adecuadamente a los objetos dentro de otros objetos.

Cuando el análisis pasa a la etapa del diseño de la estructura de objetos, identificamos las clases. Se definen las superclases, subclases, rutas de herencia y los métodos a utilizar y se lleva a cabo el detalle del diseño de la estructura de datos.

Análisis orientado a objetos:

La experiencia de algunos analistas nos lleva a aplicar en primer lugar el criterio orientado a objetos porque esta aproximación es mejor a la hora de servir de ayuda para organizar la complejidad innata de los sistemas de software, al igual que ha servido de ayuda para describir la complejidad organizada de sistemas complejos tan diversos como los computadores, plantas, galaxias o grandes instituciones sociales. Los sistemas orientados a objetos son también más resistentes al cambio y por lo tanto están mejor preparados para evolucionar en el tiempo, porque su diseño está basado en formas intermedias estables. El modelo de objetos ha influido incluso en las fases iniciales del ciclo de vida del desarrollo del software. El análisis orientado a objetos (AOO) enfatiza la construcción de modelos del mundo real utilizando una visión del mundo orientado a objetos: El análisis orientado a objetos es un método de análisis que examina los requisitos desde la perspectiva de las clases y objetos que se encuentran en el vocabulario del dominio del problema. Básicamente los productos del análisis orientado a objetos sirven como modelos de los que se puede partir para un diseño orientado a objetos; los productos del diseño orientado a objetos pueden utilizarse entonces como anteproyectos para la implementación completa de unos sistemas utilizando métodos de programación orientado a objetos, de esta forma se relacionan AOO, DOO y POO. Se insiste que se ha encontrado un gran valor en la construcción de modelos que se centran en las “cosas” que se encuentran en el espacio del problema formando lo que se ha llamado una descomposición orientada a objetos.

El diseño orientado a objetos es el método que lleva a una descomposición orientado a objetos. Ofrece un rico conjunto de modelos que reflejan la importancia de plasmar explícitamente las jerarquías de clases y de objetos de los sistemas que diseña. El análisis orientado a objetos (AOO) se basa en conceptos sencillos, conocidos desde la infancia y que aplicamos continuamente: objetos y atributos, él todo y las partes, clases y miembros. Puede parecer llamativo que se haya tardado tanto tiempo en aplicar estos conceptos al desarrollo de software. Posiblemente, una

de las razones es el éxito de los métodos de análisis estructurados, basados en los conceptos de flujo de información, que monopolizaron el análisis de sistemas de software durante los últimos veinte años.

El AOO ofrece un enfoque nuevo para el análisis de requisitos de sistemas software. En lugar de considerar el software desde una perspectiva clásica de entrada - proceso - salida, como los métodos estructurados clásicos se basan en modelar el sistema mediante los objetos que forman parte de él y las relaciones estáticas o dinámicas entre estos objetos. Este enfoque pretende conseguir modelos que se ajusten mejor al problema real a partir del conocimiento del llamado dominio del problema.

Ventajas del AOO:

- ✓ **Dominio del problema:** El paradigma OO es más que una forma de programar. Es una forma de pensar acerca de un problema desde el punto de vista del mundo real en vez de desde el punto de vista del ordenador. El AOO permite analizar mejor el dominio del problema, sin pensar en términos de implementar el sistema de un ordenador, permite, además, pasar directamente el dominio del problema al modelo del sistema.
- ✓ **Comunicación:** El concepto OO es más simple y está menos relacionado con la informática que el concepto de flujo de datos. Esto permite una mejor comunicación entre el analista y el experto en el dominio del problema.
- ✓ **Consistencia:** Los objetos encapsulan tanto atributos como operaciones. Debido a esto, el AOO reduce la distancia entre el punto de vista de los datos y el punto de vista del proceso, dejando menos lugar a inconsistencias disparidades entre ambos modelos.
- ✓ **Expresión de características comunes:** El paradigma lo utiliza la herencia para expresar explícitamente las características comunes de una serie de objetos; estas características comunes quedan escondidas en otros enfoques y llevan a duplicar entidades en el análisis y código en los programas. Sin embargo, el paradigma OO pone especial énfasis en la reutilización y proporciona mecanismos efectivos que permiten reutilizar aquello que es común sin impedir por ello describir las diferencias.
- ✓ **Resistencia al cambio:** Los cambios en los requisitos afectan notablemente a la funcionalidad de un sistema por lo que afectan mucho al software desarrollando con métodos estructurados. Sin embargo, los cambios afectan en mucha menos medida a los objetos que componen o maneja el sistema, que son mucho más estables. Las modificaciones necesarias para adaptar una aplicación basada en objetos a un cambio de requisitos suelen estar mucho más localizadas.
- ✓ **Reutilización:** Aparte de la reutilización interna, basada en la expresión explícita de características comunes, el paradigma OO desarrolla modelos mucho más próximos al mundo real, con lo que aumentan las posibilidades de reutilización. Es probable que en futuras aplicaciones nos encontremos con objetos iguales o similares a los de la actual.

Desventajas del AOO:

- ✓ **Descomposición funcional:** El análisis estructurado se basa fundamentalmente en la descomposición funcional del sistema que queremos construir. Esta descomposición funcional requiere traducir el dominio del problema en una serie de funciones y subfunciones, es decir, el analista debe comprender primero el dominio del problema y a continuación documentar las funciones y subfunciones que debe proporcionar el sistema. El problema es que no existe un mecanismo para comprobar si la especificación del sistema expresa con exactitud los requisitos del sistema.
- ✓ **Flujo de datos:** El análisis estructurado muestra como fluye la información a través del sistema. Aunque este enfoque se adapta bien al uso de sistemas informáticos para implementar al sistema, no es la forma habitual de pensar.
- ✓ **Modelo de datos:** El análisis estructurado moderno incorpora modelos de datos, además de modelos de procesos y de comportamiento. Sin embargo, la relación entre los modelos es muy débil, y hay muy poca influencia de un modelo en otro. En la práctica, los modelos de procesos y de datos de un mismo sistema se parecen muy poco. En muchos casos son visiones irreconciliables, no del mismo sistema sino de dos puntos de vista totalmente diferentes de organizar la solución. Lo ideal sería que ambos modelos se complementasen, no por oposición si no de forma que el desarrollo de uno facilitase el desarrollo del otro.

- **Diseño orientado a objetos (definición de la arquitectura del sistema, diseño del componente del dominio: de la interfase humana, del problema, de la administración de tareas y de la administración de datos, refinamiento de clases y objetos) (R, E)**

4. Herramientas CASE

Hoy en día, muchas empresas se han extendido a la adquisición de herramientas CASE (Ingeniería Asistida por Computadora), con el fin de automatizar los aspectos clave de todo el proceso de desarrollo de un sistema, desde el principio hasta el final e incrementar su posición en el mercado competitivo, pero obteniendo algunas veces elevados costos en la adquisición de la herramienta y costos de entrenamiento de personal así como la falta de adaptación de la herramienta a la arquitectura de la información y a las metodologías de desarrollo utilizadas por la organización. Por otra parte, algunas herramientas CASE no ofrecen o evalúan soluciones potenciales para los problemas relacionados con sistemas o virtualmente no llevan a cabo ningún análisis de los requerimientos de la aplicación. Sin embargo, el CASE proporciona un conjunto de herramientas semiautomatizadas y automatizadas que están desarrollando una cultura de ingeniería nueva para muchas empresas. Uno de los objetivos más importante del CASE (a largo plazo) es conseguir la generación automática de programas desde una especificación a nivel de diseño. Ahora bien, con la aparición de las redes de ordenadores en empresas y universidades ha surgido en el mundo de la informática la tecnología cliente / servidor. Son muchas de las organizaciones que ya cuentan con un número considerable de aplicaciones cliente / servidor en operación: Servidores de Bases de Datos y Manejadores de Objetos Distribuidos. Cliente / servidor es una tecnología de bajo costo que proporciona recursos compartidos, escalabilidad, integridad, encapsulamiento de servicios, etc. Pero al igual que toda tecnología, el desarrollo de aplicaciones cliente / servidor requiere que la persona tenga conocimientos, experiencia y habilidades en procesamiento de transacciones, diseño de base de datos, redes de ordenadores y diseño gráfica de interfase. El objeto de estudio está centrado en determinar ¿cuáles son las influencias de las herramientas CASE en las empresas desarrolladoras de sistemas de información cliente / servidor? Y ¿cuáles son las tendencias actuales de las empresas fabricantes de sistemas cliente / servidor?. Como soporte, se planteará un marco teórico que explicará la filosofía cliente / servidor, herramientas CASE y sistemas de información, y posteriormente se procederá a responder a las preguntas mencionadas anteriormente.

• Tipos (*uppercase, lowercase*) (R)

Clasificación de las herramientas case:

Como ya hemos comentado en los apartados precedentes CASE es una combinación de herramientas software (aplicaciones) y de metodologías de desarrollo :

- ✓ Las herramientas permiten automatizar el proceso de desarrollo del software.
- ✓ Las metodologías definen los procesos automatizar.
- ✓ Una primera clasificación del CASE es considerando su amplitud :
- ✓ TOOLKIT: es una colección de herramientas integradas que permiten automatizar un conjunto de tareas de algunas de las fases del ciclo de vida del sistema informático: Planificación estratégica, Análisis, Diseño, Generación de programas.
- ✓ WORKBENCH: Son conjuntos integrados de herramientas que dan soporte a la automatización del proceso completo de desarrollo del sistema informático. Permiten cubrir el ciclo de vida completo. El producto final aportado por ellas es un sistema en código ejecutable y su documentación.
- ✓ Una segunda clasificación es teniendo en cuenta las fases (y/o tareas) del ciclo de vida que automatizan:
- ✓ UPPER CASE: Planificación estratégica, Requerimientos de Desarrollo Funcional de Planes Corporativos.
- ✓ MIDDLE CASE: Análisis y Diseño.
- ✓ LOWER CASE: Generación de código, test e implantación

• Componentes (R)

De una forma esquemática podemos decir que una herramienta CASE se compone de los siguientes elementos:

- ✓ Repositorio (diccionario) donde se almacenan los elementos definidos o creados por la herramienta, y cuya gestión se realiza mediante el apoyo de un Sistema de Gestión de Base de Datos (SGBD) o de un sistema de gestión de ficheros.
- ✓ Metamodelo (no siempre visible), que constituye el marco para la definición de las técnicas y metodologías soportadas por la herramienta.
- ✓ Carga o descarga de datos, son facilidades que permiten cargar el repertorio de la herramienta CASE con datos provenientes de otros sistemas, o bien generar a partir de la propia herramienta esquemas de base de datos, programas, etc. que pueden, a su vez, alimentar otros sistemas. Este elemento proporciona así un medio de comunicación con otras herramientas.
- ✓ Comprobación de error es, facilidades que permiten llevar a cabo un análisis de la exactitud, integridad y consistencia de los esquemas generados por la herramienta.

- ✓ Interfaz de usuario, que constará de editores de texto y herramientas de diseño gráfico que permitan, mediante la utilización de un sistema de ventanas, iconos y menús, con la ayuda del ratón, definir los diagramas, matrices, etc. que incluyen las distintas metodologías.

III. Administración del desarrollo y la calidad del sistema

1. Administración de proyectos

• Estimación de tiempo y costo de cada actividad (R, E)

Duración estimada de cada actividad:

Estimar cuánto durará cada actividad desde el momento que se inicie hasta que se termine este tiempo o duración estimada tiene que ser el tiempo total transcurrido el tiempo que se haga el trabajo más cualquier tiempo de espera relacionado.

Cálculo de tiempos:

Una vez que se tiene la duración para cada actividad es importante que se determine si las actividades a realizarse serán realmente terminadas en el tiempo posible.

Tiempos de inicio y terminación más tempranos.

Conociendo la duración estimada para cada actividad en la red y utilizando como referencia el tiempo de inicio del proyecto se pueden calcular los tiempos siguientes para cada actividad:

- a) Tiempo de inicio más temprano TIT (ES Earliest Start Time). Es lo más pronto que se puede iniciar una actividad en particular, y calculada sobre la base de tiempo de inicio estimado del proyecto y de las duraciones estimadas de las actividades precedentes.
- b) Tiempo de terminación más temprano TTT (EF Earliest Finish Time). Es lo más pronto en que se puede terminar una actividad en particular y se calcula sumando la duración estimada de la actividad al tiempo de inicio más temprano de la misma.

$$EF = ES + \text{Duración estimada}$$

Los tiempos ES y EF se determinan calculando hacia delante, es decir, trabajando a través del diagrama de red desde el inicio del proyecto hasta el final del mismo. Al hacer esto se debe seguir una regla:

El tiempo de inicio más temprano de una actividad particular es el mayor tiempo de terminación más temprano de todas las precedentes inmediatas a esta actividad.

Tiempos de inicio y terminación más tardíos:

Con los tiempos de duración de cada actividad se podrá calcular los tiempos de inicio y terminación tardíos.

- a) Tiempo de terminación más tardío (LF). Es lo más tarde que se puede completar una actividad en particular para que todo el proyecto se termine en la fecha acordada.
- b) Tiempo de inicio más tardío (LS). Es la fecha más tardía en que se puede iniciar una actividad en particular para que todo el proyecto se complete en su fecha de terminación requerida.

$$LS = LF - \text{Duración estimada}$$

Los tiempos LS y LF se determinan calculando hacia atrás, es decir, trabajando el diagrama de red hasta el inicio del mismo, atendiendo la siguiente regla:

El tiempo de terminación más tardío para una actividad en particular es el menor de los tiempos de inicio más tardíos de todas las actividades que surgen directamente de esa en particular.

Estimación de costos del proyecto:

Los costos se estiman sobre el desarrollo de la propuesta por el contratista o el equipo de trabajo. La sección de costos de una propuesta probablemente consista en tablas de gastos estimados como son:

1. Mano de obra
2. Materiales
3. Alquiler de equipo o instalaciones
4. Subcontratistas o asesores

Elaboración del presupuesto del proyecto:

- a) El costo estimado del proyecto se asigna a los diversos paquetes de trabajo en la estructura de división del trabajo.
- b) El importe de cada paquete se distribuye a lo largo de su duración, con el fin de determinar cuánto del presupuesto se gastó en cualquier momento.

• Planificación temporal de proyectos (gráficas de GANTT y ruta crítica) (R, E)

Ruta crítica:

Es la ruta más larga que se tiene para concluir una actividad o proyecto desde que inicia hasta que llega a su actividad final. Al trayecto más largo en el diagrama de red global se le denomina "ruta crítica". Esta está determinada por los valores negativos que indicarán el trayecto más largo. Cuando existan valores positivos, en el cálculo de la holgura total son rutas no críticas. Los valores de cero o negativos en la holgura total se les denomina ruta crítica.

2. Aseguramiento de la calidad

• Concepto de calidad (R)

La calidad en los proyectos significa básicamente cumplir con los requerimientos del cliente. El hecho de incorporar calidad en los productos y servicios del proyecto requiere comprender los requerimientos del proyecto y luego orientar el trabajo hacia el logro del resultado final buscado. Calidad no significa exceder los requerimientos del cliente, sino satisfacerlos.

• Estándares (R)

ISO – Organización Internacional de Estándares.

• Especificaciones (R)

• Factores de calidad (corrección, confiabilidad, eficiencia, facilidad de uso, de prueba y de mantenimiento, adaptabilidad y flexibilidad, portabilidad, reuso, completud, facilidad de auditoría) (R)

La construcción de software requiere el cumplimiento de numerosas características. Entre ellas se destacan las siguientes:

- ✓ Eficiencia: La eficiencia de un software es su capacidad para hacer un buen uso de los recursos que manipula.
- ✓ Transportabilidad (portabilidad). La transportabilidad o portabilidad es la facilidad con la que un software puede ser transportado sobre diferentes sistemas físicos o lógicos.
- ✓ Verificabilidad (facilidad de verificación). La verificabilidad o facilidad de verificación de un software es su capacidad para soportar los procedimientos de validación y de aceptar juegos de test o ensayo de programas.
- ✓ Integridad. La integridad es la capacidad de un software a proteger sus propios componentes contra los procesos que no tenga el derecho de acceder.
- ✓ Facilidad de utilización. Un software es fácil de utilizar si se puede comunicar consigo de manera cómoda.
- ✓ Corrección (exactitud). Capacidad de los productos software de realizar exactamente las tareas definidas por su especificación.
- ✓ Robustez. Capacidad de los productos software de funcionar incluso en situaciones anormales.

• Métricas de calidad (diferencias entre el sistema y las especificaciones y estándares) (R)

• Revisiones técnicas (R)

IV. Industria del software

1. Proceso de desarrollo

• Pruebas betas (R)

Es la depuración de un producto utilizando personal ajeno a la compañía que desarrolla el producto. Las pruebas beta vienen después de las pruebas alfa, y antes de que salga la versión comercial.

Las versiones BETA corresponden a las versiones de prueba de un programa, previas a la versión final que es la que se vende, el software recién desarrollado se envía a potenciales clientes seleccionados de antemano y a usuarios finales experimentados, los cuales prueban la funcionalidad del nuevo programa y determinan si éste contiene fallas o errores operativos. Por lo general, la prueba beta es una de las últimas fases que el programador de software realiza antes de lanzar el producto comercialmente a un mercado. Sin embargo, si los usuarios de las pruebas beta comunican que el software tiene problemas operativos o un buen número de errores, el programador corrige las falencias y envía una nueva versión beta para ser evaluada por segunda vez.

Una prueba beta es la segunda fase de prueba del software, en la cual un muestreo del público al que está dirigido prueba el producto. (Beta es la segunda letra del alfabeto griego.)

Originalmente, el término "prueba alfa" indicaba la primera fase de prueba en un proceso de desarrollo de software. La primera fase incluye la prueba de unidades, de componentes y de sistema.

Las pruebas beta pueden ser consideradas una "prueba previa al lanzamiento". Las versiones de prueba beta de software actualmente se distribuyen a un amplio público en la Red, en parte para hacerle al programa una prueba "en el mundo real" y en parte para ofrecer una visión adelantada de lo que será el próximo lanzamiento.

• Control de versiones (R)

• Soporte técnico (R)

El soporte de sistemas es el mantenimiento permanente de un sistema después de que haya sido puesto en explotación. Ello incluye tanto el mantenimiento estricto de los programas como las posibles mejoras que pueden añadirse al sistema.

El soporte de sistemas es realizado por diseñadores y constructores de sistemas como apoyo a los usuarios de los sistemas. En nuestro modelo en pirámide, está relacionado con los bloques PERSONAS, DATOS, ACTIVIDADES y REDES implantados por medio de la TECNOLOGIA.

El soporte de sistemas constaba de cuatro actividades permanentes:

1. Corregir errores (también llamado mantenimiento).
2. Recuperar el sistema.
3. Asistir a los usuarios del sistema.
4. Adaptar el sistema ante nuevas necesidades (también llamado reingeniería).

TRATAMIENTO DE INFORMACION

A. Bases de datos

I. Modelado y diseño

1. Conceptos generales

• Niveles interno, conceptual y externo de las arquitecturas correspondientes (R)

Una base de datos es en esencia una colección de archivos relacionados entre sí, de la cual los usuarios pueden extraer información sin considerar las fronteras de los archivos. Un objetivo importante de un sistema de base de datos es proporcionar a los usuarios una visión abstracta de los datos, es decir, el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos. Sin embargo para que el sistema sea manejable, los datos se deben extraer eficientemente. Existen diferentes niveles de abstracción para simplificar la interacción de los usuarios con el sistema; Interno, conceptual y externo, específicamente el de almacenamiento físico, el del usuario y el del programador.

Nivel físico.

Es la representación del nivel más bajo de abstracción, en éste se describe en detalle la forma en como de almacenan los datos en los dispositivos de almacenamiento(por ejemplo, mediante señaladores o índices para el acceso aleatorio a los datos).

Nivel conceptual.

El siguiente nivel más alto de abstracción, describe que datos son almacenados realmente en la base de datos y las relaciones que existen entre los mismos, describe la base de datos completa en términos de su estructura de diseño. El nivel conceptual de abstracción lo usan los administradores de bases de datos, quienes deben decidir qué información se va a guardar en la base de datos.

Consta de las siguientes definiciones:

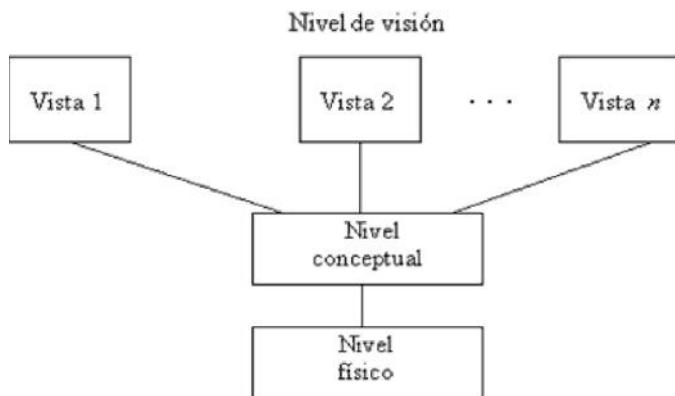
- ✓ Definición de los datos: Se describen el tipo de datos y la longitud de campo todos los elementos direccionables en la base. Los elementos por definir incluyen artículos elementales (atributos), totales de datos y registros conceptuales (entidades).
- ✓ Relaciones entre datos: Se definen las relaciones entre datos para enlazar tipos de registros relacionados para el procesamiento de archivos múltiples.

En el nivel conceptual la base de datos aparece como una colección de registros lógicos, sin descriptores de almacenamiento. En realidad los archivos conceptuales no existen físicamente. La transformación de registros conceptuales a registros físicos para el almacenamiento se lleva a cabo por el sistema y es transparente al usuario.

Nivel de visión.

Nivel más alto de abstracción, es lo que el usuario final puede visualizar del sistema terminado, describe sólo una parte de la base de datos al usuario acreditado para verla. El sistema puede proporcionar muchas visiones para la misma base de datos.

La interrelación entre estos tres niveles de abstracción se ilustra en la siguiente figura.



• Administración de la base de datos (R)

Administrador de Bases de Datos

Denominado por sus siglas como: DBA, Database Administrator.

Es la persona encargada y que tiene el control total sobre el sistema de base de datos, sus funciones principales son:

- ✓ **Definición de esquema.** Es el esquema original de la base de datos se crea escribiendo un conjunto de definiciones que son traducidas por el compilador de DDL a un conjunto de tablas que son almacenadas permanentemente en el diccionario de datos.
- ✓ **Definición de la estructura de almacenamiento del método de acceso.** Estructuras de almacenamiento y de acceso adecuados se crean escribiendo un conjunto de definiciones que son traducidas por el compilador del lenguaje de almacenamiento y definición de datos.
- ✓ **Concesión de autorización para el acceso a los datos.** Permite al administrador de la base de datos regular las partes de las bases de datos que van a ser accedidas por varios usuarios.
- ✓ **Especificación de limitantes de integridad.** Es una serie de restricciones que se encuentran almacenados en una estructura especial del sistema que es consultada por el gestor de base de datos cada vez que se realice una actualización al sistema.

Usuarios de las bases de datos.

Podemos definir a los usuarios como toda persona que tenga todo tipo de contacto con el sistema de base de datos desde que este se diseña, elabora, termina y se usa. Los usuarios que acceden una base de datos pueden clasificarse como:

- ✓ **Programadores de aplicaciones.** Los profesionales en computación que interactúan con el sistema por medio de llamadas en DML (Lenguaje de Manipulación de Datos), las cuales están incorporadas en un programa escrito en un lenguaje de programación (Por ejemplo, COBOL, PL/I, Pascal, C, etc.)
- ✓ **Usuarios sofisticados.** Los usuarios sofisticados interactúan con el sistema sin escribir programas. En cambio escriben sus preguntas en un lenguaje de consultas de base de datos.
- ✓ **Usuarios especializados.** Algunos usuarios sofisticados escriben aplicaciones de base de datos especializadas que no encajan en el marco tradicional de procesamiento de datos.
- ✓ **Usuarios ingenuos.** Los usuarios no sofisticados interactúan con el sistema invocando a uno de los programas de aplicación permanentes que se han escrito anteriormente en el sistema de base de datos, podemos mencionar al usuario ingenuo como el usuario final que utiliza el sistema de base de datos sin saber nada del diseño interno del mismo por ejemplo: un cajero.

2. Organización de archivos

• Secuencial indexado (E)

Dependiendo de la manera en que se accesen los registros de un archivo, se le clasifica como SECUENCIAL o como DIRECTO. En el caso de los archivos de ACCESO SECUENCIAL, para tener acceso al registro localizado en la posición N, se deben haber accedido los N-1 registros previos, en un orden secuencial. Cuando se tienen pocos registros en un archivo, o que los registros son pequeños, la diferencia entre los tiempos de acceso de forma secuencial y directa puede no ser perceptible para el usuario; sin embargo, la diferencia viene a ser significativa cuando se manejan archivos con grandes cantidades de información. La forma de manejar los archivos de acceso

secuencial es más sencilla en la mayoría de los lenguajes de programación, por lo que su estudio se antepone al de los archivos de acceso directo. El manejo secuencial de un archivo es recomendable cuando se deben procesar todos o la mayoría de los registros, como por ejemplo en los casos de una nómina de empleados o en la elaboración de reportes contables.

• Árboles B (R)

Los árboles B y los árboles B+ son casos especiales de árboles de búsqueda. Un árbol de búsqueda es un tipo de árbol que sirve para guiar la búsqueda de un registro, dado el valor de uno de sus campos. Los índices multinivel de la sección anterior pueden considerarse como variaciones de los árboles de búsqueda. Cada bloque o nodo del índice multinivel puede tener hasta p valores del campo de indexación y p punteros. Los valores del campo de indexación de cada nodo guían al siguiente nodo (que se encuentra en otro nivel), hasta llegar al bloque del fichero de datos que contiene el registro deseado. Al seguir un puntero, se va restringiendo la búsqueda en cada nivel a un subárbol del árbol de búsqueda, y se ignoran todos los nodos que no estén en dicho subárbol. Los árboles de búsqueda difieren un poco de los índices multinivel. Un árbol de búsqueda de orden p es un árbol tal que cada nodo contiene como mucho $p - 1$ valores del campo de indexación y p punteros:

$$(P_1, K_1, P_2, K_2, \dots, P_{q-1}, K_{q-1}, P_q)$$

donde $q \leq p$, cada P_i es un puntero a un nodo hijo y cada K_i es un valor de búsqueda proveniente de algún conjunto ordenado de valores. Se supone que todos los valores de búsqueda son únicos. Un árbol de búsqueda debe cumplir, en todo momento, las siguientes restricciones:

$$K_1 < K_2 < \dots < K_{q-1}$$

3. Dentro de cada nodo:

4. Para todos los valores X del subárbol al que apunta P_i , se tiene: $K_{i-1} < X < K_i$ para $1 < i < q$, $X < K_i$ para $i = 1$ y $K_{i-1} < X$ para $i = q$.

Al buscar un valor X siempre se sigue el puntero apropiado de acuerdo con las condiciones de la segunda restricción. Para insertar valores de búsqueda en el árbol y eliminarlos, sin violar las restricciones anteriores, se utilizan algoritmos que no garantizan que el árbol de búsqueda esté equilibrado (que todas las hojas estén al mismo nivel). Es importante mantener equilibrados los árboles de búsqueda porque esto garantiza que no habrá nodos en niveles muy profundos que requieran muchos accesos a bloques durante una búsqueda. Además, las eliminaciones de registros pueden hacer que queden nodos casi vacíos, con lo que hay un desperdicio de espacio importante que también provoca un aumento en el número de niveles. El árbol B es un árbol de búsqueda, con algunas restricciones adicionales, que resuelve hasta cierto punto los dos problemas anteriores. Estas restricciones adicionales garantizan que el árbol siempre estará equilibrado y que el espacio desperdiciado por la eliminación, si lo hay, nunca será excesivo. Los algoritmos para insertar y eliminar se hacen más complejos para poder mantener estas restricciones. No obstante, la mayor parte de las inserciones y eliminaciones son procesos simples, se complican sólo en circunstancias especiales: cuando se intenta insertar en un nodo que está lleno o cuando se intenta borrar en un nodo que está ocupado hasta la mitad. Un árbol B de orden p se define del siguiente modo:

8. La estructura de cada nodo interno tiene la forma:

$$(P_1, (K_1, Pr_1), P_2, (K_2, Pr_2), P_3, (K_3, Pr_3), \dots, P_{q-1}, (K_{q-1}, Pr_{q-1}), P_q)$$

donde $q \leq p$. Cada P_i es un puntero a un nodo interno del árbol y cada Pr_i es un puntero al registro del fichero de datos que tiene el valor K_i en el campo de búsqueda o de indexación.

9. Dentro de cada nodo se cumple:

10. Para todos los valores X del campo de indexación del subárbol al que apunta P_i , se cumple: $K_{i-1} < X < K_i$ para $1 < i < q$, $X < K_i$ para $i = 1$ y $K_{i-1} < X$ para $i = q$.

11. Cada nodo tiene, como mucho, p punteros a nodos del árbol.

12. Cada nodo, excepto la raíz y las hojas, tiene, al menos, $\lceil p/2 \rceil$ punteros a nodos del árbol. El nodo raíz tiene, como mínimo, dos punteros a nodos del árbol, a menos que sea el único nodo del árbol.

13. Un nodo con q punteros a nodos, $q \leq p$, tiene $q - 1$ valores del campo de indexación.

14. Todos los nodos hoja están al mismo nivel. Los nodos hoja tienen la misma estructura que los nodos internos, pero los punteros a nodos del árbol son nulos.

Como se puede observar, en los árboles B todos los valores del campo de indexación aparecen alguna vez en algún nivel del árbol, junto con un puntero al fichero de datos. En un árbol B+ los punteros a datos se almacenan sólo en los nodos hoja del árbol, por lo cual, la estructura de los nodos hoja difiere de la de los nodos internos. Los nodos hoja tienen una entrada por cada valor del campo de indexación, junto con un puntero al registro del fichero de datos. Estos nodos están enlazados para ofrecer un acceso ordenado a los registros a través del campo de indexación. Los nodos hoja de un árbol B+ son similares al primer nivel (nivel base) de un índice. Los nodos internos del árbol B+ corresponden a los demás niveles del índice. Algunos valores del campo de indexación se repiten en los nodos internos del árbol B+ con el fin de guiar la búsqueda. En un árbol B+ de orden p la estructura de un nodo interno es la siguiente:

7. Todo nodo interno es de la forma:

$$(P_1, K_1, P_2, K_2, P_3, K_3, \dots, P_{q-1}, K_{q-1}, P_q)$$

donde $q \leq p$. Cada P_i es un puntero a un nodo interno del árbol.

8. Dentro de cada nodo interno se cumple: $K_1 < K_2 < \dots < K_{q-1}$.

9. Para todos los valores X del campo de indexación del subárbol al que apunta P_i , se cumple: $K_{i-1} < X \leq K_i$ para $1 < i < q$, $X \leq K_i$ para $i = 1$ y $K_{i-1} < X$ para $i = q$.

10. Cada nodo interno tiene, como mucho, $\lceil p/2 \rceil$ punteros a nodos del árbol.

11. Cada nodo interno, excepto la raíz, tiene, al menos, $\lfloor p/2 \rfloor$ punteros a nodos del árbol. El nodo raíz tiene, como mínimo, dos punteros a nodos del árbol si es un nodo interno.

12. Un nodo interno con q punteros a nodos, $q \leq p$, tiene $q - 1$ valores del campo de indexación.

La estructura de los nodos hoja de un árbol B+ de orden p es la siguiente:

5. Todo nodo hoja es de la forma:

$$((K_1, Pr_1), (K_2, Pr_2), (K_3, Pr_3), \dots, (K_{q-1}, Pr_{q-1}), P_{siguiente})$$

donde $q \leq p$. Cada Pr_i es un puntero al registro de datos que tiene el valor K_i en el campo de indexación, y $P_{siguiente}$ es un puntero al siguiente nodo hoja del árbol.

6. Dentro de cada nodo hoja se cumple: $K_1 < K_2 < \dots < K_{q-1}$, $q \leq p$.

7. Cada nodo hoja tiene, al menos, $\lfloor p/2 \rfloor$ valores.

8. Todos los nodos hoja están al mismo nivel.

Como las entradas en los nodos internos de los árboles B+ contienen valores del campo de indexación y punteros a nodos del árbol, pero no contienen punteros a los registros del fichero de datos, es posible ``empaquetar'' más entradas en un nodo interno de un árbol B+ que en un nodo similar de un árbol B. Por tanto, si el tamaño de bloque

(nodo) es el mismo, el orden p será mayor para el árbol B+ que para el árbol B. Esto puede reducir el número de niveles del árbol B+, mejorándose así el tiempo de acceso. Como las estructuras de los nodos internos y los nodos hoja de los árboles B+ son diferentes, su orden p puede ser diferente. Se ha demostrado por análisis y simulación que después de un gran número de inserciones y eliminaciones aleatorias en un árbol B, los nodos están ocupados en un 69% cuando se estabiliza el número de valores del árbol. Esto también es verdadero en el caso de los árboles B+. Si llega a suceder esto, la división y combinación de nodos ocurrirá con muy poca frecuencia, de modo que la inserción y la eliminación se volverán muy eficientes. Cuando los árboles se definen sobre un campo no clave, los punteros a datos pasan a ser punteros a bloques de punteros a datos (se añade un nivel de indirección).

• Hashing asociativo (R)

A partir de una clave, y por medio de una función (f) encontrar el RID de ese registro. ¿Cuál debe ser la f?:

Algoritmo de Direcccionamiento:

f: C -----> 0,...N f(Cj)=n para cada valor no se repite

Se trata de buscar la función en la que se produzca el mínimo de sinónimos y huecos (distribuye los valores entre 0 y N). Las más utilizadas son las de números aleatorios y son:

1. PLEGAMIENTO se pasa el nº a binario y se suma la primera mitad del número con su segunda mitad y el resultado es la dirección.
2. CUADRADOS GENERALES elevamos el nº al cuadrado y se cogen las cifras centrales a conveniencia.
3. METODO DE CONGRUENCIA se sigue la fórmula $C=(a*C+b)\bmod N$
4. DESPLAZAMIENTO se van sumando las cifras, para que quede una cifra más pequeña (en sucesivos pasos).

Para evitar los sinónimos y los huecos se puede utilizar **Páginas de Overflow**, pero es muy lioso.

Entonces aparece el HASHING, se definen una serie de cubos para ir almacenando los valores.

V.Gr.

Se sigue la fórmula $((K \bmod 100) \bmod 2) + 1$

El resultado es el cubo donde hay que almacenarlo. Si el cálculo nos lleva a un cubo cuyos 10 registros están llenos, entonces se va a la página de overflow y se almacena allí. Lo primero que hay que controlar es el tamaño de los cubos, que será lo suficientemente grande para no utilizar mucho la página de overflow y lo suficientemente pequeño para que no queden muchos huecos. Los registros se pueden ordenar dentro del cubo y así las busquedas son más fáciles. La transformación de la clave para obtener la dirección, tiene que ser la más adecuada (pocos sinónimos). Los criterios para elegir el número de páginas de overflow (1/3 del nº de cubos).

• Métodos de acceso secundario (R, E)

Encadenada (por punteros) se utiliza para indexar por un campo que no es clave. El campo escogido será un campo con pocos valores (rango) pero que se repite mucho.

V.Gr. S# Ciudad

S1 Vigo

S2 Ourense

S3 Lugo

S4 Coruña

S5 Ourense

S6 Vigo

S7 Coruña

Se enlazan con punteros, con lo que las inserciones son mucho más rápidas y fáciles.

Esta técnica se utiliza porque al trabajar con las tablas de indices, si queremos insertar no es posible. Entonces se utiliza la Tabla Relocalizable:

3. Modelo entidad-relación

• Representación del modelo mediante el diagrama entidad-relación (R)

El modelo E-R se basa en una percepción del mundo real, la cual está formada por objetos básicos llamados entidades y las relaciones entre estos objetos así como las características de estos objetos llamados atributos.

Entidades y conjunto de entidades

Una entidad es un objeto que existe y se distingue de otros objetos de acuerdo a sus características llamadas atributos. Las entidades pueden ser concretas como una persona o abstractas como una fecha. Un conjunto de entidades es un grupo de entidades del mismo tipo. Por ejemplo el conjunto de entidades CUENTA, podría representar al conjunto de cuentas de un banco X, o ALUMNO representa a un conjunto de entidades de todos los alumnos que existen en una institución. Una entidad se caracteriza y distingue de otra por los atributos, en ocasiones llamadas propiedades, que representan las características de una entidad. Los atributos de una entidad pueden tomar un conjunto de valores permitidos al que se le conoce como dominio del atributo. Así cada entidad se describe por medio de un conjunto de parejas formadas por el atributo y el valor de dato. Habrá una pareja para cada atributo del conjunto de entidades. Ejemplo:

Hacer una descripción en pareja para la entidad alumno con los atributos No_control, Nombre y Especialidad.

Nombre_atributo, Valor
 No_control , 96310418
 Nombre , Sánchez Osuna Ana
 Esp , LI

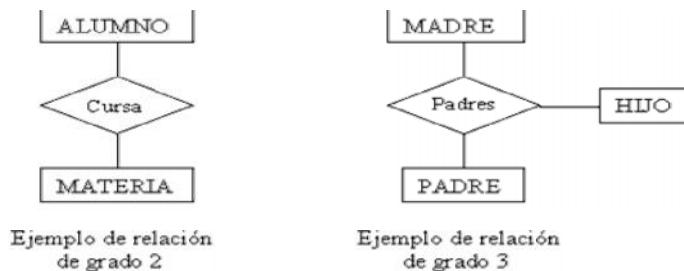
O considerando el ejemplo del Vendedor cuyos atributos son: RFC, Nombre, Salario.

Nombre_atributo, Valor
 RFC , COMD741101YHR
 Nombre , Daniel Colín Morales
 Salario , 3000

Relaciones y conjunto de relaciones.

Una relación es la asociación que existe entre dos o más entidades. Un conjunto de relaciones es un grupo de relaciones del mismo tipo. La cantidad de entidades en una relación determina el grado de la relación, por ejemplo la relación ALUMNO-MATERIA es de grado 2, ya que intervienen la entidad ALUMNO y la entidad MATERIA, la relación PADRES, puede ser de grado 3, ya que involucra las entidades PADRE, MADRE e HIJO. Aunque el modelo E-R permite relaciones de cualquier grado, la mayoría de las aplicaciones del modelo sólo consideran relaciones del grado 2. Cuando son de tal tipo, se denominan relaciones binarias. La función que tiene una relación se llama papel, generalmente no se especifican los papeles o roles, a menos que se quiera aclarar el significado de una relación.

Diagrama E-R (sin considerar los atributos, sólo las entidades) para los modelos exemplificados:



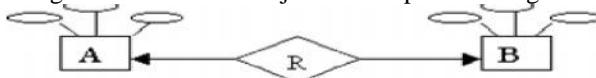
Limitantes de mapeo.

Existen 4 tipos de relaciones que pueden establecerse entre entidades, las cuales establecen con cuantas entidades de tipo B se pueden relacionar una entidad de tipo A:

Tipos de relaciones:

✓ Relación uno a uno.

Se presenta cuando existe una relación como su nombre lo indica uno a uno, denominado también relación de matrimonio. Una entidad del tipo A solo se puede relacionar con una entidad del tipo B, y viceversa. Por ejemplo: la relación asignación de automóvil que contiene a las entidades EMPLEADO, AUTO, es una relación 1 a 1, ya que asocia a un empleado con un único automóvil por lo tanto ningún empleado posee más de un automóvil asignado, y ningún vehículo se asigna a más de un trabajador. Es representado gráficamente de la siguiente manera:



A: Representa a una entidad de cualquier tipo diferente a una entidad B.

R: en el diagrama representa a la relación que existe entre las entidades.

El extremo de la flecha que se encuentra punteada indica el uno de la relación, en este caso, una entidad A ligada a una entidad B.

✓ Relación uno a muchos.

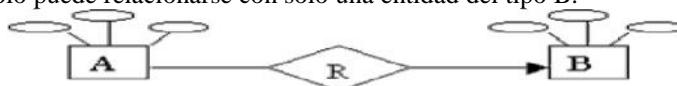
Significa que una entidad del tipo A puede relacionarse con cualquier cantidad de entidades del tipo B, y una entidad del tipo B solo puede estar relacionada con una entidad del tipo A. Su representación gráfica es la siguiente:



Nótese en este caso que el extremo punteado de la flecha de la relación de A y B, indica una entidad A conectada a muchas entidades B.

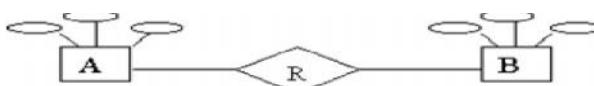
✓ Muchos a uno.

Indica que una entidad del tipo B puede relacionarse con cualquier cantidad de entidades del tipo A, mientras que cada entidad del tipo A solo puede relacionarse con solo una entidad del tipo B.



✓ Muchas a muchas.

Establece que cualquier cantidad de entidades del tipo A pueden estar relacionados con cualquier cantidad de entidades del tipo B.



A los tipos de relaciones antes descritos, también se le conoce como cardinalidad. La cardinalidad nos especifica los tipos de relaciones que existen entre las entidades en el modelo E-R y establecer con esto las validaciones necesarias para conseguir que los datos de la instancia (valor único en un momento dado de una base de datos) correspondan con la realidad.

Algunos ejemplos de cardinalidades de la vida común pueden ser:

✓ Uno a uno.

El noviazgo, el RFC de cada persona, El CURP personal, El acta de nacimiento, ya que solo existe un solo documento de este tipo para cada una de las diferentes personas.

✓ Uno a muchos.

Cliente – Cuenta en un banco, Padre-Hijos, Camión-Pasajeros, zoologico- animales, árbol – hojas.

✓ Muchos a muchos.

Arquitecto – proyectos, fiesta – personas, estudiante – materias.

NOTA: Cabe mencionar que la cardinalidad para cada conjunto de entidades depende del punto de vista que se le da al modelo en estudio, claro esta, sujetándose a la realidad.

Otra clase de limitantes lo constituye la dependencia de existencia.

Refiriéndonos a las mismas entidades A y B, decimos que si la entidad A depende de la existencia de la entidad B, entonces A es dependiente de existencia por B, si eliminamos a B tendríamos que eliminar por consecuente la entidad A, en este caso B es la entidad Dominante y A es la entidad subordinada.

Llaves primarias.

Como ya se ha mencionado anteriormente, la distinción de una entidad entre otra se debe a sus atributos, lo cual lo hacen único. Una llave primaria es aquel atributo el cual consideramos clave para la identificación de los demás atributos que describen a la entidad. Por ejemplo, si consideramos la entidad ALUMNO del Instituto Tecnológico de La Paz, podríamos tener los siguientes atributos: Nombre, Semestre, Especialidad, Dirección, Teléfono, Número de control, de todos estos atributos el que podremos designar como llave primaria es el número de control, ya que es diferente para cada alumno y este nos identifica en la institución. Claro que puede haber más de un atributo que

pueda identificarse como llave primaria en este caso se selecciona la que consideremos más importante, los demás atributos son denominados llaves secundarias. Una clave o llave primaria es indicada gráficamente en el modelo E-R con una línea debajo del nombre del atributo.

Diagrama Entidad-Relación

Denominado por sus siglas como: E-R; Este modelo representa a la realidad a través de un esquema gráfico empleando los términos de entidades, que son objetos que existen y son los elementos principales que se identifican en el problema a resolver con el diagramado y se distinguen de otros por sus características particulares denominadas atributos, el enlace que une las entidades esta representada por la relación del modelo. Recordemos que un rectángulo nos representa a las entidades; una elipse a los atributos de las entidades, y una etiqueta dentro de un rombo nos indica la relación que existe entre las entidades, destacando con líneas las uniones de estas y que la llave primaria de una entidad es aquel atributo que se encuentra subrayado. A continuación mostraremos algunos ejemplos de modelos E-R, considerando las cardinalidades que existen entre ellos:

✓ Relación Uno a Uno.

Problema: Diseñar el modelo E-R, para la relación Registro de automóvil que consiste en obtener la tarjeta de circulación de un automóvil con los siguientes datos:- Automóvil- Modelo, Placas, Color - Tarjeta de circulación - Propietario, No_serie, Tipo.



Indicamos con este ejemplo que existe una relación de pertenencia de uno a uno, ya que existe una tarjeta de circulación registrada por cada automóvil. En este ejemplo, representamos que existe un solo presidente para cada país.



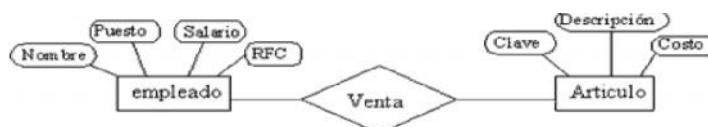
✓ Relación muchos a muchos.

El siguiente ejemplo indica que un cliente puede tener muchas cuentas, pero que una cuenta puede llegar a pertenecer a un solo cliente (Decimos puede, ya que existen cuentas registradas a favor de más de una persona).



Reducción de diagramas E-R a tablas

Un diagrama E-R, puede ser representado también a través de una colección de tablas. Para cada una de las entidades y relaciones existe una tabla única a la que se le asigna como nombre el del conjunto de entidades y de las relaciones respectivamente, cada tabla tiene un número de columnas que son definidas por la cantidad de atributos y las cuales tienen el nombre del atributo. La transformación de nuestro ejemplo Venta en la que intervienen las entidades de Vendedor con los atributos RFC, nombre, puesto, salario y Artículo con los atributos Clave, descripción, costo. Cuyo diagrama E-R es el siguiente:



Entonces las tablas resultantes siguiendo la descripción anterior son:

Tabla Empleado

Nombre	Puesto	Salario	RFC
Teófilo	Vendedor	2000	TEAT701210XYZ
Cesar	Auxiliar ventas	1200	COV741120ABC

Tabla artículo

Clave	Descripción	Costo
A100	Abanico	460
C260	Colcha matrimonial	1200

Tabla Venta

RFC	Clave
TEAT701210XYZ	C260
COV741120ABC	A100

Nótese que en la tabla de relación - Venta -, contiene como atributos a las llaves primarias de las entidades que intervienen en dicha relación, en caso de que exista un atributo en las relaciones, este atributo es anexado como una fila más de la tabla. Por ejemplo si anexamos el atributo fecha a la relación venta, la tabla que se originaría sería la siguiente:

RFC	Clave	Fecha
TEAT701210XYZ	C260	10/12/96
COV741120ABC	A100	11/12/96

Generalización y especialización

Generalización.

Es el resultado de la unión de 2 o más conjuntos de entidades (de bajo nivel) para producir un conjunto de entidades de más alto nivel. La generalización se usa para hacer resaltar los parecidos entre tipos de entidades de nivel más bajo y ocultar sus diferencias. La generalización consiste en identificar todos aquellos atributos iguales de un conjunto de entidades para formar una entidad(es) global(es) con dichos atributos semejantes, dicha entidad(es) global(es) quedara a un nivel más alto al de las entidades origen.

Ejemplo:

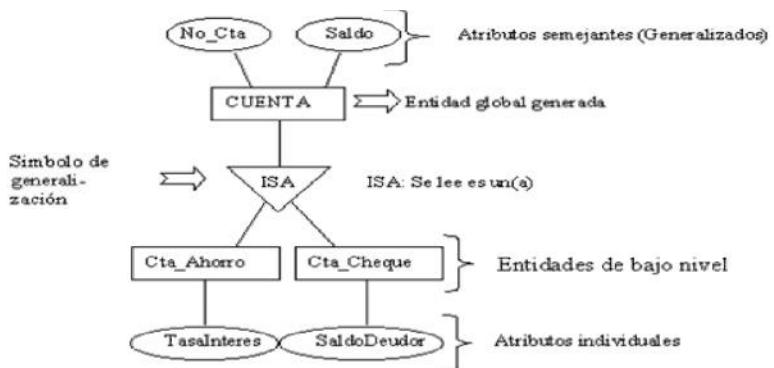
Tomando el ejemplo del libro de fundamentos de base de datos de Henry F. Korth.

Donde:

Se tiene las entidades Cta_Ahorro y Cta_Cheques, ambas tienen los atributos semejantes de No_Cta y Saldo, aunque además de estos dos atributos, Cta_Ahorro tiene el atributo Tasa_Interes y Cta_Cheques el atributo Saldo_Deudor. De todos estos atributos podemos juntar (generalizar) No_Cta y Saldo que son iguales en ambas entidades.

Entonces tenemos:

Podemos leer esta gráfica como: La entidad Cta_Ahorro hereda de la entidad CUENTA los atributos No_Cta y saldo, además del atributo de TasaInteres, de forma semejante Cta_Cheque tiene los atributos de No_Cta, Saldo y SaldoDeudor.



Como podemos observar la Generalización trata de eliminar la redundancia (repetición) de atributos, al englobar los atributos semejantes. La entidad(es) de bajo nivel cuentan (heredan) todos los atributos correspondientes.

Especialización:

Es el resultado de tomar un subconjunto de entidades de alto nivel para formar un conjunto de entidades de más bajo nivel.

- ✓ En la generalización cada entidad de alto nivel debe ser también una entidad de bajo nivel. La especialización no tiene este límite.
- ✓ se representa por medio de un triángulo denominado con la etiqueta "ISA", se distingue de la generalización por el grosor de las líneas que conectan al triángulo con las entidades.
- ✓ La especialización denota la diferencia entre los conjuntos de entidades de alto y bajo nivel.

Agregación.

La agregación surge de la limitación que existe en el modelado de E-R, al no permitir expresar las relaciones entre relaciones de un modelo E-R en el caso de que una relación X se quiera unir con una entidad cualquiera para formar otra relación. La Generalización consiste en agrupar por medio de un rectángulo a la relación (representada por un rombo) junto con las entidades y atributos involucrados en ella, para formar un grupo que es considerado una entidad y ahora sí podemos relacionarla con otra entidad. Para exemplificar lo anterior consideremos el ejemplo del libro de fundamentos de Base de Datos de Henry F. Korth. En donde el problema consiste en que existen trabajando muchos empleados que trabajan en diferentes proyectos, pero dependiendo del trabajo que realiza en pueden llegar a utilizar un equipo o maquinaria; en este problema intervienen 3 entidades: Empleado, Proyecto y Maquinaria, el diagrama E-R correspondiente es:

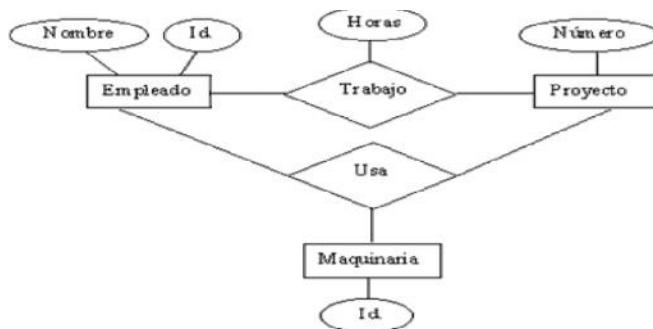


Diagrama E-R con relaciones redundantes

Como el modelo E-R no permite la unión entre dos o más relaciones, la relación trabajo es englobada como si fuera una entidad más de la relación usa, gráficamente queda como:

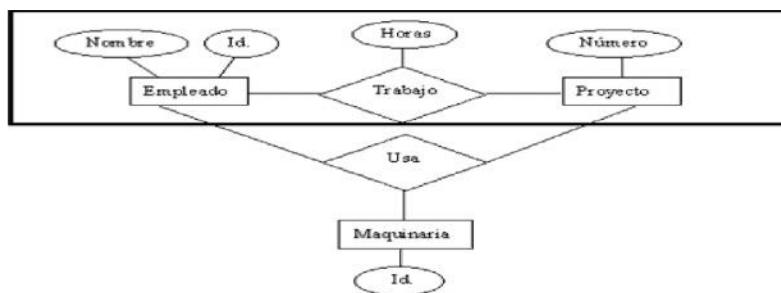


Diagrama E-R con agregación

Ahora podemos decir que la entidad trabajo se relaciona con la entidad maquinaria a través de la relación usar. Para indicarnos que un trabajo usa un determinado equipo o maquinaria según el tipo de trabajo que se trate.

4. El modelo jerárquico

- **Modelo conceptual (R)**

- **Estructura de almacenamiento (R)**

Una base de datos jerárquica consiste en una colección de registros que se conectan entre sí por medio de enlaces. Los registros son similares a los expuestos en el modelo de red. Cada registro es una colección de campos (atributos), que contienen un solo valor cada uno de ellos. Un enlace es una asociación o unión entre dos registros exclusivamente. Por tanto, este concepto es similar al de enlace para modelos de red. Consideremos la base de datos, nuevamente, que contiene la relación alumno - materia de un sistema escolar. Existen dos tipos de registros en este sistema, alumno y materia. El registro alumno consta de tres campos: NombreA, Control y Esp; El registro Materia esta compuesto de tres campos: Clave, NombreM y Cred. En este tipo de modelos la organización se establece en forma de árbol, donde la raíz es un nodo ficticio. Así tenemos que, una base de datos jerárquica es una colección de árboles de este tipo. El contenido de un registro específico puede repetirse en varios sitios(en el mismo árbol o en varios árboles).

La repetición de los registros tiene dos desventajas principales:

- ✓ Puede producirse una inconsistencia de datos
- ✓ El desperdicio de espacio.

Diagramas de estructura de árbol

Un diagrama de estructura de árbol es la representación de un esquema de la base de datos jerárquica, de ahí el nombre, ya que un árbol está desarrollado precisamente en orden descendente formando una estructura jerárquica.

Este tipo de diagrama está formado por dos componentes básicos:

- ✓ Rectángulos: que representan a los de registros.
- ✓ Líneas: que representan a los enlaces o ligas entre los registros.

Un diagrama de árbol tiene el propósito de especificar la estructura global de la base de datos. Un diagrama de estructura de árbol es similar a un diagrama de estructura de datos en el modelo de red. La principal diferencia es que en el modelo de red los registros se organizan en forma de un grafo arbitrario, mientras que en modelo de estructura de árbol los registros se organizan en forma de un árbol con raíz.

Características de las estructuras de árbol:

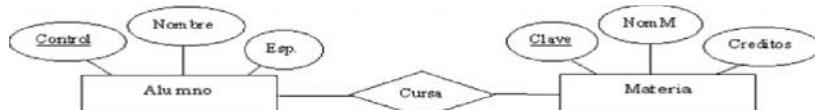
- ✓ El árbol no puede contener ciclos.
- ✓ Las relaciones que existen en la estructura deben ser de tal forma que solo existan relaciones muchos a uno o uno a uno entre un padre y un hijo.



Diagrama de estructura de árbol

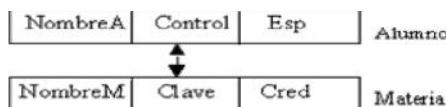
En este diagrama podemos observar que las flechas están apuntando de padres a hijos. Un padre (origen de una rama) puede tener una flecha apuntando a un hijo, pero un hijo siempre puede tener una flecha apuntando a su padre. El esquema de una base de datos se representa como una colección de diagramas de estructura de árbol. Para cada diagrama existe una única instancia de árbol de base de datos. La raíz de este árbol es un nodo ficticio. Los hijos de ese nodo son instancias de los registros de la base de datos. Cada una de las instancias que son hijos pueden tener a su vez, varias instancias de varios registros. Las representaciones según las cardinalidades son:

Consideremos la relación alumno-materia sin atributo descriptivo.

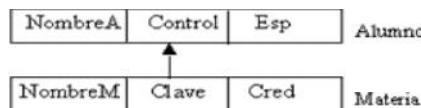


La transformación según las cardinalidades seria:

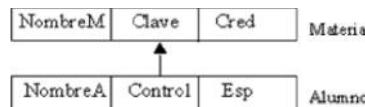
Cuando la relación es uno a uno.



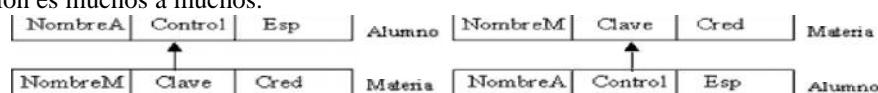
Cuando la relación es uno a muchos.



Cuando la relación es muchos a uno.



Cuando la relación es muchos a muchos.

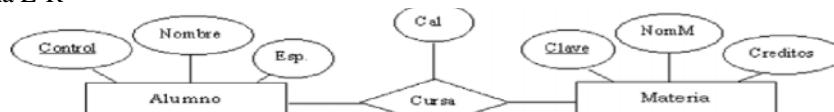


Cuando la relación tiene atributos descriptivos, la transformación de un diagrama E-R a estructura de árbol se lleva a cabo cubriendo los siguientes pasos:

- ✓ Crear un nuevo tipo de registro.
- ✓ Crear los enlaces correspondientes.

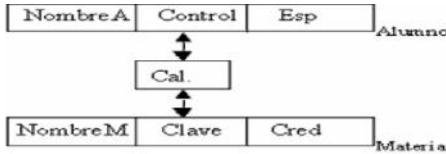
Consideremos que a la relación Alumno-Materia añadimos el atributo Cal a la relación que existe entre ambas, entonces nuestro modelo E-R resulta:

Añadir el diagrama E-R

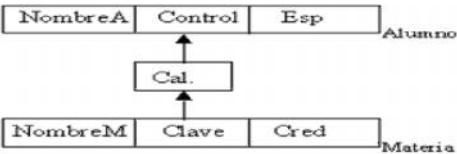


Según las cardinalidades los diagramas de estructura de árbol pueden quedar de la siguiente manera:

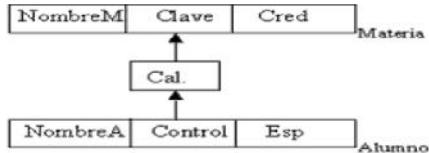
Cuando la relación es uno a uno.



Cuando la relación es uno a muchos.



Cuando la relación es Muchos a uno.



Cuando la relación es Muchos a Muchos.

Si la relación es muchos a muchos entonces la transformación a diagramas de árbol es un poco más compleja debido a que el modelo jerárquico solo se pueden representar las relaciones uno a uno o uno a muchos. Existen varias formas distintas de transformar este tipo de relaciones a estructura de árbol, sin embargo todas las formas constituyen la repetición de algunos registros. La decisión de qué método de transformación debe utilizarse depende de muchos factores, entre los que se incluyen:

- ✓ El tipo de consultas esperadas en la base de datos.
- ✓ El grado al que el esquema global de base de datos que se está modelando se ajusta al diagrama E-R dado.

A continuación se describe la forma de transformar un diagrama E-R a estructura de árbol con relaciones muchos a muchos. Suponemos el ejemplo de la relación alumno-materia.

Crear dos diagramas de estructura de árbol distintos T1 y T2, cada uno de los cuales incluye los tipos de registro alumno y materia, en el árbol T1 la raíz es alumno y en T2 la raíz es materia. Crear los siguientes enlaces:

- ✓ Un enlace muchos a uno del registro cuenta al registro Alumno, en T1
- ✓ Un enlace muchos a uno del tipo de registro cliente al tipo de registro materia en T2.

Como se muestra en el siguiente diagrama:



Recuperación de datos

Para manipular la información de una base de datos jerárquico, es necesario emplear un lenguaje de manipulación de datos, el lenguaje consta de varias órdenes que están incorporadas en un lenguaje principal, Pascal.

La orden Get

La recuperación de datos se realiza mediante esta orden, se realizan las siguientes acciones:

- ✓ Localiza un registro en la base de datos y actualiza a un puntero que es el que mantiene la dirección del último registro accesado.
- ✓ Copia los datos solicitados a un tipo de registro apropiado para la consulta.
- ✓ La orden Get debe especificar en cual de los árboles de la base de datos se va a buscar.

Para revisar todos los registros de forma consistente, se debe imponer un orden en los registros. El que normalmente se usa es el preorden, el cuál consiste en iniciar la búsqueda por la raíz y continua buscando por los subárboles de izquierda a derecha recursivamente. Así pues, empezamos por la raíz, visitamos el hijo más a la izquierda, y así sucesivamente, hasta que alcancemos un nodo hoja (sin hijos). A continuación movemos hacia atrás al padre de la hoja y visitamos el hijo más a la izquierda no visitado. Continuamos de esta forma hasta que se visite todo el árbol.

Existen dos ordenes Get diferentes para localizar registros en un árbol de base de datos. La orden más simple tiene la forma:

```
get first <Tipo de registro>
where <Condición>
```

La cláusula where es opcional. La <Condición> que se adjunta es un predicado que puede implicar a cualquier tipo de registro que sea un antecesor de <tipo registro> o el <Tipo de registro> mismo. La orden get localiza el primer registro (en preorden) del tipo <Tipo registro> en la base de datos que satisfaga la <condición > de la cláusula where. Si se omite la cláusula where, entonces se localiza el primer registro del tipo <Tipo registro>. Una vez que se encuentra ese registro, se hace que el puntero que tiene la dirección del último registro accesado apunte a ese registro y se copia el contenido del registro en un registro apto para la consulta.

Para ilustrar lo anterior veamos los ejemplos:

Realicemos una consulta que imprima El nombre del alumno llamado Luis A. (consideraremos la relación Alumno-Materia que hemos estado manejando.)

```
get first Alumno
where Alumno.NombreA="Luis A.";
print (Alumno.Control)
```

Ahora consideremos que deseamos la consulta de los nombres de las materias en donde el alumno de nombre Luis A. A obtenido una calificación igual a 100 (si es que existe)

```
get first Alumno
where Alumno.NombreA="Luis A." and Cursa.cal= 100;
if DB-Status=0 then print (Materia.NombreM);
```

La condición involucra a la variable DB-Status, la cual nos indica si se encontró o no el registro. La orden get first, solo nos muestra el primer registro encontrado que satisfaga la orden de consulta, sin embargo puede haber más de ellos, para localizar a los demás registros empleamos la orden Get next.

Cuya estructura es:

```
get next <Tipo de registro>
where <Condición>
```

La cual localiza el siguiente registro en preorden que satisface la condición. Si se omite la cláusula where, entonces se localiza el siguiente registro del tipo <Tipo registro>.

Actualización de datos.

Creación de nuevos registros. La orden utilizada para la inserción de registros es:

```
insert <tipo de registro>
where <Condición>
```

donde: tipo registro contiene los datos de los campos del registro a insertar.

Sí se incluye la cláusula where, el sistema busca en el árbol de la base de datos (en preorden) un registro que satisfaga la condición dada, una vez encontrado, el registro creado se inserta en el árbol como un hijo más a la izquierda. Si se omite la cláusula where, el registro nuevo es insertado en la primera posición (en preorden) en el árbol de la base de datos donde se pueda insertar un registro del mismo tipo que el nuevo.

Ejemplos:

Consideremos que queremos añadir una nueva alumna cuyo nombre es Delia Siordia con número de control 99310168 de la carrera de LI; entonces la inserción del nuevo registro sería de la siguiente manera:

```
Alumno.NombreA:="Delia Siordia";
Alumno.Control:="99310168";
Alumno.Esp:="ISC";
insert Alumno;
```

Consideremos que deseamos crear la alta de la materia de matemáticas 1 a la alumna con número de control 99310168.

```
Materia.NombreM:="Matemáticas 1";
Materia.Clave:="SCB9334";
Materia.Cred:=8;
insert Materia;
where Alumno.Control="99310168";
```

Modificación de registros existentes.

La instrucción para efectuar cambios a los registros es:

```
Replace
```

Esta instrucción no requiere los datos del registro a modificar como argumento, el registro que se afectará será aquel al que este apuntando el puntero de actualidad, que debe ser el registro que se desea modificar. Ejemplo:

Consideremos que deseamos reemplazar la carrera de la alumna con número de control 99310168.

```
Get hold first Alumno
where Alumno.Control="99310168";
Alumno.Esp:="LI";
replace;
```

Se agrega la palabra hold para que el sistema se entere que se va a modificar un registro.

Eliminación de un registro

Para eliminar un registro se debe apuntar al puntero de actualidad hacia ese registro, después se ejecuta la orden delete, al igual que en la orden replace, se debe poner la orden Hold. Ejemplo:

Consideremos que deseamos borrar al alumno con número de control 99310168.

```
Get hold first Alumno;
where Alumno.Control=99310168;
delete;
```

También se puede borrar un registro(raíz), lo cuál eliminaría todas sus derivaciones (hijos).

Ejemplo:

Consideremos que deseamos eliminar al alumno con número de control 99310168 y todas sus materias, entonces la instrucción quedaría:

```
get hold first Alumno
where Alumno.Control="99310168";
delete;
```

Registros virtuales.

Como se describió anteriormente, en las relaciones muchos a muchos se requería la repetición de datos para conservar la organización de la estructura del árbol de la base de datos. La repetición de la información genera 2 grandes problemas:

- ✓ La actualización puede generar inconsistencia de los datos.
- ✓ Se genera un desperdicio considerable de espacio.

Para solventar estos problemas se introdujo el concepto de registro virtual, el cual no contiene datos almacenados, si no un puntero lógico a un registro físico determinado.

Cuando se va a repetir un registro en varios árboles de la base de datos, se mantiene una sola copia de ese registro en uno de los árboles y empleamos en los otros registros la utilización de un registro virtual que contiene la dirección del registro físico original.

5. El modelo de red

- **Modelo conceptual (R)**

- **Estructura de almacenamiento (R)**

Una base de datos de red como su nombre lo indica, está formado por una colección de registros, los cuales están conectados entre sí por medio de enlaces. El registro es similar a una entidad como las empleadas en el modelo entidad-relación. Un registro es una colección de campos (atributos), cada uno de los cuales contiene solamente almacenado un solo valor, el enlace es la asociación entre dos registros exclusivamente, así que podemos verla como una relación estrictamente binaria. Una estructura de datos de red, llamada algunas veces estructura plex, abarca más que la estructura de árbol porque un nodo hijo en la estructura de red puede tener más de un parente. En otras palabras, la restricción de que en un árbol jerárquico cada hijo puede tener un solo parente, se hace menos severa. Así, la estructura de árbol se puede considerar como un caso especial de la estructura de red tal como lo muestra la siguiente figura. Para ilustrar la estructura de los registros en una base de datos de red, consideremos la base de datos alumno-materia, los registros en lenguaje Pascal entonces quedarían como:

```
type alumno= record
    NombreA:string[30];
    Control:string[8];
    Esp: string[3];
    end;
type materia = record
    Clave:string[7];
    NombreM:string[25];
    Cred:string[2];
    end;
```

Diagramas de estructura de datos.

Un diagrama de estructura de datos es un esquema que representa el diseño de una base de datos de red. Este modelo se basa en representaciones entre registros por medio de ligas, existen relaciones en las que participan solo dos entidades(binarias) y relaciones en las que participan más de dos entidades (generales) ya sea con o sin atributo descriptivo en la relación. La forma de diagramado consta de dos componentes básicos:
Celdas: representan a los campos del registro.

Líneas: representan a los enlaces entre los registros.

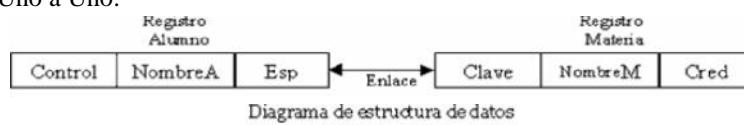
Un diagrama de estructura de datos de red, especifica la estructura lógica global de la base de datos; su representación gráfica se basa en el acomodo de los campos de un registro en un conjunto de celdas que se ligan con otro(s) registro(s), ejemplificaremos esto de la siguiente manera: Consideremos la relación alumno-cursa-materia donde la relación cursa no tiene atributos descriptivos :



Las estructuras de datos según la cardinalidad se representan en los siguientes casos:

Cuando el enlace no tiene atributos descriptivos

Caso 1. Cardinalidad Uno a Uno.



Caso 2. Cardinalidad Muchos a uno.

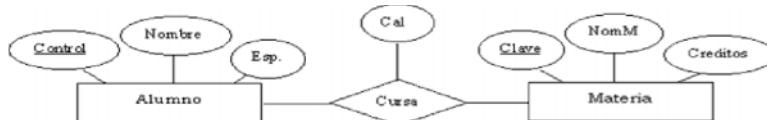


Caso 3. Cardinalidad Muchos a muchos.



Cuando el enlace tiene atributos descriptivos.

Consideremos que a la relación cursa le agregamos el atributo Cal (calificación), nuestro modelo E-R quedaría de la siguiente manera:



La forma de convertir a diagramas de estructura de datos consiste en realizar lo siguiente:

1. Realizar la representación de los campos del registro agrupándolos en sus celdas correspondientes.
2. Crear nuevo registro, denominado **Calif**, para este caso, con un solo campo, el de cal (calif).
3. Crear los enlaces indicando la cardinalidad de :

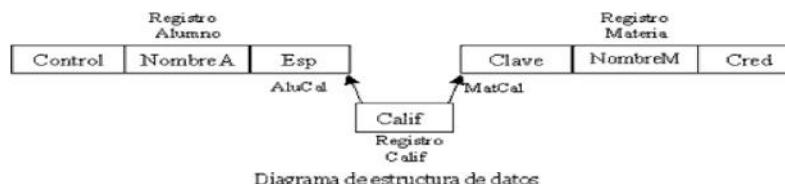
AluCal, del registro Calif al registro **Alumno**.

MatCal, del registro **Calif** al registro **Materia**.

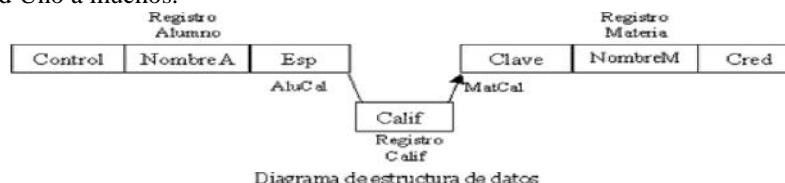
AluCal y MatCal son solo los nombres que emplearemos para identificar el enlace, pueden ser otros y no son empleados para otra cosa.

Los diagramas de estructuras de datos según la cardinalidad se transforman en:

Caso 1. Cardinalidad uno a uno.



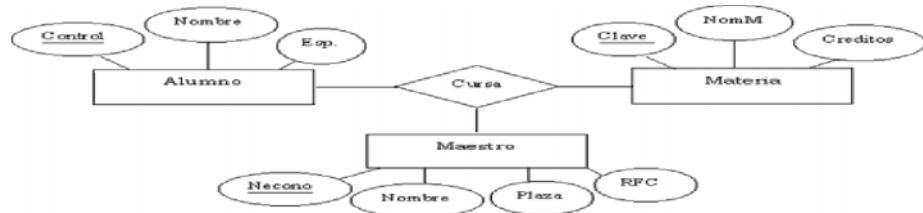
Caso 2. Cardinalidad Uno a muchos.



Caso 3. Cardinalidad Muchos a muchos.



Diagramas de estructura de datos cuando intervienen más de dos entidades y el enlace no tiene atributos descriptivos. Consideremos que a la relación alumno-cursa-materia le agregamos la entidad maestro, quien es el que imparte dicha materia. Nuestro diagrama E-R quedaría de la siguiente manera:



La transformación a diagramas de estructura de datos se realiza mediante los siguientes pasos:

- 1.Crear los respectivos registros para cada una de las entidades que intervienen en el modelo.
- 2.Crear un nuevo tipo de registro que llamaremos Renlace, que puede no tener campos o tener solo uno que contenga un identificador único, el identificador lo proporcionará el sistema y no lo utiliza directamente el programa de aplicación, a este registro se le denomina también como registro ficticio o de enlace o unión.

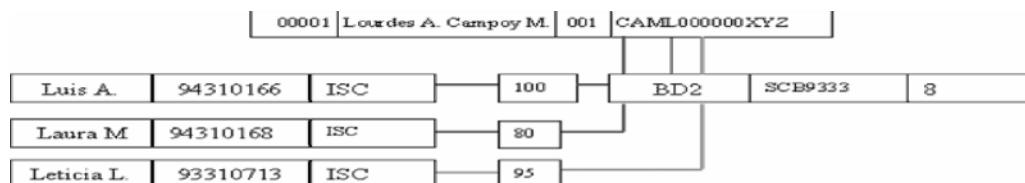
Siguiendo los pasos anteriores nuestra estructura finalmente es: (Considerando una relación con cardinalidad Uno a Uno)



Ahora si nuestro enlace tuviera atributos descriptivos, se crea el registro con los campos respectivos y se liga indicando el tipo de cardinalidad de que se trate. En este caso tomamos el ejemplo anterior con cardinalidad uno a uno y le agregamos a la relación el atributo calif. (calificación).



Considerando el anterior diagrama de estructura de datos, una instancia de este sería: La estructura quedaría:



Este diagrama nos indica que los alumnos Luis A. Laura M. y Leticia L. cursaron la materia Base de datos 2 con La maestra Ing. Lourdes A. Campoy M. obteniendo una calificación de 100,80,95 respectivamente. Este modelo fue desarrollado en 1971 por un grupo conocido como CODASYL: Conference on Data System Languages, Data Base Task Group, de ahí el nombre; este grupo es el que desarrolló los estándares para COBOL, el modelo CODASYL ha evolucionado durante los últimos años y existen diversos productos DBMS orientados a transacciones, sin embargo hoy día, estos productos están de salida, ya que este modelo es complejo y no cohesivo; los diseñadores y programadores deben de tener mucho cuidado al elaborar bases de datos y aplicaciones DBTG, además este modelo tiene mucho enfoque de COBOL, gran parte a las deficiencias detectadas en la actualidad se le atribuye a

que este modelo fue desarrollado muy pronto antes de que se establecieran correctamente los conceptos esenciales de la tecnología de bases de datos.

6. El modelo relacional

• Esquemas de base de datos relacional (R)

La ventaja del modelo relacional es que los datos se almacenan, al menos conceptualmente, de un modo en que los usuarios entienden con mayor facilidad. Los datos se almacenan como tablas y las relaciones entre las filas y las tablas son visibles en los datos. Este enfoque permite a los usuarios obtener información de la base de datos sin asistencia de sistemas profesionales de administración de información. Las características más importantes de los modelos relacionales son:

1. Es importante saber que las entradas en la tabla tienen un solo valor (son atómicos); no se admiten valores múltiples, por lo tanto la intersección de un renglón con una columna tiene un solo valor, nunca un conjunto de valores.
2. Todas las entradas de cualquier columna son de un solo tipo. Por ejemplo, una columna puede contener nombres de clientes, y en otra puede tener fechas de nacimiento. Cada columna posee un nombre único, el orden de las columnas no es de importancia para la tabla, las columnas de una tabla se conocen como atributos. Cada atributo tiene un dominio, que es una descripción física y lógica de valores permitidos.
3. No existen 2 filas en la tabla que sean idénticas.
4. La información en las bases de datos son representados como datos explícitos, no existen apuntadores o ligas entre las tablas.

En el enfoque relacional es sustancialmente distinto de otros enfoques en términos de sus estructuras lógicas y del modo de las operaciones de entrada/salida. En el enfoque relacional, los datos se organizan en tablas llamadas relaciones, cada una de las cuales se implanta como un archivo. En terminología relacional una fila en una relación representa un registro o una entidad; Cada columna en una relación representa un campo o un atributo. Así, una relación se compone de una colección de entidades(o registros) cuyos propietarios están descritos por cierto número de atributos predeterminados implantados como campos.

Estructura de las bases de datos relacionales

La arquitectura relacional se puede expresar en términos de tres niveles de abstracción: nivel interno, conceptual y de visión. La arquitectura relacional consta de los siguientes componentes:

- ✓ **Modelo relacional de datos:** En el nivel conceptual, el modelo relacional de datos está representado por una colección de relaciones almacenadas. Cada registro de tipo conceptual en un modelo relacional de datos se implanta como un archivo almacenado distinto.
- ✓ **Submodelo de datos:** Los esquemas externos de un sistema relacional se llaman submodelos relacionales de datos; cada uno consta de uno a más escenarios (vistas) para describir los datos requeridos por una aplicación dada. Un escenario puede incluir datos de una o más tablas de datos. Cada programa de aplicación está provisto de un buffer ("Área de trabajo de usuario") donde el DBMS puede depositar los datos recuperados de la base para su procesamiento, o puede guardar temporalmente sus salidas antes de que el DBMS las escriba en la base de datos.
- ✓ **Esquema de almacenamiento:** En el nivel interno, cada tabla base se implanta como un archivo almacenado. Para las recuperaciones sobre las claves principal o secundaria se pueden establecer uno o más índices para accesar un archivo almacenado.
- ✓ **Sublenguaje de datos:** Es un lenguaje de manejo de datos para el sistema relacional, el álgebra relacional y cálculo relacional, ambos lenguajes son "relacionalmente completos", esto es, cualquier relación que pueda derivarse de una o más tablas de datos, también se puede derivar con u solo comando del sublenguaje. Por tanto, el modo de operación de entrada/Salida en un sistema relacional se puede procesar en la forma: una tabla a la vez en lugar de: un registro a la vez; en otras palabras, se puede recuperar una tabla en vez de un solo registro con la ejecución de un comando del sublenguaje de datos.

Lenguajes de consulta formales.

Son los lenguajes en el que los usuarios solicitan información de la base de datos. Estos lenguajes son generalmente de más alto nivel que los lenguajes de programación. Los lenguajes de consulta pueden clasificarse como procedimentales y no procedimentales;

En el lenguaje del tipo procedimental el usuario da las instrucciones al sistema para que realice una secuencia de operaciones en la base de datos para calcular el resultado deseado.

En el lenguaje no procedimental, el usuario describe la información deseada sin dar un procedimiento específico para obtener dicha información.

El álgebra relacional es un lenguaje de consulta formal procedimental, el álgebra relacional define operadores que funcionan sobre las tablas (de una manera similar a los operadores +,-,etc. del álgebra común) para llegar al resultado deseado. El álgebra relacional es difícil de utilizar, debido en parte a que es procedimental, esto es, al utilizar el álgebra relacional no sólo debemos saber lo que queremos, también cómo obtenerlo. En el proceso de bases de datos comerciales el álgebra relacional se utiliza de manera poco frecuente. Aunque unos cuantos productos exitosos DBMS sí tienen opciones del álgebra relacional, éstas son poco utilizadas en vista de su complejidad. El álgebra relacional toma dos o más tablas como entrada produce una nueva tabla como resultado de la serie de operaciones. Las operaciones fundamentales en el álgebra relacional son seleccionar, proyectar, producto cartesiano, renombrar, unión y diferencia de conjuntos. Además de las operaciones fundamentales existen otras operaciones como son: intersección de conjuntos, producto natural, división y asignación.

Operaciones fundamentales

Las operaciones seleccionar, proyectar y renombrar, son denominadas operaciones unitarias ya que operan sobre una tabla. Las otras operaciones operan sobre pares de relaciones y, por tanto se llaman operaciones binarias.

✓ *La operación seleccionar.*

Esta operación selecciona tuplas (filas) que satisfacen una instrucción (condición) dada de una tabla. Se representa por medio de paréntesis.

(nombre_tabla WHERE condición);

La oración de la instrucción después de la cláusula WHERE puede incluir condiciones de igualdad como =,=,<,>,>=,<=,además que se puede hacer una oración más compleja usando los conectores y (^) y o (v).

✓ *La operación Proyectar.*

Consiste en identificar las columnas (atributos en el modelo E-R) que nos interesa conocer. Se representa por medio de corchetes. Si este se omite indicara que se desea obtener todas las columnas de la tabla en cuestión.

(nombre_tabla WHERE condición) [Nombre_atributo];

✓ *La operación Producto cartesiano.*

Consiste en multiplicar todas las tuplas entre tablas, obteniendo como resultado una tabla que contiene todas las columnas de ambas tablas. Se especifica con la orden TIMES.

Nombre_tabla TIMES Nombre_tabla;

✓ *La operación Join.*

Consiste en obtener el producto (multiplicación) de todas las tuplas de una tabla con las de la otra, para posteriormente evaluar aquellas cuyo campo en común sea igual generando como resultado una nueva tabla que tiene como tuplas (renglones) que cumplen con la condición establecida. Se representa con la orden JOIN. La orden Join es colocada entre las dos tablas a multiplicar después de que la primera especifica la operación de selección y proyección.

(Tabla)[atributo] JOIN (Tabla)[Atributo];

✓ *La operación Divide.*

Toma dos relaciones, una binaria y la otra unaria, construye una relación formada por todos los valores de un atributo de la relación binaria que concuerdan (en el otro atributo) con todos los valores de la relación unaria. Se representa con la orden DIVIDE BY.

NomTablaBin DIVIDE BY NomTablaUna

✓ *La operación Diferencia.*

Construye una relación formada por todas las tuplas (filas) de la primera relación que no aparezcan en la segunda de las dos relaciones especificadas. Se representa con la orden MINUS.

Nom_tablaA MINUS NomTablaB;

✓ *La operación Unión.*

Construye una relación formada por todas las tuplas de la primera relación y todas las tuplas de la segunda relación. El requisito es que ambas relaciones sean del mismo tipo.

Nom_TablaA UNION Nom_tablaB

✓ *La operación intersección.*

Construye una nueva tabla compuesta por todas las tuplas que están en la primera y segunda tabla.

Nom_TablaA INTERSEC Nom_tablaB

Ejemplos:

Para exemplificar las notaciones anteriores consideremos el ejemplo

ALUMNO - cursa - MATERIA, que tienen los siguientes atributos:

ALUMNO	- cursa	- MATERIA
NControl	NControl	Clave
NombreA	Clave	NombreM
Especialidad	Calif	Créditos
Dirección		

Representando en tablas a los atributos quedarían de la siguiente forma:

Tabla alumno:

NControl	NombreA	Especialidad	Dirección

Tabla cursa:

NControl	Clave	Calif

Tabla materia:

Clave	NombreM	Créditos

1.- Obtener el nombre de todos los alumnos que están inscritos en la Institución.

(Alumno) [NombreA];

2.- Obtener el nombre de los alumnos que cursan la materia Base de datos 1 cuya clave es SCB9333

(Alumno) JOIN (Cursa where Clave='SCB9333') [NombreA];

3.- Obtener los nombres de los alumnos de la especialidad de Ing. Sistemas que cursan la materia Base de datos 2.

((Alumno)[especialidad,NombreA,NControl]

JOIN (Cursa) where especialidad = 'ISC')[Clave,NombreA])

JOIN (Materia where NombreM='BD2')[NombreA];

En el álgebra relacional no solo debemos saber lo que queremos si no también como obtenerlo, al realizar las consultas debemos especificar el nombre de la tabla a utilizar en caso de que deseemos realizar una operación con un atributo que las otras tablas no tienen debemos "arrastrar" dicho atributo para poder utilizarlo, como lo es en el caso anterior, en donde requerimos el nombre del alumno que solamente lo tiene la tabla alumno, pro también deseamos que se cumpla la condición NombreM=BD2, como no podemos relacionar directamente a ambas tablas empleamos la tabla cursa de donde obtenemos la clave de las materias y mantenemos el nombre del alumno (NombreA) finalmente con la orden JOIN se combinan las tablas por el campo común que tienen que es clave así que obtenemos una tabla con todas las materias que cursan los alumnos de ISC, de donde seleccionamos solo aquella que se llame BD2 con la orden Join obtenemos esta nueva tabla de donde por último proyectamos el atributo NombreA que hemos venido "arrastrando".

Lenguajes de consultas comerciales

Un lenguaje de consulta comercial proporciona una interfaz más amigable al usuario. Un ejemplo de este tipo de lenguaje es el SQL, (Structured Query Language, Lenguaje de Consulta Estructurado). Las partes más importantes del SQL son:

✓ DDL: Lenguaje de definición de datos (que nos permite crear las estructuras)

- ✓ DML: Lenguaje de manipulación de datos (que nos permite tener acceso a las estructuras para suprimir, modificar e insertar)

La estructura básica de una expresión en SQL contiene 3 partes, Select, From y Where.

- ✓ La cláusula Select se usa para listar los atributos que se desean en el resultado de una consulta.
- ✓ From, Lista las relaciones que se van a examinar en la evaluación de la expresión.
- ✓ Where, es la definición de las condiciones a las que puede estar sujeta una consulta.

La consulta típica de SQL tiene la siguiente forma:

```
Select A1,A2,A3...An
From r1,r2,r3...rm
Where Condición(es)
```

Donde:

A1,A2,A3...An: Representan a cada atributo(s) o campos de las tablas de la base de datos relacional.

R1,r2,r3...rm: Representan a la(s) tabla(s) involucradas en la consulta.

Condición: Es el enunciado que rige el resultado de la consulta.

Si se omite la cláusula Where, la condición es considerada como verdadera, la lista de atributos (A1,A2..An) puede sustituirse por un asterisco (*), para seleccionar todos los atributos de todas las tablas que aparecen en la cláusula From.

Funcionamiento del SQL.

El SQL forma el producto cartesiano de las tablas involucradas en la cláusula From, cumpliendo con la condición establecida en la orden Where y después proyecta el resultado con la orden select. Para nuestros ejemplos consideremos una tabla llamada CURSO, que contiene los siguientes campos:

Nombre del campo	Descripción
NumC	Número del curso, único para identificar cada curso
NombreC	Nombre del curso, también es único
DescC	Descripción del curso
Creditos	Créditos, número de estos que gana al estudiante al cursarlo
Costo	Costo del curso.
Deptlo	Departamento académico que ofrece el curso.

Datos contenidos en la tabla CURSO

NumC	NombreC	DescC	Creditos	Costo	Deptlo
A01	Liderazgo	Para público General	10	100.00	Admón.
S01	Introducción a la inteligencia artificial	Para ISC y LI	10	90.00	Sistemas.
C01	Construcción de torres	Para IC y Arquitectura	8	0.00	Ciencias
B01	Situación actual y perspectivas de la alimentación y la nutrición	Para IB	8	80.00	Bioquímica
E01	Historia presente y futuro de la energía solar	IE e II	10	100.00	Electromecánica.
S02	Tecnología OLAP	Para ISC y LI	8	100.00	Sistemas
C02	Tecnología del concreto y de las Estructuras	Para IC	10	100.00	Ciencias
B02	Metabolismo de lípidos en el camarón	Para IB	10	0.00	Bioquímica
E02	Los sistemas eléctricos de potencia	Para IE	10	100.00	Electromecánica
S03	Estructura de datos	Para ISC y LI	8	0.00	Sistemas
A01	Diseño bioclimático	Para Arquitectura	10	0.00	Arquitectura
C03	Matemáticas discretas	General	8	0.00	Ciencias
S04	Circuitos digitales	Para ISC	10	0.00	Sistemas
S05	Arquitectura de Computadoras	Para ISC	10	50.00	Sistemas

I01	Base de Datos Relacionales	Para ISC y LI	10	150.00	Informática
-----	----------------------------	---------------	----	--------	-------------

Ejemplos de consultas:

OBTENCIÓN DE UNA TABLA ENTERA

Obtener toda la información disponible sobre un curso donde Costo sea 0.

```
SELECT *
FROM CURSO
WHERE Costo=0.00
```

Resultado de la consulta anterior.

NumC	NombreC	DescC	Creditos	Costo	DeptO
C01	Construcción de torres	Para IC y Arquitectura	8	0.00	Ciencias
B02	Metabolismo de lípidos en el camarón	Para IB	10	0.00	Bioquímica
S03	Estructura de datos	Para ISC y LI	8	0.00	Sistemas
A01	Diseño bioclimático	Para Arquitectura	10	0.00	Arquitectura
C03	Matemáticas discretas	General	8	0.00	Ciencias

Colocamos un * debido a que no nos limitan la información de la tabla, es decir nos piden que mostremos todos los datos atributo de la tabla CURSO. Como la única condición en la sentencia WHERE es que la tarifa del curso sea igual a 0, esta consulta regresa todas las tuplas donde se encuentre que Costo = 0.00. Debido a que Costo es un campo numérico, la condición solo puede comparar con campos del mismo tipo. Para representar valores negativos se antepone a la izquierda el signo (-), en este ejemplo se considera solo el signo (=) para establecer la condición, sin embargo otros operadores que se pueden utilizar son:

- ✓ Menor que <
- ✓ Mayor que >
- ✓ Menor o igual que <=
- ✓ Mayor o igual que >=
- ✓ Diferente <>

Además de los operadores booleanos AND, NOT, OR.

Cabe señalar que en la sentencia Where cuando se requiere establecer condiciones con cadenas, estas son delimitadas por apóstrofes (''). Las expresiones de cadenas son comparadas carácter por carácter, dos cadenas son iguales solo si coinciden todos los caracteres de las mismas.

Ejemplos de consultas con cadenas:

Obtener toda la información sobre cualquier curso que ofrezca el departamento de Ciencias.

```
SELECT *
FROM CURSO
WHERE Depto = 'Ciencias';
```

Resultado de la consulta.

NumC	NombreC	DescC	Creditos	Costo	DeptO
C01	Construcción de torres	Para IC y Arquitectura	8	0.00	Ciencias
C02	Tecnología del concreto y de las Estructuras	Para IC	10	100.00	Ciencias
S04	Circuitos digitales	Para ISC	10	0.00	Sistemas

VISUALIZACIÓN DE COLUMNAS ESPECIFICADAS.

En los ejemplos anteriores obtenímos toda la tabla completa, ahora veremos como mostrar solo algunos atributos específicos de una tabla.

Obtener los valores NumC,NombreC y Depto, en este orden de toda la tabla curso.

```
SELECT NumC, NombreC, Depto
FROM CURSO;
```

Resultado de la consulta:

NumC	NombreC	DeptO
------	---------	-------

A01	Liderazgo	Admón.
S01	Introducción a la inteligencia artificial	Sistemas.
C01	Construcción de torres	Ciencias
B01	Situación actual y perspectivas de la alimentación y la nutrición	Bioquímica
E01	Historia presente y futuro de la energía solar	Electromecánica.
S02	Tecnología OLAP	Sistemas
C02	Tecnología del concreto y de las Estructuras	Ciencias
B02	Metabolismo de lípidos en el camarón	Bioquímica
E02	Los sistemas eléctricos de potencia	Electromecánica
S03	Estructura de datos	Sistemas
A01	Diseño bioclimático	Arquitectura
C03	Matemáticas discretas	Ciencias
S04	Circuitos digitales	Sistemas
S05	Arquitectura de Computadoras	Sistemas
I01	Base de Datos Relacionales	Informática

Observamos que en este caso no se tiene la sentencia Where, no existe condición, por lo tanto, todas las filas de la tabla CURSO se recuperan, pero solo se visualizan las tres columnas especificadas. Así mismo, empleamos la (,) para separar los campos que deseamos visualizar.

VISUALIZACIÓN DE UN SUBCONJUNTO DE FILAS Y COLUMNAS

Seleccionar los valores NumC, Depto y Costo para todos los cursos que tengan un Costo inferior a \$100

```
SELECT NumC, Depto, Costo
FROM CURSO
WHERE Costo < 100.00
```

Como resultado de esta consulta se obtendrán todas aquellas tuplas que tengan un costo en CTARIFA menor que 100, y se visualizaran solo los campos de NumC, Depto,Costo. Podemos observar que este ejemplo cubre el formato general de una consulta SQL.

La palabra clave DISTINCT

DISTINCT, es una palabra reservada que elimina las filas que duplicadas en el resultado de una consulta.

Visualizar todos los departamentos académicos que ofrezcan cursos, rechazando los valores duplicados.

```
SELECT DISTINCT Depto
FROM CURSO;
```

Resultado de la consulta

Deptó
Administración
Sistemas
Ciencias
Bioquímica
electromecánica
Arquitectura
Informática

La palabra DISTINCT va estrictamente después de la palabra SELECT. De no haberse utilizado la palabra DISTINCT, el resultado hubiera mostrado todas las tuplas del atributo Depto que se encontraran, es decir, se hubiera visualizado la columna de Depto completamente.

EMPLEO DE LOS CONECTORES BOOLEANOS (AND, OR, NOT)

Para emplear las condiciones múltiples dentro de la sentencia WHERE, utilizamos los conectores lógicos.

El conector AND.

Este conector pide al sistema que seleccione una sola columna únicamente si ambas condiciones se cumplen. Obtener toda la información sobre todos los cursos que ofrece el departamento Sistemas que tengan una tarifa igual a 0.

```
SELECT *
FROM CURSO
WHERE Depto='Sistemas' AND Costo=0.00;
```

El resultado de esta consulta sería todas aquellas tuplas que cumplan exactamente con las dos condiciones establecidas.

El conector OR.

Este conector al igual que el AND permite conectar condiciones múltiples en la sentencia WHERE, a diferencia del conector AND, el OR permite la selección de filas que cumplan con una sola de las condiciones establecidas a través de este conector. Obtener toda la información existente sobre cualquier curso ofrecido por los departamentos Arquitectura o Bioquímica.

```
SELECT *
FROM CURSO
WHERE Depto = 'Arquitectura' OR Depto= 'Bioquímica';
```

El resultado de esta consulta será la de visualizar todas aquellas tuplas donde se cumpla cualquiera de las 2 condiciones, es decir mostrara todas las tuplas que tengan en el atributo Depto=Arquitectura o Bioquímica.

El conector NOT

Este nos permite marcar aquellas tuplas que por alguna razón no deseamos visualizar. Obtener el nombre del curso y del departamento de todos los cursos que no sean ofrecidos por el departamento Sistemas.

```
SELECT NombreC, Depto
FROM CURSO
WHERE NOT (Deptode='Sistemas');
```

JERARQUÍA DE OPERADORES BOOLEANOS.

En orden descendente (de mayor a menor prioridad)

- ✓ NOT
- ✓ AND
- ✓ OR

Existen dos formas para realizar consultas: Join de Querys y Subquerys. Cuando en la sentencia From colocamos los nombres de las tablas separados por comas se dice que efectuamos una consulta de la forma Join de Querys, en este caso se requiere anteponer el nombre de la tabla y un punto al nombre del atributo. En el Join de Querys el resultado que se produce con las tablas que intervienen en la consulta es la concatenación de las tablas, en donde los valores de una columna de la primera tabla coinciden con los valores de una segunda tabla, la tabla de resultado tiene una fila por cada valor coincidente que resulte de las dos tablas originales. Para ejemplificar esto, consideremos 2 tablas: Tabla1 y Tabla2, entonces:

C1	C2	C3		CA	CB
A	AAA	10		35	R
B	BBB	45		10	S
C	CCC	55		65	T
D	DDD	20		20	U
E	EEE	20		90	V
F	FFF	90		90	W
G	GGG	15		75	X
H	HHH	90		90	Y
				35	Z

Resultado de la operación Join:

C1	C2	C3	CA	CB
A	AAA	10	10	S
D	DDD	20	20	U
E	EEE	20	20	U
F	FFF	90	90	V
F	FFF	90	90	W
F	FFF	90	90	Y
H	HHH	90	90	V
H	HHH	90	90	W
H	HHH	90	90	Y

Como podemos observar, la comparación se efectuó por las columnas C3 y CA, que son donde se encontraron valores iguales, el resultado muestra una tupla por cada coincidencia encontrada. Cuando las consultas se anidan se conoce como Subquerys o subconsultas. Este tipo de consulta obtiene resultados parciales reduciendo el espacio requerido para realizar una consulta. Nota: Todas las consultas que se resuelven con subquerys pueden resolverse con Join de Querys, pero no todas las consultas hechas con Join de Querys pueden resolverse utilizando Subquerys. Para exemplificar lo anterior consideremos el ejemplo ALUMNO-cursa-MATERIA, que tienen los siguientes atributos:

NControl	NControl	Clave
NombreA	Clave	NombreM
Especialidad	Calif	Creditos
Dirección		

Representando en tablas a los atributos quedarían de la siguiente forma:

Tabla alumno:

NControl	NombreA	Especialidad	Dirección

Tabla cursa:

NControl	Clave	Calif

Tabla materia:

Clave	NombreM	Creditos

Obtener el nombre de la materia que cursa el alumno con número de control 97310211 con créditos igual a ocho.

```

SELECT NombreA
FROM Materia
WHERE creditos='8' and clave in(SELECT clave
                                FROM cursa
                                WHERE NControl='97310211');
    
```

Obtener el número de control del alumno que tenga alguna calificación igual a 100

```

SELECT DISTINCT(NControl)
FROM Cursa
WHERE Calif='100';
    
```

Obtener el nombre de las materias que cursa el alumno Salvador Chávez.

```

SELECT NombreM
FROM Materia
WHERE Clave in (SELECT DISTINCT(Clave)
                  FROM Cursa
                  WHERE NControl in (SELECT NControl
                                      FROM capedsc@hotmai...
    
```

```
FROM Alumno  
WHERE NombreA='Salvador Chávez');
```

FUNCIONES AVANZADAS APLICABLES A CONSULTAS

Existen funciones que permiten la agilización de consultas similares a una hoja de cálculo, ya que trabajan en base a renglones y columnas.

COUNT (): Cuenta el número de tuplas en la columna establecida

MIN (): Localiza el valor mínimo de la columna establecida

MAX (): Localiza el valor máximo de la columna establecida.

AVG (): Obtiene el promedio de valores de la columna establecida

SUM (): Obtiene el valor total que implican los valores obtenidos en la columna establecida.

Ejemplos:

Obtener el número de alumnos que existen en la carrera de Ingeniería en Sistemas Computacionales.

```
SELECT Count(*)  
FROM Alumno  
WHERE especialidad='ISC';
```

Obtener la máxima calificación que ha obtenido J.M. Cadena.

```
SELECT Max(Calif)  
FROM Cursa  
WHERE NControl IN (SELECT NControl  
                    FROM Alumno  
                    WHERE NombreA= 'J.M. Cadena '');
```

Obtener el promedio de calificaciones de Salvador Chávez.

```
SELECT Avg(Calif)  
FROM Cursa  
WHERE NCotrol IN (SELECT NControl  
                   FROM Alumno  
                   WHERE NombreA='Salvador Chávez');
```

Obtener la suma total de las calificaciones obtenidas por Daniel Colín.

```
SELECT Sum(Calif)  
FROM Cursa  
WHERE NControl IN (SELECT NControl  
                   FROM Alumno  
                   WHERE NombreA='Daniel Colín');
```

Hasta aquí hemos visto el manejo sencillo de realizar consultas con SQL, hay que destacar que en la realización de consultas anidadas se tiene que poner cuidando a la prioridad de los operadores, teniendo cuidado también al momento de agrupar los paréntesis que involucran las condiciones con los operadores.

Modificación de la Base de datos

Como se mencionó al inicio de este apartado del SQL, esté cuenta con módulos DDL, para la definición de datos que nos permite crear o modificar la estructura de las tablas. Las instrucciones para realizar estas operaciones son:

- ✓ CREATE TABLE: Nos permite crear una tabla de datos vacía.
- ✓ INSERT: Permite almacenar registros en una tabla creada.
- ✓ UPDATE: Permite modificar datos de registros almacenados en la tabla.
- ✓ DELETE: Borra un registro entero o grupo de registros de una tabla.
- ✓ CREATE INDEX: Crea un índice que nos puede auxiliar para las consultas.
- ✓ DROP TABLE: Permite borrar una tabla.
- ✓ DROP INDEX: Borra el índice indicado.

Para ejemplificar las instrucciones anteriores consideremos el ejemplo

ALUMNO - cursa - MATERIA, que tienen los siguientes atributos:
NControl NControl Clave

NombreA	Clave	NombreM
Especialidad	Calif	Creditos
Dirección		

* Estructura de la sentencia CREATE TABLE.

CREATE TABLE <Nombre de la tabla>

```
(  
    Atributo1: tipo de dato longitud ,  
    Atributo2: tipo de dato longitud ,  
    Atributo3: tipo de dato longitud ,  
    :  
    :  
    Atributon: tipo de dato longitud ,  
    PRIMARY KEY (Opcional) );
```

Los campos pueden definirse como NOT NULL de manera opcional excepto en la llave primaria para lo cual es obligatorio. Además al definir la llave primaria se genera automáticamente un índice con respecto al campo llave; para definir la llave la denotamos dentro de los paréntesis de PRIMARY KEY.

Ejemplo:

Crear la tabla alumno con los atributos antes descritos, tomando como llave el numero de control.

CREATE TABLE Alumno

```
(  
    NControl char(8) NOT NULL,  
    NombreA char(20),  
    Especialidad char(3),  
    Dirección char(30),  
    PRIMARY KEY (NControl) );
```

Tabla Alumno:

NControl NombreA Especialidad Dirección

Pueden existir más de una llave primaria, esto es si se requiere, se crearán tantos índices como llaves primarias se establezcan. Pueden existir tantos campos Not Null (No nulos) como se requieran; En si estructurar la creación de una tabla es siempre parecida al ejemplo anterior.

* Estructura de la sentencia INSERT

```
INSERT  
INTO Nombre de la tabla a la que se le va a insertar el registro  
VALUES (Conjunto de valores del registro ) ;
```

Ejemplo:

Insertar en la tabla Alumno, antes creada los datos del alumno Daniel colín, con numero de control 95310518 de la especialidad de Ingeniería civil, con domicilio Abasolo Norte #45.

```
INSERT  
INTO Alumno  
VALUES("95310518","Daniel Colín","IC","Abasolo Norte #45") ;
```

Nótese que la inserción de los datos se realiza conforme la estructura que se implantó en la tabla, es decir en el orden en que se creo dicha tabla. En caso de querer omitir un dato que no sean no nulos solamente se ponen las comillas indicando el vacío de la cadena.

* Estructura de la Sentencia CREATE INDEX

```
CREATE INDEX Nombre que se le asignara al índice.  
ON Nombre de la taba a la cual se le creara el índice (Campo(s) por el cual se creara el índice);
```

Ejemplo:

Crear un índice de la tabla Alumno por el campo Especialidad.

```
CREATE INDEX Indice1  
ON Alumno(Especialidad);
```

Este índice contendrá a todos los alumnos ordenados por el campo especialidad.

```
CREATE INDEX UNIQUE INDEX Indice2  
ON Alumno (Especialidad);
```

En la creación de este índice utilizamos la sentencia UNIQUE, es un indicador para permitir que se cree un índice único por especialidad, esta sentencia siempre se coloca antes de CREATE INDEX. En este ejemplo se creara un índice que contenga un alumno por especialidad existente.

* Estructura de la sentencia UPDATE
UPDATE Nombre de la tabla en donde se modificaran los datos.
SET Valores
WHERE (Condición);

Ejemplo:

Modificar el número de control del registro de Daniel Colín de la Tabla alumno por el número 96310518.
UPDATE Alumno
SET NControl '96310518'
WHERE NombreA='Daniel Colín';

* Estructura de la sentencia DROP TABLE
DROP TABLE Nombre de la tabla a borrar ;

Ejemplo:

Borrar la tabla Alumno creada anteriormente.
DROP TABLE Alumno;

* Estructura de la sentencia DROP INDEX
DROP INDEX Nombre del índice a borrar;

Ejemplo:

Borrar el índice Indice1 creado anteriormente.
DROP INDEX Indice1;

* Estructura de la sentencia DELETE
DELETE
FROM Nombre de la tabla
WHERE Condición;

Ejemplos:

- Borrar el registro cuyo número de control es 95310386.
DELETE
FROM Alumno
WHERE Control='95310386';

- Borrar todos los registros de la tabla alumno.

```
DELETE  
FROM Alumno;
```

En el primer ejemplo, se borrara todo el registro(todos los datos), del alumno con número de control = 95310386. En el segundo ejemplo se borraran todos los registros de la tabla alumno, pero sin borrar la estructura de la tabla, ya que la orden Delete solo borra registros, la sentencia Drop Table es la que borra toda la estructura de la tabla junto con los registros de la misma.

Vistas.

Una vista se define en SQL usando la orden CREATE VIEW. Para definir una vista debemos dar a la vista un nombre y declarar la consulta que calcula la vista. Una vez que establecemos una vista, podemos ejecutar una sentencia SELECT que refiera a esa vista. El sistema asociará la vista SQL con una tabla base y extraerá y visualizará, entonces, los datos de la tabla base. Esto significa que una vista no contiene datos duplicados de una tabla base. No tiene absolutamente ningún dato, puesto que no es una tabla real, todo el proceso se realiza con los datos almacenados en la tabla base. Es decir se percibe como una tabla virtual. Las órdenes que se utilizan para la manipulación de vistas son:

- ✓ CREATE VIEW: Crea una tabla virtual.
- ✓ DROP VIEW: Elimina una vista creada anteriormente.

Estructura de la sentencia CREATE VIEW.

CREATE VIEW Nombre de la vista AS (Expresión de consulta);

Para nuestros ejemplos consideremos de nuevo la tabla llamada CURSO, que contiene los siguientes campos:

Nombre del campo	Descripción
NumC	Número del curso, único para identificar cada curso
NombreC	Nombre del curso, también es único
DescC	Descripción del curso
Creditos	Créditos, número de estos que gana al estudiante al cursarlo
Costo	Costo del curso.
Depto	Departamento académico que ofrece el curso.

Que contiene los siguientes datos:

NumC	NombreC	DescC	Créditos	Costo	Depto
A01	Liderazgo	Para público General	10	100.00	Admón.
S01	Introducción a la inteligencia artificial	Para ISC y LI	10	90.00	Sistemas.
C01	Construcción de torres	Para IC y Arquitectura	8	0.00	Ciencias
B01	Situación actual y perspectivas de la alimentación y la nutrición	Para IB	8	80.00	Bioquímica
E01	Historia presente y futuro de la energía solar	IE e II	10	100.00	Electromecánica.
S02	Tecnología OLAP	Para ISC y LI	8	100.00	Sistemas
C02	Tecnología del concreto y de las Estructuras	Para IC	10	100.00	Ciencias
B02	Metabolismo de lípidos en el camarón	Para IB	10	0.00	Bioquímica
E02	Los sistemas eléctricos de potencia	Para IE	10	100.00	Electromecánica
S03	Estructura de datos	Para ISC y LI	8	0.00	Sistemas
A01	Diseño bioclimático	Para Arquitectura	10	0.00	Arquitectura
C03	Matemáticas discretas	General	8	0.00	Ciencias
S04	Circuitos digitales	Para ISC	10	0.00	Sistemas
S05	Arquitectura de Computadoras	Para ISC	10	50.00	Sistemas
I01	Base de Datos Relacionales	Para ISC y LI	10	150.00	Informática

Ejemplos:

* Crear una vista (tabla virtual), denominada CursosS, que contenga las filas solo correspondientes a cursos ofrecidos por el departamento Sistemas. La vista deberá contener todas las columnas de la tabla CURSO, con la excepción de la columna Depto, la secuencia, de izquierda a derecha de las columnas, deberá ser: NombreC, NumC, Creditos, Costo Y DescC.

```
CREATE VIEW CursosS AS
    SELECT NombreC, NumC, Creditos, Costo, DescC
    FROM CURSO
    WHERE DescC='Sistemas';
```

Observemos que después del nombre de la vista ponemos la sentencia AS, esto para definir la estructura de la vista, la estructura en si de la vista esta formada por la consulta anteriormente vista utilizando la orden SELECT.

* Crear una vista denominada CursosCaros, correspondientes a las filas de la tabla CURSO, en donde la tarifa excede de \$150, las columnas de la vista deberán tener los nombres ClaveCurso, NombreCurso y CostoCaro.

```
CREATE VIEW CursosSCaros(ClaveCurso,NombreCurso,CostoCaro) As  
    SELECT NumC,NOMBREC, Costo  
    FROM Curso  
    WHERE Costo > 150;
```

Observamos que después del nombre de la vista CursosCaros ponemos los nombres que se nos pidieron tuvieran los campos de la vista(ClaveCurso,...), después se realiza la consulta correspondiente para generar el resultado deseado.

Visualizar las vistas

Creamos una tabla virtual que contiene los datos de las consultas que deseamos, ahora nos falta visualizar estos datos, para ello utilizamos la sentencia SELECT y realizamos la consulta:

```
SELECT * FROM CursosCaros;
```

De esta consulta podemos observar que mostramos todos los campos que la vista contiene, aunque podemos visualizar solo alguno de ellos, también observamos que sustituimos el nombre de la vista por el de la tabla junto a la sentencia FROM, esto es por que una vista es una tabla virtual, pero guarda los datos como cualquier tabla normal.

Eliminar una vista

Como si fuera una tabla normal, las vistas también pueden borrarse, para ello utilizamos la sentencia DROP VIEW.
Estructura de la sentencia DROP VIEW.

```
DROP VIEW Nombre de la vista a borrar;
```

Ejemplo: Borrar la vista CursosCaros creada anteriormente.

```
DROP VIEW CursosCaros;
```

7. Diseño relacional

• Proceso de normalización (R)

Peligros en el diseño de bases de datos relacionales.

Uno de los retos en el diseño de la base de datos es el de obtener una estructura estable y lógica tal que:

- ✓ El sistema de base de datos no sufra de anomalías de almacenamiento.
- ✓ El modelo lógico pueda modificarse fácilmente para admitir nuevos requerimientos.

Una base de datos implantada sobre un modelo bien diseñado tiene mayor esperanza de vida aun en un ambiente dinámico, que una base de datos con un diseño pobre. En promedio, una base de datos experimenta una reorganización general cada seis años, dependiendo de lo dinámico de los requerimientos de los usuarios. Una base de datos bien diseñada tendrá un buen desempeño aunque aumente su tamaño, y será lo suficientemente flexible para incorporar nuevos requerimientos o características adicionales. Existen diversos riesgos en el diseño de las bases de datos relacionales que afecten la funcionalidad de la misma, los riesgos generalmente son la redundancia de información y la inconsistencia de datos. La normalización es el proceso de simplificar la relación entre los campos de un registro. Por medio de la normalización un conjunto de datos en un registro se reemplaza por varios registros que son más simples y predecibles y, por lo tanto, más manejables. La normalización se lleva a cabo por cuatro razones:

- ✓ Estructurar los datos de forma que se puedan representar las relaciones pertinentes entre los datos.
- ✓ Permitir la recuperación sencilla de los datos en respuesta a las solicitudes de consultas y reportes.
- ✓ Simplificar el mantenimiento de los datos actualizándolos, insertándolos y borrándolos.
- ✓ Reducir la necesidad de reestructurar o reorganizar los datos cuando surjan nuevas aplicaciones.

En términos más sencillos la normalización trata de simplificar el diseño de una base de datos, esto a través de la búsqueda de la mejor estructuración que pueda utilizarse con las entidades involucradas en ella.

Pasos de la normalización:

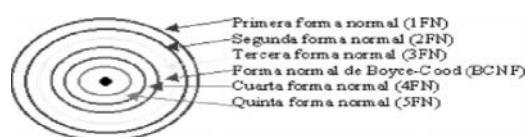
1. Descomponer todos los grupos de datos en registros bidimensionales.
2. Eliminar todas las relaciones en la que los datos no dependan completamente de la llave primaria del registro.
3. Eliminar todas las relaciones que contengan dependencias transitivas.

La teoría de normalización tiene como fundamento el concepto de formas normales; se dice que una relación está en una determinada forma normal si satisface un conjunto de restricciones.

Primera y segunda forma normal.

Formas normales. Son las técnicas para prevenir las anomalías en las tablas. Dependiendo de su estructura, una tabla puede estar en primera forma normal, segunda forma normal o en cualquier otra.

Relación entre las formas normales:



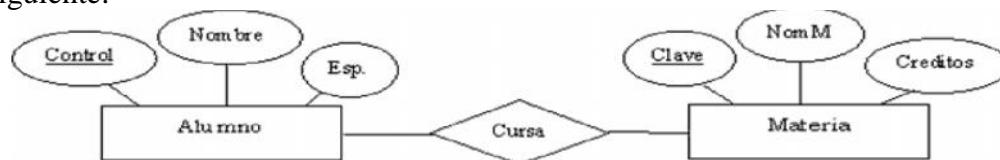
Primera forma normal.

Definición formal: Una relación R se encuentra en 1FN si y solo sí por cada renglón columna contiene valores atómicos.

Abreviada como 1FN, se considera que una relación se encuentra en la primera forma normal cuando cumple lo siguiente:

- ✓ Las celdas de las tablas poseen valores simples y no se permiten grupos ni arreglos repetidos como valores, es decir, contienen un solo valor por cada celda.
- ✓ Todos los ingresos en cualquier columna(atributo) deben ser del mismo tipo.
- ✓ Cada columna debe tener un nombre único, el orden de las columnas en la tabla no es importante.
- ✓ Dos filas o renglones de una misma tabla no deben ser idénticas, aunque el orden de las filas no es importante.

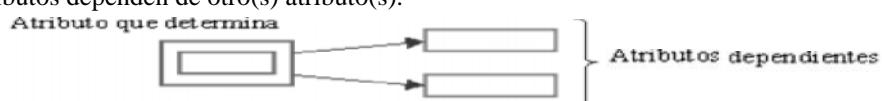
Por lo general la mayoría de las relaciones cumplen con estas características, así que podemos decir que la mayoría de las relaciones se encuentran en la primera forma normal. Para ejemplificar como se representan gráficamente las relaciones en primera forma normal consideremos la relación alumno cursa materia cuyo diagrama E-R es el siguiente:



Como esta relación maneja valores atómicos, es decir un solo valor por cada uno de los campos que conforman a los atributos de las entidades, ya se encuentra en primera forma normal, gráficamente así representamos a las relaciones en 1FN.

Segunda forma normal.

Para definir formalmente la segunda forma normal requerimos saber que es una dependencia funcional: Consiste en edificar que atributos dependen de otro(s) atributo(s).



Definición formal: Una relación R está en 2FN si y solo si está en 1FN y los atributos no primos dependen funcionalmente de la llave primaria.

Una relación se encuentra en segunda forma normal, cuando cumple con las reglas de la primera forma normal y todos sus atributos que no son claves (llaves) dependen por completo de la clave . De acuerdo con esta definición, cada tabla que tiene un atributo único como clave, esta en segunda forma normal. La segunda forma normal se representa por dependencias funcionales como:



Nótese que las llaves primarias están representadas con doble cuadro, las flechas nos indican que de estos atributos se puede referenciar a los otros atributos que dependen funcionalmente de la llave primaria.

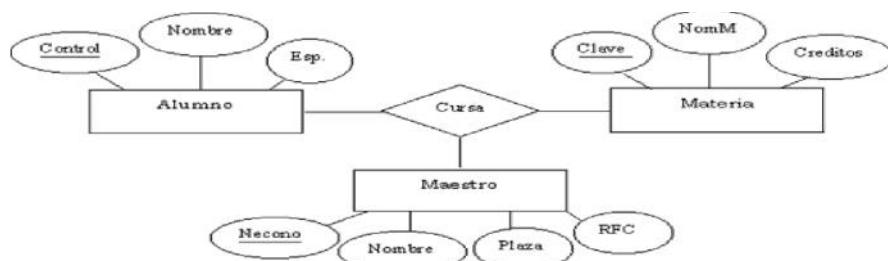
Tercera forma normal y la forma normal de Boyce Codd.

Para definir formalmente la 3FN necesitamos definir dependencia transitiva: En una afinidad (tabla bidimensional) que tiene por lo menos 3 atributos (A,B,C) en donde A determina a B, B determina a C pero no determina a A.

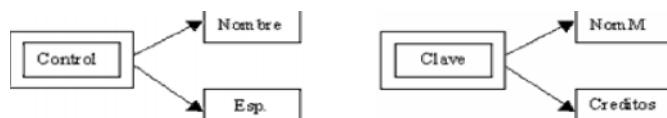
Tercera forma normal.

Definición formal: Una relación R está en 3FN si y solo si esta en 2FN y todos sus atributos no primos dependen no transitivamente de la llave primaria.

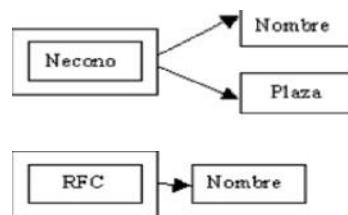
Consiste en eliminar la dependencia transitiva que queda en una segunda forma normal, en pocas palabras una relación esta en tercera forma normal si está en segunda forma normal y no existen dependencias transitivas entre los atributos, nos referimos a dependencias transitivas cuando existe más de una forma de llegar a referencias a un atributo de una relación. Por ejemplo, consideremos el siguiente caso:



Tenemos la relación alumno-cursa-materia manejada anteriormente, pero ahora consideraremos al elemento maestro, gráficamente lo podemos representar de la siguiente manera:



Podemos darnos cuenta que se encuentra graficado en segunda forma normal, es decir que todos los atributos llave están indicados en doble cuadro indicando los atributos que dependen de dichas llaves, sin embargo en la llave Necono tiene como dependientes a 3 atributos en el cual el nombre puede ser referenciado por dos atributos: Necono y RFC (Existe dependencia transitiva), podemos solucionar esto aplicando la tercera forma normal que consiste en eliminar estas dependencias separando los atributos, entonces tenemos:



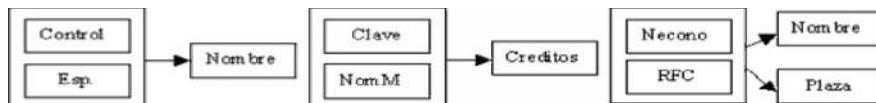
Forma normal de Boyce Codd.

Determinante: Uno o más atributos que, de manera funcional, determinan otro atributo o atributos. En la dependencia funcional $(A,B) \rightarrow C$, (A,B) son los determinantes.

Definición formal:

Una relación R está en FNBC si y solo si cada determinante es una llave candidato.

Denominada por sus siglas en inglés como BCNF; Una tabla se considera en esta forma si y sólo si cada determinante o atributo es una llave candidato. Continuando con el ejemplo anterior, si consideramos que en la entidad alumno sus atributos control y nombre nos puede hacer referencia al atributos esp., entonces decimos que dichos atributos pueden ser llaves candidato. Gráficamente podemos representar la forma normal de Boyce Codd de la siguiente forma:



Obsérvese que a diferencia de la tercera forma normal, agrupamos todas las llaves candidato para formar una global (representadas en el recuadro) las cuales hacen referencia a los atributo que no son llaves candidato.

Cuarta y quinta forma normal

Cuarta forma normal.

Definición formal: Un esquema de relaciones R está en 4FN con respecto a un conjunto D de dependencias funcionales y de valores múltiples si, para todas las dependencias de valores múltiples en D de la forma $X \rightarrow> Y$, donde $X \subset R$ y $Y \subset R$, se cumple por lo menos una de estas condiciones:

- * $X \rightarrow> Y$ es una dependencia de valores múltiples trivial.
- * X es una superllave del esquema R.

Para entender mejor aún esto consideremos una afinidad (tabla) llamada estudiante que contiene los siguientes atributos: Clave, Especialidad, Curso tal y como se demuestra en la siguiente figura:

Clave	Especialidad	Curso
S01	Sistemas	Natación
S01	Bioquímica	Danza
S01	Sistemas	Natación
B01	Bioquímica	Guitarra
C03	Civil	Natación

Suponemos que los estudiantes pueden inscribirse en varias especialidades y en diversos cursos. El estudiante con clave S01 tiene su especialidad en sistemas y Bioquímica y toma los cursos de Natación y danza, el estudiante B01 tiene la especialidad en Bioquímica y toma el curso de Guitarra, el estudiante con clave C03 tiene la especialidad de Civil y toma el curso de natación. En esta tabla o relación no existe dependencia funcional porque los estudiantes pueden tener distintas especialidades, un valor único de clave puede poseer muchos valores de especialidades al igual que de valores de cursos. Por lo tanto existe dependencia de valores múltiples. Este tipo de dependencias produce redundancia de datos, como se puede apreciar en la tabla anterior, en donde la clave S01 tiene tres registros para mantener la serie de datos en forma independiente lo cual ocasiona que al realizarse una actualización se requiera de demasiadas operaciones para tal fin. Existe una dependencia de valores múltiples cuando una afinidad tiene por lo menos tres atributos, dos de los cuales poseen valores múltiples y sus valores dependen solo del tercer atributo, en otras palabras en la afinidad R (A,B,C) existe una dependencia de valores múltiples si A determina valores múltiples de B, A determina valores múltiples de C, y B y C son independientes entre sí. En la tabla anterior Clave determina valores múltiples de especialidad y clave determina valores múltiples de curso, pero especialidad y curso son independientes entre sí. Las dependencias de valores múltiples se definen de la siguiente manera: Clave $\rightarrow> \text{Especialidad}$ y Clave $\rightarrow> \text{Curso}$; Esto se lee "Clave multidetermina a Especialidad, y clave multidetermina a Curso". Para eliminar la redundancia de los datos, se deben eliminar las dependencias de valores múltiples. Esto se logra construyendo dos tablas, donde cada una almacena datos para solamente uno de los atributos de valores múltiples.

Para nuestro ejemplo, las tablas correspondientes son:

Tabla Eespecialidad

Clave	Especialidad
S01	Sistemas
B01	Bioquímica
C03	Civil

Tabla ECurso

Clave	Curso
S01	Natación
S01	Danza
B01	Guitarra
C03	Natación

Quinta forma normal.

Definición formal: Un esquema de relaciones R está en 5FN con respecto a un conjunto D de dependencias funcionales, de valores múltiples y de producto, si para todas las dependencias de productos en D se cumple por lo menos una de estas condiciones:

- * (R1, R2, R3, ... Rn) es una dependencia de producto trivial.
- * Toda Ri es una superllave de R.

La quinta forma normal se refiere a dependencias que son extrañas. Tiene que ver con tablas que pueden dividirse en subtablas, pero que no pueden reconstruirse.

8. Arquitecturas cliente-servidor

• Protocolos de acceso nativos y ODBC (R, E)

Arquitectura cliente/servidor, arquitectura hardware y software adecuada para el proceso distribuido, en el que la comunicación se establece de uno a varios. Un proceso es un programa en ejecución. Proceso cliente es el que solicita un servicio. Proceso servidor es el capaz de proporcionar un servicio. Un proceso cliente se puede comunicar con varios procesos servidores y un servidor se puede comunicar con varios clientes. Los procesos pueden ejecutarse en la misma máquina o en distintas máquinas comunicadas a través de una red. Por lo general, la parte de la aplicación correspondiente al cliente se optimiza para la interacción con el usuario, ejecutándose en su propia máquina, mientras que la parte correspondiente al servidor proporciona la funcionalidad multiusuario centralizada y se ejecuta en una máquina remota. Una aplicación cliente/servidor típica es un servidor de base de datos al que varios usuarios realizan consultas simultáneamente. El proceso cliente realiza una consulta, el proceso servidor le envía las tablas resultantes de la consulta y el proceso cliente las interpreta y muestra el resultado en pantalla.

9. Bases de datos distribuidas

Una Base de Datos Distribuida es, una base de datos construida sobre una red computacional y no por el contrario en una máquina aislada. La información que constituye la base de datos esta almacenada en diferentes sitios en la red, y las aplicaciones que se ejecutan accesan datos en distintos sitios.

Una Base de Datos Distribuida entonces es una colección de datos que pertenecen lógicamente a un sólo sistema, pero se encuentra físicamente esparcido en varios "sitios" de la red. •Un sistema de base de datos distribuidas se compone de un conjunto de sitios, conectados entre sí mediante algún tipo de red de comunicaciones, en el cual :

1. cada sitio es un sistema de base de datos en sí mismo, pero
2. los sitios han convenido en trabajar juntos (si es necesario) con el fin de que un usuario de cualquier sitio pueda obtener acceso a los datos de cualquier punto de la red tal como si todos los datos estuvieran almacenados en el sitio propio del usuario.

En consecuencia, la llamada "base de datos distribuida" es en realidad una especie de objeto virtual, cuyas partes componentes se almacenan físicamente en varias bases de datos "reales" distintas ubicadas en diferentes sitios. De

hecho, es la unión lógica de esas bases de datos. En otras palabras, cada sitio tiene sus propias bases de datos "reales" locales, sus propios usuarios locales, sus propios DBMS y programas para la administración de transacciones (incluyendo programas de bloqueo, bitácoras, recuperación, etc), y su propio administrador local de comunicación de datos (administrador DC). En particular un usuario dado puede realizar operaciones sobre los datos en su propio sitio local exactamente como si ese sitio no participara en absoluto en el sistema distribuido (al menos, ése es uno de los objetivos). Así pues, el sistema de bases de datos distribuidas puede considerarse como una especie de sociedad entre los DBMS individuales locales de todos los sitios. Un nuevo componente de software en cada sitio (en el aspecto lógico, una extensión del DBMS local) realiza las funciones de sociedad necesarias; y es la combinación de este nuevo componente y el DBMS ya existente lo que constituye el llamado "sistema de administración de bases de datos distribuidas" (DDBMS, distributed database management system).

Conceptos básicos.

El sistema de administración de Base de Datos Distribuida (DDBMS), esta formado por las transacciones y los administradores de base de datos distribuidos de todas las computadoras. Tal DDBMS en un esquema genérico implica un conjunto de programas que operan en diversas computadoras. Estos programas pueden ser subsistemas de un producto único DDBMS, concedionado por un sólo fabricante, o también pudiera resultar de una colección de programas de fuentes dispares : algunos considerados por fabricantes y algunos otros escritos en casa.

Un administrador de base de datos (DTM) es un programa que recibe solicitudes de procesamiento de los programas de consulta o de transacciones y a su vez las traduce en acciones para los administradores de la base de datos . Una función importante del DTM es coordinar y controlar dichas acciones. • Cada sitio tiene sus propias bases de datos "reales" locales, sus propios usuarios locales, sus propios DBMS y programas para administración de transacciones y su propio administrador local de comunicación de datos.

La diferencia principal entre los sistemas de bases de datos centralizados y los distribuidos es que en los primeros, los datos residen en una sola localidad, mientras que, en lo últimos, se encuentran en varias localidades. Cada localidad puede procesar transacciones locales , es decir, aquellas que sólo acceden a datos que residen en esa localidad. Además, una localidad puede participar en la ejecución de transacciones globales , es decir, aquellas que acceden a datos de varias localidades, ésta requiere comunicación entre las localidades. • Una transacción local es la que accede a cuentas en la localidad individual donde se inicio. En cambio, una transacción global accede a cuentas de una localidad distinta a la localidad donde se inicio o a cuentas de varias localidades diferentes.

Se asume que cada localidad ejecuta el mismo software de control de base de datos distribuida. Si no se cumple esta premisa, será muy difícil manejar transacciones globales.

¿Por qué son deseables las bases de datos distribuidas?

La respuesta básica a esta pregunta es que por lo regular las empresas ya están distribuidas, por lo menos desde el punto de vista lógico (en divisiones, departamentos, proyectos, etc) y muy probablemente en el sentido físico también (en plantas, talleres, laboratorios, y demás), de lo cual se desprende que en general la información ya está también distribuida, porque cada unidad de organización dentro de la empresa mantendrá por fuerza los datos pertinentes a su propio funcionamiento. Así pues, un sistema distribuido permite que la estructura de la base de datos refleje la estructura de la empresa : los datos locales se pueden mantener en forma local, donde por lógica deben estar, pero al mismo tiempo es posible obtener acceso a datos remotos en caso necesario. Un principio fundamental de los sistemas de bases de datos distribuidos

El principio fundamental de las bases de datos distribuidas :

Desde el punto de vista del usuario, un sistema distribuido deberá ser idéntico a un sistema no distribuido.

En otras palabras, los usuarios de un sistema distribuido deberán comportarse exactamente como si el sistema no estuviera distribuido. Todos los problemas de los sistemas distribuidos son (o deberían ser) internos o a nivel de realización, no externos o a nivel del usuario. Llamaremos al principio fundamental recién identificado la "regla cero" de los sistemas distribuidos. La regla cero conduce a varios objetivos o reglas secundarios - doce en realidad- siguientes :

1. Autonomía local.
2. No dependencia de un sitio central.
3. Operación continua.

4. Independencia con respecto a la localización.
5. Independencia con respecto a la fragmentación.
6. Independencia de réplica.
7. Procesamiento distribuido de consultas.
8. Manejo distribuido de transacciones.
9. Independencia con respecto al equipo.
10. Independencia con respecto al sistema operativo.
11. Independencia con respecto a la red.
12. Independencia con respecto al DBMS.

Estas doce reglas no son todas independientes entre sí, ni son por fuerza exhaustivas, ni tienen todas la misma importancia (diferentes usuarios darán diferentes grados de importancia a diferentes reglas en diferentes ambientes). Sin embargo, sí son útiles como fundamento para entender la tecnología distribuida y como marco de referencia para caracterizar la funcionalidad de sistemas distribuidos específicos. Un último punto introductorio: es importante distinguir los sistemas distribuidos de bases de datos verdaderos, generalizados, de los sistemas que tan solo ofrecen algún tipo de acceso remoto a los datos (llamados a veces sistemas de procesamiento distribuido o sistemas de red). En un " sistema de acceso remoto a los datos ", el usuario podría ser capaz de trabajar con datos de un sitio remoto, o aun con datos de varios sitios remotos al mismo tiempo, pero " se notan las costuras " ; el usuario definitivamente está consciente (en mayor o menor grado) de que los datos son remotos, y debe comportarse de manera acorde. En cambio, en un sistema distribuido verdadero, las costuras son invisibles. Las doce reglas.

1. Autonomía Local. Los sitios de un sistema distribuido deben ser autónomos . La autonomía local significa que todas las operaciones en un sitio dado se controlan en ese sitio; ningún sitio X deberá depender de algún otro sitio Y para su buen funcionamiento (pues de otra manera el sitio X podría ser incapaz de trabajar, aunque no tenga en sí problema alguno, si cae el sitio Y, situación a todas luces indeseable). La autonomía local implica también un propietario y una administración locales de los datos, con responsabilidad local : todos los datos pertenecen " en realidad" a una base de datos local, aunque sean accesibles desde algún sitio remoto. Por tanto, las cuestiones de seguridad, integridad y representación en almacenamiento de los datos locales permanecen bajo el control de la instalación local.
2. No dependencia de un sitio central. La autonomía local implica que todos los sitios deben tratarse igual; no debe haber dependencia de un sitio central "maestro" para obtener un servicio central, como por ejemplo un procesamiento centralizado de las consultas o una administración centralizada de las transacciones, de modo que todo el sistema dependa de ese sitio central. Este segundo objetivo es por tanto un corolario del primero (si se logra el primero, se logrará pro fuerza el segundo) . Pero la "no dependencia de un sitio central" es deseable por sí misma, aun si no se logra la autonomía local completa. Por ello vale la pena expresarlo como un objetivo separado. La dependencia de un sitio central sería indeseable al menos por las siguientes razones : en primer lugar, ese sitio central podrí ser un cuello de botella; en segundo lugar, el sistema sería vulnerable ; si el sitio central sufriera un desperfecto, todo el sistema dejaría de funcionar.
3. Operación continua. En un sistema distribuido, lo mismo que en uno no distribuido, idealmente nunca debería haber necesidad de apagar a propósito el sistema Es decir, el sistema nunca debería necesitar apagarse para que se pueda realizar alguna función, como añadirse un nuevo sitio o instalar una versión mejorada del DBMS en un sitio ya existente.
4. Independencia con respecto a la localización. La idea básica de la independencia con respecto a la localización (también conocida como transparencia de localización) es simple : no debe ser necesario que los usuarios sepan dónde están almacenados físicamente los datos, sino que más bien deben poder comportarse - al menos desde un punto de vista lógico - como si todos los datos estuvieran almacenados en su propio sitio local. La independencia con respecto a la localización es deseable porque simplifica los programas de los usuarios y sus actividades en la terminal. En particular, hace posible la migración de datos de un sitio a otro sin anular la validez de ninguno de esos programas o actividades. Esta posibilidad de migración es deseable pues permite modificar la distribución de los datos dentro de la red en respuesta a cambios en los requerimientos de desempeño.
5. Independencia con respecto a la fragmentación. Un sistema maneja fragmentación de los datos si es posible dividir una relación en partes o "fragmentos" para propósitos de almacenamiento físico. La fragmentación es deseable por razones de desempeño: los datos pueden almacenarse en la localidad donde se utilizan con mayor frecuencia, de manera que la mayor parte de las operaciones sean sólo locales y se reduzca al tráfico en la red. Por ejemplo, la relación empleados EMP podría fragmentarse de manera que los registros de los empleados de Nueva York se almacenen en el sitio de Nueva York, en tanto que los registros de los empleados de Londres se almacenan en el sitio de Londres. Existen en esencia dos clases de fragmentación, horizontal y vertical,

correspondientes a las operaciones relacionales de restricción y proyección; respectivamente. En términos más generales, un fragmento puede ser cualquier subrelación arbitraria que pueda derivarse de la relación original mediante operaciones de restricción y proyección (excepto que, en el caso de la proyección es obvio que las proyecciones deben conservar la clave primaria de la relación original). La reconstrucción de la relación original a partir de los fragmentos se hace mediante operaciones de reunión y unión apropiadas (reunión en el caso de fragmentación vertical, y la unión en casos de fragmentación horizontal). Ahora llegamos a un punto principal : un sistema que maneja la fragmentación de los datos deberá ofrecer también una independencia con respecto a la fragmentación (llamada también transparencia de fragmentación). La independencia con respecto a la fragmentación (al igual que la independencia con respecto a la independencia con respecto a la localización) es deseable porque simplifica los programas de los usuarios y sus actividades en la terminal.

6. Independencia de réplica. Un sistema maneja réplica de datos si una relación dada (ó en términos más generales, un fragmento dado en una relación) se puede representar en el nivel físico mediante varias copias réplicas, en muchos sitios distintos. La réplica es deseable al menos por dos razones : en primer lugar, puede producir un mejor desempeño (las aplicaciones pueden operar sobre copias locales en vez de tener que comunicarse con sitios remotos) ; en segundo lugar, también puede significar una mejor disponibilidad (un objeto estará disponible para su procesamiento en tanto esté disponible por lo menos una copia, al menos para propósitos de recuperación). La desventaja principal de las réplicas es desde luego que cuando se pone al día un cierto objeto copiado, deben ponerse al día todas las réplicas de ese objeto : el problema de la propagación de actualizaciones. La réplica como la fragmentación, debe ser "transparente para el usuario". En otras palabras , un sistema que maneja la réplica de los datos deberá ofrecer también una independencia de réplica (conocida también como transparencia de réplica); es decir, los usuarios deberán poder comportarse como si sólo existiera una copia de los datos. La independencia de réplica es buena porque simplifica los programas de los usuarios y sus actividades en la terminal. En particular, permite la creación y eliminación dinámicas de las réplicas en cualquier momento en respuesta a cambios en los requerimientos, sin anular la validez de esos programas o actividades de los usuarios.
7. Procesamiento distribuido de consultas. En este aspecto debemos mencionar dos puntos amplios. Primero consideremos la consulta "obtener los proveedores de partes rojas en Londres". Supongamos que el usuario está en la instalación de Nueva York y los datos están en el sitio de Londres. Supongamos también que son n registros de Londres a Nueva York. Si, por otro lado, el sistema no es relacional, sino de un registro a la vez, la consulta implicará en esencia $2n$ mensajes : n de Nueva York a Londres solicitando el siguiente registro, y n de Londres a Nueva York para devolver ese siguiente registro. Así, el ejemplo ilustra el punto de que un sistema relacional tendrá con toda probabilidad un mejor desempeño que uno no relacional (para cualquier consulta que solicite varios registros), quizás en varios órdenes de magnitud. En segundo lugar, la optimización es todavía más importante en un sistema distribuido que en uno centralizado. Lo esencial es que, en una consulta como la anterior, donde están implicados varios sitios, habrá muchas maneras de trasladar los datos en la red para satisfacer la solicitud, y es crucial encontrar una estrategia suficiente. Por ejemplo, una solicitud de unión de una relación Rx almacenada en el sitio X y una relación Ry almacenada en el sitio Y podría llevarse a cabo trasladando Rx a Y o trasladando Ry a X, o trasladando las dos a un tercer sitio Z .
8. Manejo distribuido de transacciones. El manejo de transacciones tiene dos aspectos principales, el control de recuperación y el control de concurrencia, cada uno de los cuales requiere un tratamiento más amplio en el ambiente distribuido. Para explicar ese tratamiento más amplio es preciso introducir primero un término nuevo, "agente". En un sistema distribuido, una sola transacción puede implicar la ejecución de código en varios sitios (en particular puede implicar actualizaciones en varios sitios). Por tanto, se dice que cada transacción está compuesta de varios agentes, donde un agente es el proceso ejecutado en nombre de una transacción dada en determinado sitio. Y el sistema necesita saber cuándo los agentes son parte de la misma transacción; por ejemplo, es obvio que no puede permitirse un bloqueo mutuo entre dos agentes que sean parte de la misma transacción. La cuestión específica del control de recuperación; : para asegurar, pues que una transacción dada sea atómica (todo o nada) en el ambiente distribuido, el sistema debe asegurarse de que todos los agentes correspondientes a esa transacción se comprometan al unísono o bien que retrocedan al unísono. Este efecto puede lograrse mediante el protocolo de compromiso en dos fases. En cuanto al control de concurrencia, esta función en un ambiente distribuido estará basada con toda seguridad en el bloqueo, como sucede en los sistemas no distribuidos.
9. Independencia con respecto al equipo. En realidad, no hay mucho que decir acerca de este tema, el título lo dice todo. Las instalaciones de cómputo en el mundo real por lo regular incluyen varias máquinas diferentes - máquinas IBM, DEC, HP, UNISYS, PC etc- y existe una verdadera necesidad de poder integrar los datos en todos esos sistemas y presentar al usuario "una sola imagen del sistema". Por tanto conviene ejecutar el mismo DBMS en diferentes equipos, y además lograr que esos diferentes equipos participen como socios iguales en un sistema distribuido.

10. Independencia con respecto al sistema operativo. Este objetivo es un corolario del anterior. Es obvia la conveniencia no sólo de poder ejecutar el mismo DBMS en diferentes equipos, sino también poder ejecutarlo en diferentes sistemas operativos y lograr que una versión MVS y una UNIX y una PC/DOS participen todas en el mismo sistema distribuido.
11. Independencia con respecto a la red. Si el sistema ha de poder manejar múltiples sitios diferentes, con equipo distinto y diferentes sistemas operativos, resulta obvia la conveniencia de poder manejar también varias redes de comunicación distintas.
12. Independencia con respecto al DBMS. Bajo este título consideramos las implicaciones de relajar la suposición de homogeneidad estricta. Puede alegarse que esa suposición es quizás demasiado rígida. En realidad, no se requiere sino que los DBMS en los diferentes sitios manejen todos la misma interfaz ; no necesitan ser por fuerza copias del mismo sistema.

Ventajas

Existen cuatro ventajas del procesamiento de bases de datos distribuidas. La primera, puede dar como resultado un mejor rendimiento que el que se obtiene por un procesamiento centralizado. Los datos pueden colocarse cerca del punto de su utilización, de forma que el tiempo de comunicación sea más corto. Varias computadoras operando en forma simultánea pueden entregar más volumen de procesamiento que una sola computadora. Segundo, los datos duplicados aumentan su confiabilidad. Cuando falla una computadora, se pueden obtener los datos extraídos de otras computadoras. Los usuarios no dependen de la disponibilidad de una sola fuente para sus datos .Una tercera ventaja , es que los sistemas distribuidos pueden variar su tamaño de un modo más sencillo. Se pueden agregar computadoras adicionales a la red conforme aumentan el número de usuarios y su carga de procesamiento. A menudo es más fácil y más barato agregar una nueva computadora más pequeña que actualizar una computadora única y centralizada. Después, si la carga de trabajo se reduce, el tamaño de la red también puede reducirse.

Por último , los sistemas distribuidos se puede adecuar de una manera más sencilla a las estructuras de la organización de los usuarios.

Nota: Los siguientes puntos están basados en una referencia bibliográfica distinta, y me pareció importante hablar sobre ello, espero y aclare el tema en cuestión. Utilización compartida de los datos y distribución del control. Si varias localidades diferentes están conectadas entre si, entonces un usuario de una localidad puede acceder a datos disponibles en otra localidad. La ventaja principal de compartir datos por medio de la distribución es que cada localidad pueda controlar hasta cierto punto los datos almacenados localmente.

Fiabilidad y disponibilidad:

Si se produce un fallo en una localidad en un sistema distribuido, es posible que las demás localidades puedan seguir trabajando. En particular si los datos se repiten en varias localidades, una transacción o aplicación que requiere un dato específico puede encontrarlo en más de una localidad. Así el fallo, de una localidad no implica necesariamente la desactivación del sistema.

Agilización del procesamiento de consultas:

Si una consulta comprende datos de varias localidades, puede ser posible dividir la consulta en varias subconsultas que se ejecuten en paralelo en distintas localidades. En los casos en que hay repetición de los datos, el sistema puede pasar la consulta a las localidades mas ligeras de carga.

Ejemplo de sistemas: Para efectos de referencia posterior, mencionaremos brevemente algunas de las realizaciones de sistemas distribuidos más conocidas. En primer término, los prototipos. Entre los sistemas investigados, tres de los más conocidos son:

1. SDD-1 creado en la división de investigación de Computer Corporation of America (CCA) a finales de la década de los 1970 y principios de la siguiente.
2. R* (pronunciado "R estrella"), versión distribuida del prototipo System R elaborada en IBM Research a principios de la década de 1980 y ;
3. INGRES distribuido, versión distribuida del prototipo INGRES, creada también a principios de la década de 1980 en la University of California en Berkeley.

Pasando a productos comerciales, algunos de los más conocidos son :

1. INGRES/STAR de Relational Technology, Inc.
2. SQL*STAR, de Oracle Corp. y
3. DB2 versión 2 Edición 2, de IBM.

Desventajas

- ✓ Procesamiento de consultas
- ✓ Administración de catálogo
- ✓ Propagación de Actualizaciones
- ✓ Control de recuperación
- ✓ Control de Concurrencia

Las primeras dos desventajas de las bases de datos distribuidas son las mismas que las dos primeras ventajas. Primero, el rendimiento puede ser peor para el procesamiento distribuido que para el procesamiento centralizado. Depende de la naturaleza de la carga de trabajo, la red, el DDBMS y las estrategias utilizadas de concurrencia y de falla, así como las ventajas del acceso local a los datos y de los procesadores múltiples, ya que éstos pueden ser abrumados por las tareas de coordinación y de control requeridas. Tal situación es probable cuando la carga de trabajo necesita un gran número de actualizaciones concurrentes sobre datos duplicados, y que deben estar muy distribuidos. Segundo, el procesamiento de base de datos distribuida puede resultar menos confiable que el procesamiento centralizado. De nuevo, depende de la confiabilidad de las computadoras de procesamiento, de la red, del DDBMS, de las transacciones y de las tasas de error en la carga de trabajo. Un sistema distribuido puede estar menos disponible que uno centralizado. Estas dos desventajas indican que un procesamiento distribuido no es ninguna panacea. A pesar de que tiene la promesa de un mejor rendimiento y de una mayor confiabilidad, tal promesa no está garantizada. Una tercera desventaja es su mayor complejidad, a menudo se traduce en altos gastos de construcción y mantenimiento. Ya que existen más componentes de hardware, hay más cantidad de cosas por aprender y más interfaces susceptibles de fallar. El control de concurrencia y recuperación de fallas puede convertirse en algo complicado y difícil de implementar, puede empujar a una mayor carga sobre programadores y personal de operaciones y quizás se requiera de personal más experimentado y más costoso. El procesamiento de bases de datos distribuido es difícil de controlar. Una computadora centralizada reside en un entorno controlado, con personal de operaciones que supervisa muy de cerca, y las actividades de procesamiento pueden ser vigiladas, aunque a veces con dificultad. En un sistema distribuido, las computadoras de proceso, residen muchas veces en las áreas de trabajo de los usuarios. En ocasiones el acceso físico no está controlado, y los procedimientos operativos son demasiado suaves y efectuados por personas que tienen escasa apreciación o comprensión sobre su importancia. En sistemas centralizados, en caso de un desastre o catástrofe, la recuperación puede ser más difícil de sincronizar.

Nota: De las desventajas que se mencionan a continuación, están relacionadas con las doce reglas mencionadas en el capítulo anterior.

Procesamiento de Consultas

El problema más grande es que las redes de comunicación (las de larga distancia en especial) son lentas. El objetivo es reducir al mínimo el tráfico en la red y esto implica que el proceso mismo de optimización de consultas debe ser distribuido, además del proceso de ejecución de las consultas. Es decir un proceso representativo consistirá en un paso de optimización global, seguido de pasos de optimización local en cada uno de los sitios afectados.

Administración de Catálogo

En un sistema distribuido, el catálogo del sistema incluirá no solo la información usual acerca de las relaciones, índices, usuarios, sino también toda la información de control necesaria para que el sistema pueda ofrecer la independencia deseada con respecto a la localización, la fragmentación y la replicación.

1. Centralizado (" no depender de un sitio central")
2. Replicas completas (" falta de autonomía, toda la actualización debe ser propagada a cada sitio ")
3. Dividido (muy costoso)
4. Combinación de 1 y 3 (" no depender de un sitio central ")

Propagación de Actualizaciones

El problema básico con la réplica de datos, es la necesidad de propagar cualquier modificación de un objeto lógico dado a todas las copias almacenadas de ese objeto. Un problema que surge es que algún sitio donde se mantiene una copia del objeto puede NO estar disponible, y fracasaría; la modificación si cualquiera de las copias no esté disponible. Para tratar este problema se habla de "una copia primaria" y funciona así :

- ✓ Una de las copias del objeto se designa como copia primaria, y las otras serán secundarias.
- ✓ Las copias primarias de los distintos objetos están en sitios diferentes.
- ✓ Las operaciones de actualización se consideran completas después de que se ha modificado la copia primaria.

El sitio donde se encuentra esa copia se encarga entonces de propagar la actualización a las copias secundarias.

Recuperación

Basado en el protocolo de compromiso de dos fases. El compromiso de dos fases es obligatorio en cualquier ambiente en el cual una sola transacción puede interactuar con varios manejadores de recursos autónomos, pero tiene especial importancia en un sistema distribuido porque los manejadores de recursos en cuestión (o sea los DBMS locales) operan en sitios distintos y por tanto son muy autónomos. En particular, son vulnerables a fallas independientes. Surgen los siguientes puntos :

1. El objetivo de "no dependencia de un sitio central" dicta que la función de coordinador no debe asignarse a un sitio específico de la red, sino que deben realizarla diferentes sitios para diferentes transacciones. Por lo regular se encarga de ella el sitio en el cual se inicia la transacción en cuestión.
2. El proceso de compromiso en dos fases requiere una comunicación entre el coordinador y todos los sitios participantes, lo cual implica más mensajes y mayor costo extra.
3. Si el sitio Y actúa como participante en un proceso de compromiso en dos fases coordinado por el sitio X, el sitio Y deberá hacer lo ordenado pro el sitio X (compromiso o retroceso, según se aplique), lo cual implica otra pérdida de autonomía local.
4. En condiciones ideales, nos gustaría que el proceso de compromiso en dos fases funcionara aun en caso de presentarse fallas de sitios o de la red en cualquier punto. Idealmente, el proceso debería ser capaz de soportar cualquier tipo concebible de falla. Por desgracia es fácil ver que este problema es en esencia imposible de resolver; es decir, no existe un protocolo finito que garantice el compromiso al unísono de una transacción exitosa por parte de todos los agentes, o el retroceso al unísono de una transacción no exitosa en caso de fallas arbitrarias.

Concurrencia

Este concepto tiene que ver con la definición de un agente. El manejo de transacciones tiene dos aspectos principales, el control de recuperación y el control de concurrencia. En un sistema distribuido, una sola transacción puede implicar la ejecución de código en varios sitios (puede implicar actualizaciones en varios sitios), entonces se dice que una transacción está compuesta por varios agentes, donde un agente es el proceso ejecutado en nombre de una transacción dada en determinado sitio. Y el sistema necesita saber cuando dos agentes son parte de la misma transacción.

• Fragmentación (R)

Existen tres tipos de fragmentación:

- ✓ Horizontal
- ✓ Vertical
- ✓ Mixta

Las reglas para validar la fragmentación son:

- ✓ Completitud: la descomposición de una relación R en un conjunto de fragmentos $R_1, R_2, R_3, \dots, R_N$ es completo si y solo si cada elemento de información en R se encuentra en algún R_i .
- ✓ Reconstrucción: si la relación R se descompone en un conjunto de fragmentos $R_1, R_2, R_3, \dots, R_N$, entonces debe existir algún operador relacional ∇ tal que: $R = \nabla_{1 \leq i \leq n} R_i$
- ✓ No-Join: si la relación R se descompone en un conjunto de fragmentos $R_1, R_2, R_3, \dots, R_N$ y un elemento de dato d_i está en R_j , entonces d_i no debe estar en otro fragmento R_k ($k \neq j$)

• **Replicación (R)**

Localización de fragmentos

- ✓ No replicada: cada fragmento está en un solo site
- ✓ Replicada: que existe una copia del fragmento en algún site. Hay dos tipos:
- ✓ Parcialmente: sólo para algunos fragmentos existe una copia.
- ✓ Totalmente: de todos los fragmentos existe al menos una copia.

II. Lenguajes de bases de datos

1. Tipos de lenguajes de consulta

Un lenguaje de consulta es un lenguaje en el que el usuario solicita información de la base de datos. Los lenguajes de consulta se pueden clasificar en procedurales y no procedurales. Los lenguajes procedurales son aquellos en los cuales el usuario instruye al sistema para que lleve a cabo una serie de operaciones en la base de datos con el fin de calcular el resultado deseado. En los lenguajes no procedurales, en cambio, el usuario describe la información deseada sin dar un procedimiento concreto para obtener esta información. Los lenguajes puros para la consulta de datos son 3. El álgebra relacional, el cual es procedural y los cálculos relacionales tanto de tuplas como de dominios, los cuales son lenguajes no procedurales. Estos 3 lenguajes son rígidos y formales, por lo tanto la mayor parte de los sistemas comerciales de bases de datos relacionales ofrecen lenguajes de consulta mixtos, que además de ser ricos en sintaxis ofrecen adicionalmente sublenguajes de definición de datos, administración de seguridad y otras características. Se estudiarán brevemente dos de los tres lenguajes puros de consultas de base de datos, con el fin de presentar una base teórica para introducir SQL. Se excluirá el cálculo relacional de dominios dado que es una generalización del cálculo relacional de tuplas y queda fuera del alcance de este trabajo.

• **SQL ANSI 92: conceptos básicos, definición de datos, consultas y actualización, manejo de vistas, SQL embebido (R)**

Los sistemas de base de datos necesitan un lenguaje de consultas más cómodo para el usuario. Aunque SQL se considere un lenguaje de consultas, contiene muchas otras capacidades que incluyen características para definir estructuras de datos, modificación de datos y la especificación de restricciones de integridad. SQL se ha establecido como el lenguaje estándar de base de datos relacionales. Hay numerosas versiones de SQL. La versión original se desarrolló en el laboratorio de investigación de San Jose, California (San Jose Research Center) de IBM, este lenguaje originalmente denominado Sequel, se implementó como parte del proyecto System R, a principios de 1970 [McJones97]. Desde entonces ha evolucionado a lo que ahora se conoce como SQL (Structured Query Language, o lenguaje estructurado de consultas). En 1986, ANSI (American National Standards Institute, Instituto Nacional Americano de Normalización) e ISO (International Standards Organization, Organización Internacional de Normalización) Publicaron una norma de SQL denominada SQL-86. En 1987 IBM publicó su propia norma de SQL denominada SAA-SQL(System Application Architecture Database Interfaz, Interfaz de base de datos para arquitecturas de aplicación de sistemas). En 1989 se publicó una norma extendida para SQL (SQL-89) y actualmente los SGBD son compatibles al menos con esta norma. La norma actual de SQL de ANSI/ISO es la SQL-92. Se debe tener en cuenta que algunas implementaciones de SQL pueden ser compatibles sólo con SQL-89,

no siéndolo con SQL-92. SQL proporciona dos tipos de lenguajes diferentes: uno para especificar el esquema relacional y el otro para expresar las consultas y actualizaciones de la base de datos.

• Lenguajes de programación 3GL y 4GL (R)

Lenguajes de tercera generación. 3GL

Third Generation Language. Los lenguajes de tercera generación son aquellos lenguajes de programación utilizados por los especialistas para construir aplicaciones que incluyen el procedimiento. Es decir, el programador especifica en su programa qué tiene que hacer el ordenador y cómo debe hacerlo. Se trata de un paso más allá del lenguaje máquina. Son lenguajes de tercera generación Cobol, C, Pascal o Fortran.

Lenguajes de cuarta generación. 4GL

No existe consenso sobre lo que es un lenguaje de cuarta generación (4GL). Lo que en un lenguaje de tercera generación (3GL) como COBOL requiere cientos de líneas de código, tan solo necesita diez o veinte líneas en un 4GL. Comparado con un 3GL, que es procedural, un 4GL es un lenguaje no procedural: el usuario define qué se debe hacer, no cómo debe hacerse. Los 4GL se apoyan en unas herramientas de mucho más alto nivel denominadas herramientas de cuarta generación. El usuario no debe definir los pasos a seguir en un programa para realizar una determinada tarea, tan sólo debe definir una serie de parámetros que estas herramientas utilizarán para generar un programa de aplicación. Se dice que los 4GL pueden mejorar la productividad de los programadores en un factor de 10, aunque se limita el tipo de problemas que pueden resolver. Los 4GL abarcan:

- ✓ Lenguajes de presentación, como lenguajes de consultas y generadores de informes.
- ✓ Lenguajes especializados, como hojas de cálculo y lenguajes de bases de datos.
- ✓ Generadores de aplicaciones que definen, insertan, actualizan y obtienen datos de la base de datos.
- ✓ Lenguajes de muy alto nivel que se utilizan para generar el código de la aplicación.
- ✓ Los lenguajes SQL y QBE son ejemplos de 4GL. Hay otros tipos de 4GL:

Un generador de formularios es una herramienta interactiva que permite crear rápidamente formularios de pantalla para introducir o visualizar datos. Los generadores de formularios permiten que el usuario defina el aspecto de la pantalla, qué información se debe visualizar y en qué lugar de la pantalla debe visualizarse. Algunos generadores de formularios permiten la creación de atributos derivados utilizando operadores aritméticos y también permiten especificar controles para la validación de los datos de entrada. Un generador de informes es una herramienta para crear informes a partir de los datos almacenados en la base de datos. Se parece a un lenguaje de consultas en que permite al usuario hacer preguntas sobre la base de datos y obtener información de ella para un informe. Sin embargo, en el generador de informes se tiene un mayor control sobre el aspecto de la salida. Se puede dejar que el generador determine automáticamente el aspecto de la salida o se puede diseñar ésta para que tenga el aspecto que desee el usuario final. Un generador de gráficos es una herramienta para obtener datos de la base de datos y visualizarlos en un gráfico mostrando tendencias y relaciones entre datos. Normalmente se pueden diseñar distintos tipos de gráficos: barras, líneas, etc. Un generador de aplicaciones es una herramienta para crear programas que hagan de interfase entre el usuario y la base de datos. El uso de un generador de aplicaciones puede reducir el tiempo que se necesita para diseñar un programa de aplicación.

Los generadores de aplicaciones constan de procedimientos que realizan las funciones fundamentales que se utilizan en la mayoría de los programas. Estos procedimientos están escritos en un lenguaje de programación de alto nivel y forman una librería de funciones entre las que escoger. El usuario especifica qué debe hacer el programa y el generador de aplicaciones es quien determina cómo realizar la tarea.

III. Seguridad en las bases de datos

1. Seguridad en bases de datos

- ✓ Seguridad: Es la protección de los datos contra una alteración, destrucción o revelación no autorizada de los mismos.
- ✓ Integridad: Consiste en la exactitud o validez de los datos.

La seguridad implica la protección de los datos contra usuarios no autorizados y el controlar el que los usuarios se ajustan a realizar el trabajo que les está permitido.

La integridad comprueba que los usuarios autorizados hagan las cosas correctamente.

Similitudes de integridad y seguridad: En ambos casos el sistema debe conocer las restricciones que no se pueden violar. Las restricciones las va a establecer el administrador de la base de datos, además el sistema de gestión de bases de datos va a ser el encargado de que la interacción con los usuarios cumple las restricciones.

• Seguridad (R)

Aspectos a tratar por la seguridad:

1. Legales, sociales y éticos.
2. Controles físicos.
3. Cuestiones de política interna.
4. Problemas de operación: Para indicar a un sistema lo que un usuario puede hacer se realiza mediante contraseñas.
5. Controles del equipo: La CPU tiene que tener claves de protección de las áreas de almacenamiento.
6. La seguridad del S.O.
7. Materias de relevancia específica para el sistema mismo de la base de datos.
8. Las decisiones de lo que los usuarios pueden hacer o ver lo que van a tomar el administrador de la base de datos y la dirección de la empresa: El sistema de gestión de la base de datos debe velar para que se cumpla esto y para ello ha de contener las siguientes características:
 - ✓ Los resultados de las decisiones deben de darse a conocer al sistema y el sistema debe ser capaz de recordarlas.
 - ✓ Debe existir alguna forma de verificar una solicitud de acceso, contra las restricciones de seguridad aplicable. Para saber que restricciones se le aplican a una solicitud el sistema debe ser capaz de reconocer esa solicitud, de ahí que no solo baste con tener el identificador del usuario, sino que sea necesario tener también una contraseña.

• Integridad (R)

Reglas de integridad

Una vez definida la estructura de datos del modelo relacional, pasamos a estudiar las reglas de integridad que los datos almacenados en dicha estructura deben cumplir para garantizar que son correctos. Al definir cada atributo sobre un dominio se impone una restricción sobre el conjunto de valores permitidos para cada atributo. A este tipo de restricciones se les denomina restricciones de dominios. Hay además dos reglas de integridad muy importantes que son restricciones que se deben cumplir en todas las bases de datos relacionales y en todos sus estados o instancias (las reglas se deben cumplir todo el tiempo). Estas reglas son la regla de integridad de entidades y la regla de integridad referencial. Antes de definirlas, es preciso conocer el concepto de nulo.

Nulos

Cuando en una tupla un atributo es desconocido, se dice que es nulo. Un nulo no representa el valor cero ni la cadena vacía, éstos son valores que tienen significado. El nulo implica ausencia de información, bien porque al insertar la tupla se desconocía el valor del atributo, o bien porque para dicha tupla el atributo no tiene sentido. Ya que los nulos no son valores, deben tratarse de modo diferente, lo que causa problemas de implementación. De hecho, no todos los SGBD relacionales soportan los nulos.

Regla de integridad de entidades

La primera regla de integridad se aplica a las claves primarias de las relaciones base: ninguno de los atributos que componen la clave primaria puede ser nulo. Por definición, una clave primaria es un identificador irreducible que se utiliza para identificar de modo único las tuplas. Que es irreducible significa que ningún subconjunto de la clave

primaria sirve para identificar las tuplas de modo único. Si se permite que parte de la clave primaria sea nula, se está diciendo que no todos sus atributos son necesarios para distinguir las tuplas, con lo que se contradice la irreducibilidad. Nótese que esta regla sólo se aplica a las relaciones base y a las claves primarias, no a las claves alternativas.

Regla de integridad referencial

La segunda regla de integridad se aplica a las claves ajenas: si en una relación hay alguna clave ajena, sus valores deben coincidir con valores de la clave primaria a la que hace referencia, o bien, deben ser completamente nulos. La regla de integridad referencial se enmarca en términos de estados de la base de datos: indica lo que es un estado ilegal, pero no dice cómo puede evitarse. La cuestión es ¿qué hacer si estando en un estado legal, llega una petición para realizar una operación que conduce a un estado ilegal? Existen dos opciones: rechazar la operación, o bien aceptar la operación y realizar operaciones adicionales compensatorias que conduzcan a un estado legal. Por lo tanto, para cada clave ajena de la base de datos habrá que contestar a tres preguntas:

1. Regla de los nulos: ¿Tiene sentido que la clave ajena acepte nulos?
2. Regla de borrado: ¿Qué ocurre si se intenta borrar la tupla referenciada por la clave ajena?
 - ✓ Restringir: no se permite borrar la tupla referenciada.
 - ✓ Propagar: se borra la tupla referenciada y se propaga el borrado a las tuplas que la referencian mediante la clave ajena.
 - ✓ Anular: se borra la tupla referenciada y las tuplas que la referenciaban ponen a nulo la clave ajena (sólo si acepta nulos).
3. Regla de modificación: ¿Qué ocurre si se intenta modificar el valor de la clave primaria de la tupla referenciada por la clave ajena?
 - ✓ Restringir: no se permite modificar el valor de la clave primaria de la tupla referenciada.
 - ✓ Propagar: se modifica el valor de la clave primaria de la tupla referenciada y se propaga la modificación a las tuplas que la referencian mediante la clave ajena.
 - ✓ Anular: se modifica la tupla referenciada y las tuplas que la referenciaban ponen a nulo la clave ajena (sólo si acepta nulos).

Reglas de negocio

Además de las dos reglas de integridad anteriores, los usuarios o los administradores de la base de datos pueden imponer ciertas restricciones específicas sobre los datos, denominadas reglas de negocio. Por ejemplo, si en una oficina de la empresa inmobiliaria sólo puede haber hasta veinte empleados, el SGBD debe dar la posibilidad al usuario de definir una regla al respecto y debe hacerla respetar. En este caso, no debería permitir dar de alta un empleado en una oficina que ya tiene los veinte permitidos. Hoy en día aún existen SGBD relacionales que no permiten definir este tipo de restricciones ni las hacen respetar.

Integridad: Consideraciones generales.

La integridad la hay que programar a mano por medio del código del programa. La forma de programar una integridad es de forma declarativa:

Ejemplo: Forall SX (SX > 0).

Otra forma de programar la integridad es por medio de técnicas de compensación.

Consideraciones finales.

- ✓ Seguridad: Consiste en proteger la base de datos contra usuarios no autorizados.
- ✓ Integridad: Consiste en proteger la base de datos de los usuarios autorizados.

Tanto seguridad o integridad implican las siguientes características:

- 1.- Definición de restricciones.
- 2.- Especificar las medidas a tomar si se violan esas restricciones.
- 3.- Supervisión por parte del sistema de las operaciones de los usuarios para detectar tales violaciones..

TRATAMIENTO DE INFORMACIÓN

B. Recuperación de información

I. Archivos para grandes volúmenes de datos

1. Métodos de acceso y almacenamiento

- Estructuras de datos auxiliares para organización de información: reservorios, diccionarios de datos, directorios, estructuras de tipo hipertexto (R)

Un esquema de base de datos se especifica por medio de un conjunto de definiciones que se expresan mediante el lenguaje especial llamado lenguaje de definición de datos (DDL). El resultado de la compilación de sentencias de DDL es un conjunto de tablas las cuales se almacenan en un archivo especial llamado directorio de datos (o directorio).

Diccionario de datos.

Base de datos acerca de la terminología que se utilizará en un sistema de información. Para comprender mejor el significado de un diccionario de datos, puede considerarse su contenido como "datos acerca de los datos"; es decir, descripciones de todos los demás objetos (archivos, programas, informes, sinónimos...) existentes en el sistema. Un diccionario de datos almacena la totalidad de los diversos esquemas y especificaciones de archivos, así como sus ubicaciones. Si es completo incluye también información acerca de qué programas utilizan qué datos, y qué usuarios están interesados en unos u otros informes. Por lo general, el diccionario de datos está integrado en el sistema de información que describe. Contiene la información referente a la estructura de la base de datos.

Un sistema de base de datos necesita mantener datos acerca de las relaciones. Esta información se denomina diccionario de datos, o catálogo de sistema. Entre los tipos de información que el sistema debe almacenar están:

- ✓ Los nombres de las relaciones
- ✓ Los nombres de los atributos de cada relación
- ✓ Los dominios de los atributos
- ✓ Los nombres de las vistas definidas en la base de datos y la definición de esas vistas
- ✓ Las restricciones de integridad de cada relación

Además de esto, muchos sistemas conservan los datos siguientes de los usuarios del sistema:

- ✓ Nombres de los usuarios autorizados.
- ✓ Información acerca de los usuarios.

En los sistemas que utilizan estructuras altamente sofisticadas para almacenar relaciones, pueden conservarse datos estadísticos y descriptivos acerca de las relaciones:

- ✓ Números de tuplas en cada relación.
- ✓ Método de almacenamiento utilizado para cada relación.

Directorio (informática)

Organización jerárquica de nombres de archivos almacenados en un disco. El directorio superior se denomina directorio raíz; los directorios existentes dentro de otro directorio se denominan subdirectorios. Según la forma en que el sistema operativo soporte los directorios, los nombres de los archivos allí contenidos pueden verse y ordenarse de distintos modos, como por ejemplo alfabéticamente, por fecha o por tamaño, o en forma de iconos en una interfaz gráfica de usuario. Lo que el usuario ve como directorio está soportado en el sistema operativo en forma de tablas de datos, guardadas en el disco, que contienen las características asociadas con cada archivo, así como la ubicación de éste dentro del disco.

Un directorio de datos es un archivo que contiene un metadatos, es decir, datos sobre datos. Este archivo se consulta antes de leer o modificar los datos reales en el sistema de bases de datos.

Las estructuras de almacenamiento y los métodos de acceso usados por los sistemas de bases de datos se especifican por medio de un conjunto de definiciones en un tipo especial de DDL llamado lenguaje de almacenamiento y definición de datos. El resultado de la compilación de estas definiciones es un conjunto de instrucciones que especifican los detalles de implementación de los esquemas de bases de datos que normalmente se esconden a los usuarios.

Hipertexto

Método de presentación de información en el que el texto, las imágenes, los sonidos y las acciones están unidos mediante una red compleja y no secuencial de asociaciones que permite al usuario examinar los distintos temas, independientemente del orden de presentación de los mismos. Normalmente es el autor el que establece los enlaces de un documento hipertexto en función de la intención del mismo. Por ejemplo, viajando a través de los enlaces de Encarta, la palabra hierro dentro de un artículo puede llevar al usuario a un sistema periódico de elementos o a un artículo referido a la edad del hierro. El término hipertexto fue creado por Ted Nelson en 1965, con el fin de describir los documentos que se presentan en un ordenador o computadora, o sea, expresando la estructura no lineal de las ideas, al contrario de la estructura lineal de los libros, las películas y el habla. El término hipermedia es prácticamente un sinónimo, pero recalca los componentes no textuales del hipertexto, como animaciones, sonido y vídeo.

• Selección de métodos de acceso y almacenamiento en función del volumen de los datos (E)

Tipos de almacenamiento.

Existen varios tipos de medios de almacenamiento que se distinguen por su velocidad relativa, capacidad y resistencia a los fallos.

- ✓ **Almacenamiento volátil.** La información que reside en memoria volátil normalmente no sobrevive a las caídas del sistema. La memoria principal y la memoria caché son ejemplos de este almacenamiento. El acceso a memoria volátil es muy rápido, debido a la velocidad del acceso a memoria y a que es posible acceder a cualquier dato en memoria volátil directamente.
- ✓ **Almacenamiento no volátil.** La información que reside en memoria no volátil normalmente no sobrevive a las caídas del sistema. Los discos y las cintas magnéticas son ejemplos de este almacenamiento. El disco se utiliza para almacenamiento en línea, mientras que la cinta se utiliza para información archivada. Los discos son más fiables que la memoria principal, pero menos fiables que las cintas magnéticas. Sin embargo, ambos están sujetos a fallos (por ejemplo roturas de las cabezas) que pueden resultar en perdida de información. En el estado actual de la tecnología, el almacenamiento no volátil es más lento que el volátil por varios ordenes de magnitud. La diferencia es el resultado de que los dispositivos de disco y cinta sean electromagnéticos en vez de estar basados completamente en chips, como es el caso de la memoria volátil. En los sistemas de base de datos los discos se usan en la mayor parte de almacenamiento no volátil. Otros medios no volátiles normalmente se utilizan solo para guardar una copia de seguridad de los datos.
- ✓ **Almacenamiento estable.** La información que reside en el almacenamiento estable nunca se pierde. Para implementar una aproximación de este tipo de almacenamiento, necesitamos repetir información en varios medios de almacenamiento no volátil con modos de fallo independientes y actualizar la información de una manera controlada.

La jerarquía de almacenamiento

El sistema de base de datos reside en almacenamiento no volátil (generalmente un disco). La base de datos se divide en unidades de longitud fija llamadas *bloques* que son las unidades de asignación de almacenamiento y de transferencia de datos. Las transacciones meten información del disco en la memoria principal y después vuelven a sacar la información al disco. Las operaciones de entrada y salida se hacen en unidades de bloques. Los bloques que residen en el disco se denominan *bloques físicos*, mientras que los bloques que residen temporalmente en la memoria principal se denominan *bloques de registros intermedios (buffer)*.

Un bloque puede contener varios datos. El conjunto exacto de datos que contiene un bloque se determina por medio de la forma de organización de los datos físicos que se va a utilizar (véase el Capítulo 7). Supondremos que ningún dato se extiende sobre dos o más bloques. Esta suposición es realista para la mayor parte de aplicaciones de procesamiento de datos, como el ejemplo bancario. En los Capítulos 12-14 consideraremos situaciones en las que esta suposición no se cumple.

Los movimientos de bloques entre el disco y la memoria principal se inician por medio de las dos operaciones siguientes:

- ✓ **Input(Y),** que transfiere el bloque físico en el que reside el elemento de información X a la memoria principal.

- ✓ **Output(X)**, que transfiere el bloque de registro intermedio en el que reside X al disco y sustituye el bloque físico apropiado que allí se encuentra.

Este esquema se ilustra en la Figura 10.1.

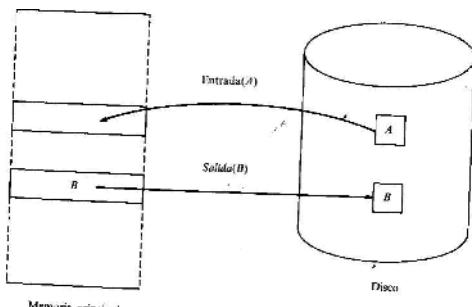


Figura 10.1. Operaciones de almacenamiento de bloques.

Las transacciones interactúan con el sistema de base de datos transfiriendo los datos desde variables del programa a la base de datos y desde la base de datos a variables del programa. Esta transferencia de datos se logra utilizando las dos operaciones siguientes:

- ✓ **read (X, xi)**, que asigna el valor del elemento de información X a la variable local xi. Esta operación se ejecuta como sigue:
 1. Si el bloque en el que reside X no está en memoria principal, entonces se ejecuta **input(X)**.
 2. Se asigna x_i al valor de X del bloque en el registro intermedio (buffer).
- ✓ **write(X, Xi)**, que asigna el valor de la variable local x_i al elemento de información X en el bloqueo que está en el registro intermedio.

Esta operación se ejecuta como sigue:

1. Si el bloque en el que reside X no está en la memoria principal, entonces se ejecuta **input(X)**.
2. Se asigna el valor de x_i a X en el bloque del registro intermedio que contiene a X.

Nótese que ambas operaciones pueden requerir la transferencia de un bloque del disco a la memoria principal. Sin embargo, no requieren específicamente la transferencia de un bloque de la memoria principal al disco.

Un bloque de registro intermedio, eventualmente se graba en el disco, ya sea porque el gestor del registro intermedio necesite el espacio-de memoria para otros propósitos o porque el sistema de base de datos desea reflejar el cambio hecho a X en el disco. Diremos que el sistema de base de datos *fuerza la salida* del bloque en el registro intermedio que contiene a X si se ejecuta una operación **output(X)**.

Cuando una transacción necesita tener acceso a un elemento de información X por primera vez, debe ejecutar **read(X, xi)**. Después de esto, todas las actualizaciones de X se realizan sobre x_i . Después de que la transacción tenga acceso a X por última vez, debe ejecutar **write(A^K, Xi)** para reflejar el cambio hecho a X en la base de datos.

La operación **output(X)** no necesita ejecutarse inmediatamente después de ejecutar **write(X, xi)**, ya que el bloque en el que reside X puede contener otros elementos de información a los que se está accediendo todavía. Por tanto, la salida real tendrá lugar más tarde. Si el sistema se cae después de ejecutar la operación **write(X, xi)** pero antes de ejecutar la operación **output(X)**, el nuevo valor de X nunca se escribe en el disco y, por tanto, se pierde.

Modelo de almacenamiento

Cuando una transacción ejecuta una instrucción **write** no es necesario escribir la actualización en el disco. El **write** sólo afecta a una copia de los datos en el registro intermedio (*buffer*) de la base de datos en la memoria principal. La operación **output** posterior que realice el sistema en una página de datos modificará a la base de datos. Igualmente, los registros de bitácora se guardan en memoria principal al crearse.

No importa cuál sea el número de transacciones concurrentes, existe un único registro intermedio (*buffer*) de la base de datos compartido y un único registro intermedio (*buffer*) de bitácora compartido. Cada transacción tiene su

propia área de trabajo en la que almacena la copia de los datos a los que accede. Comúnmente, las áreas de trabajo de las transacciones se almacenan en la memoria virtual del sistema operativo, La Figura 12.1 muestra un modelo de almacenamiento típico. La memoria principal tiene tres registros intermedios:

- ✓ **Registro intermedio del sistema.** Este registro intermedio contiene páginas de código objeto del sistema y las áreas de trabajo locales de las transacciones activas. Los datos en el registro intermedio del sistema están bajo el control del gestor de la memoria virtual del sistema operativo.
- ✓ **Registro intermedio de bitácora.** Este registro intermedio contiene páginas de registros de bitácora hasta que se graban en el almacenamiento estable.
- ✓ **Registro intermedio de la base de datos.** Este registro intermedio, que contiene páginas de la base de datos, lo gestiona el sistema de base de datos o bien el sistema operativo. Durante un volcado de la base de datos, las páginas se graban en almacenamiento estable en archivos.

El almacenamiento secundario se divide en varias categorías:

- ✓ **Código objeto del sistema,** el código para el sistema.
- ✓ **Área de recopila de memoria virtual,** el área del disco que se usa para almacenar las páginas del área de trabajo de las transacciones locales que no se guarda en la memoria principal.
- ✓ **Almacenamiento estable en línea,** una aproximación de almacenamiento estable que contiene los registros requeridos para recuperación de fallos de almacenamiento volátil.
- ✓ **Almacenamiento estable en archivos,** una aproximación fuera de línea de almacenamiento estable que contiene sólo los datos necesarios para recuperación de fallos de almacenamiento no volátil.

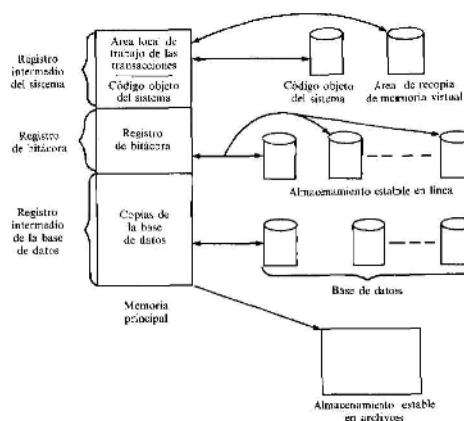


Figura 12.1. Modelo de almacenamiento.

Resaltamos que todas las transacciones comparten el registro intermedio de bitácora y el registro intermedio de la base de datos, pero cada transacción tiene su propia área de trabajo local. El almacenamiento estable en línea permite la recuperación inmediata de caídas del sistema y fallos de transacciones. Puesto que la pérdida de almacenamiento no volátil es menos frecuente, la parte del almacenamiento estable que sólo se necesita para la recuperación de dichos fallos se guarda fuera de línea.

En la práctica se utilizan varias alternativas al modelo de la Figura 12.1. Algunos o todos los registros intermedios pueden combinarse bajo el control del gestor de un registro intermedio. En algunas aplicaciones se requiere una rápida recuperación de fallos de almacenamiento no volátil. Esto se logra utilizando *discos reflejados*, en los que todas las páginas de la base de datos se almacenan dos veces en discos separados. En vez de almacenar la base de datos en un array de discos, se almacena en un array de pares de discos. Un par de este tipo se muestra en la Figura 12.2. Una operación **read** puede dirigirse a cualquier disco (basado en carga). Una operación **write** debe escribir ambos discos en el par. La operación **write** no se considera completa hasta que hayan terminado las dos escrituras en disco físico. En caso que falle un disco en un par, se asigna un nuevo disco y se copian los datos lo antes posible para crear un nuevo segundo miembro del par.

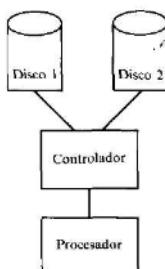


Figura 12.2. Discos reflejados con un único controlador y procesador.

Para obtener mayor resistencia a fallos, puede duplicarse no sólo el disco, sino también el controlador del disco. En sistemas de multiprocesadores, puede darse acceso físico a cada controlador al menos a dos procesadores para proporcionar resistencia a fallos del procesador. La Figura 12.3 ilustra discos reflejados con pares de controladores y procesadores.

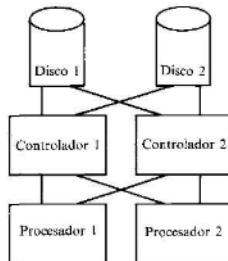


Figura 12.3. Discos reflejados con pares de controladores y procesadores.

• Manejo de índices (E)

Para permitir el acceso aleatorio rápido a los registros de un archivo se utiliza una estructura de índice. Cada estructura de índice está asociada con una clave de búsqueda determinada. Si el archivo está ordenado secuencialmente y elegimos incluir varios índices en diferentes claves de búsqueda, el índice cuya clave de búsqueda especifica el orden secuencial del archivo es el *índice primario*. Los demás se llaman *índices secundarios*. La clave de búsqueda de un índice primario es normalmente la clave primaria.

Suponemos que todos los archivos están ordenados secuencialmente y, por tanto, tienen una clave de búsqueda primaria. Dichos archivos, junto con un índice primario, se llaman *archivos de índices secuenciales*. Se encuentran entre los esquemas de indexación más antiguos usados en los sistemas de bases de datos. Están diseñados para aplicaciones que requieren tanto un procesamiento secuencial del archivo completo como un acceso aleatorio a registros individuales.

Hay dos tipos de índices que pueden usarse

- ✓ **Índice denso.** Aparece un registro índice para cada valor de la clave de búsqueda en el archivo. El registro contiene el valor de la clave de búsqueda y un puntero al registro.
- ✓ **Índice escaso.** Se crean registros índices solamente para algunos de los registros. Para localizar un registro, encontramos el registro índice con el valor de la clave de búsqueda más grande que sea menor o igual que el valor de la clave de búsqueda que estamos buscando. Empezamos en el registro al que apunta el registro índice y seguimos los punteros del archivo hasta encontrar el registro deseado.

Índice primario

La Figura 8.1 muestra un archivo secuencial de registros *depósito* tomados del ejemplo bancario. En el ejemplo de la Figura 8.1, los registros están almacenados en orden de la clave de búsqueda, utilizando *nombre-sucursal* como clave de búsqueda.

	Brighton	217	Green	750
	Downtown	101	Johnson	500
	Downtown	110	Peterson	600
	Mianus	215	Smith	700
	Perryridge	102	Hayes	400
	Perryridge	201	Williams	900
	Perryridge	218	Lyle	700
	Redwood	222	Lindsay	700
	Round Hill	305	Turner	350

Figura 8.1. Archivo secuencial de registros *depósito*.

Las Figuras 8.2 y 8.3 muestran un índice denso y uno escaso, respectivamente, para el archivo *depósito*. Supóngase que buscamos registros de la sucursal Perryridge. Usando el índice denso de la Figura 8.2, seguimos el puntero directamente al primer registro Perryridge. Procesamos este registro y seguimos el puntero en ese registro para localizar el siguiente registro en orden de la clave de búsqueda (*nombre-sucursal*). Continuamos procesando registros hasta que encontramos un registro para una sucursal distinta de Perryridge. Si estamos utilizando el índice escaso (Figura 8.3), no encontramos una entrada de índice para "Perryridge". Puesto que la última entrada (en orden alfabético) antes de "Perryridge" es "Mianus", seguimos ese puntero. Entonces leemos el archivo *depósito* en orden secuencial hasta que encontramos el primer registro Perryridge y empezamos procesando en ese punto.

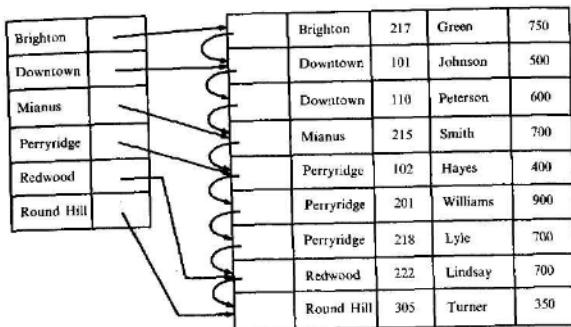


Figura 8.2. Índice denso.

Generalmente es más rápido localizar un registro con un índice denso que con un índice escaso. Sin embargo, los índices escasos tienen una ventaja sobre los índices densos, requieren menos espacio e imponen menos mantenimiento adicional para inserciones y eliminaciones.

Para que esta técnica sea completamente general, debemos considerar el caso en que los registros para un valor de la clave de búsqueda ocupan varios bloques. Es fácil modificar el esquema para manejar esta situación.

Aun cuando utilizamos un índice escaso, el índice puede llegar a ser demasiado grande para un procesamiento eficiente. En la práctica, no es raro tener un archivo con 100 000 registros, con 10 registros almacenados en cada bloque. Si tenemos un registro índice por bloque, el índice tiene 10 000 registros. Los registros índice son más pequeños que los registros de datos, por lo que podemos suponer que caben 100 registros índice en un bloque. Así pues, el índice ocupa 100 bloques.

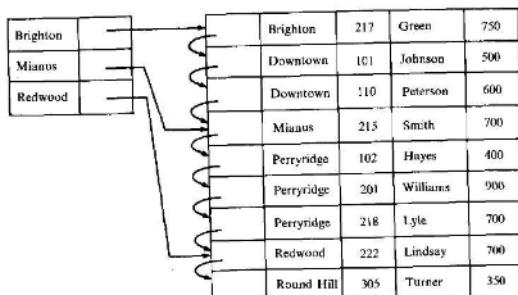


Figura 8.3. Índice escaso.

Si un índice es lo bastante pequeño como para guardarlo en memoria principal, el tiempo de búsqueda es corto. Sin embargo, si el índice es tan grande que se debe guardar en disco, una búsqueda resulta en varias lecturas de bloques del disco. Si el índice ocupa b y se emplea una búsqueda binaria, podemos leer hasta $1 + \log_2(b)$ bloques. Para el índice de 100 bloques, significa 7 lecturas de bloque. Así, el proceso de búsqueda del índice puede ser costoso.

Para resolver este problema tratamos el índice como trataríamos cualquier otro archivo secuencial, y construimos un índice escaso sobre el índice primario, como se muestra en la Figura 8.4. Para localizar un registro, primero empleamos una búsqueda binaria en el índice externo para encontrar el registro que corresponda al valor de clave de búsqueda más grande, menor o igual que el que deseamos. El puntero señala a un bloque del índice interno. Examinamos este bloque hasta que encontramos el registro que tiene el valor de clave de búsqueda más grande, menor o igual que el que deseamos. El puntero de este registro apunta al bloque del archivo que contiene el registro que estamos buscando.

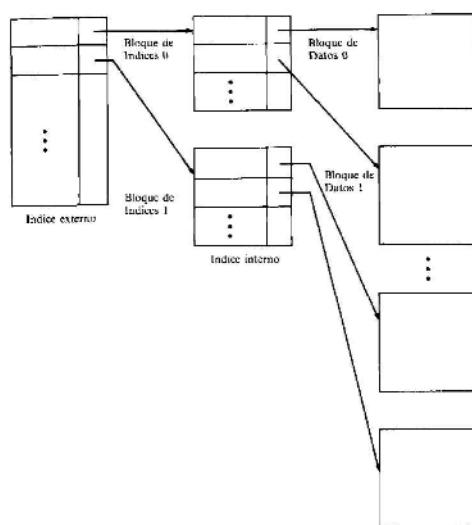


Figura 8.4 Índice escaso de dos pixeles

Utilizando los dos niveles de indexación, hemos leído únicamente un bloque de índices en vez de siete, si suponemos que el índice externo ya está en la memoria principal. Si el fichero es extremadamente grande, es posible que ni siquiera el índice exterior quepa en memoria principal. En un caso así, todavía podemos crear otro nivel de indexación. De hecho, podemos repetir este proceso tantas veces como sea necesario.

Sin embargo, en la práctica, lo normal es que basten dos niveles, y las situaciones que requieren más de tres niveles son extremadamente raras. Frecuentemente, cada nivel de índices corresponde a una unidad de almacenamiento físico. Así, podemos tener índices en los niveles de pista, cilindro y disco.

Si importar cuál sea la forma de índice que se utilice, se deben actualizar todos los índices cada vez que se inserta o elimina un registro del archivo. A continuación describimos algoritmos para actualizar índices de un solo nivel.

- elimina un registro del archivo. A continuación describimos algoritmos para actualizar índices de un solo nivel.

 - ✓ **Eliminación.** Para eliminar un registro, es necesario buscar el registro que se va a eliminar. Si el registro eliminado era el último que quedaba con ese valor particular en la clave de búsqueda, entonces eliminamos el valor de la clave de búsqueda del índice. Para índices densos, eliminamos un valor de la clave de búsqueda de la misma manera que se suprime en un archivo. Para índices escasos, eliminamos un valor de clave sustituyendo su entrada en el índice (si existe una) por el siguiente valor de la clave de búsqueda (en orden de la clave de búsqueda). Si el siguiente valor de la clave de búsqueda ya tiene una entrada de índice, eliminamos la entrada.
 - ✓ **Inserción.** Se hace una búsqueda usando el valor de la clave de búsqueda que aparece en el registro que se va a insertar. Si el índice es denso y el valor de la clave de búsqueda no aparece en el índice, lo inserta. Si el índice es escaso no se necesita hacer ningún cambio en el índice a menos que se

crea un nuevo bloque. En este caso, el primer valor de la clave de búsqueda (en orden de la clave de búsqueda) que aparezca en el nuevo bloque se inserta en la base de datos.

Índices secundarios

Los índices secundarios pueden estructurarse de forma diferente a los índices primarios. La Figura 8.5 muestra la estructura de un índice secundario que usa un nivel extra de indirección. Los punteros en el índice secundario no señalan directamente al archivo. En vez de ello, cada uno de esos punteros señala a una cubeta que contiene punteros al archivo.

Este enfoque permite almacenar juntos todos los punteros de un valor de clave de búsqueda secundaria determinado. Un enfoque así es útil en ciertos tipos de consultas para los que podemos hacer una parte considerable de procesamiento utilizando únicamente los punteros. Para las claves primarias, podemos obtener todos los punteros para un valor de la clave de búsqueda primaria determinado utilizando una revisión secuencial.

Una *revisión secuencial* en orden de clave primaria es eficiente porque los registros están almacenados físicamente en un orden que se aproxima al orden de clave primaria. Sin embargo, no podemos (excepto en raras excepciones) almacenar un archivo físicamente ordenado tanto por la clave primaria como por una clave secundaria. Como el orden de clave secundaria y el de clave física son distintos, si intentamos examinar el archivo secuencialmente en orden de clave secundaria, es probable que la lectura de cada registro requiera la lectura de un nuevo bloque del disco.

Almacenando punteros en una cubeta como se muestra en la Figura 8.5, eliminamos la necesidad de punteros adicionales en los registros mismos y de revisiones secuenciales en orden de clave secundaria.

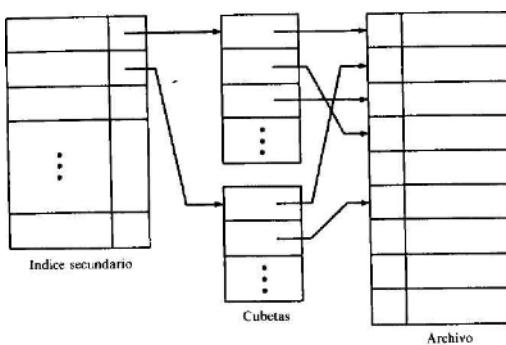


Figura 8.5. Índice secundario.

El índice secundario puede ser denso o escaso. Si es denso, entonces el puntero de cada cubeta individual señala a los registros con el valor de la clave de búsqueda apropiado. Si el índice secundario es escaso, entonces el puntero de cada cubeta individual señala a los registros con valores de la clave de búsqueda en el rango apropiado. En este caso cada entrada de cubeta es un puntero único o bien un registro que consta de dos campos: un valor de la clave de búsqueda y un puntero a algún registro de archivo.

Para ilustrar estos conceptos, considérese el índice secundario escaso de la Figura 8.6 en la clave secundaria *nombre-cliente*. Si las cubetas contienen únicamente punteros, entonces, si estamos realizando una búsqueda de "Peterson", debemos leer los tres registros a los que apuntan las entradas de la cubeta 2. Sólo una entrada apunta a un registro para el cual el valor de *nombre-cliente* es "Peterson", pero es necesario leer tres registros. Puesto que el archivo no está ordenado físicamente por *nombre-cliente*, es de esperar que esta búsqueda requiera tres lecturas de bloques del disco. Compárese este ejemplo con el ejemplo anterior del índice primario escaso de la Figura 8.3. Ya que el archivo está almacenado físicamente en orden de *nombre-sucursal*, lo normal es que baste una lectura de bloque para una búsqueda.

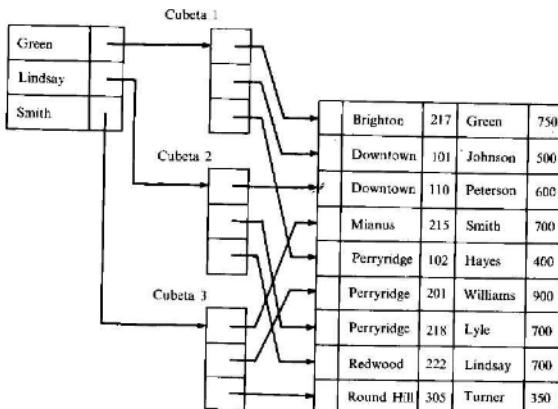


Figura 8.6. Índice secundario escaso por nombre-cliente

Asociando un valor de la clave de búsqueda con cada puntero de la cubeta, eliminamos la necesidad de leer registros con un valor de la clave de búsqueda secundaria distinto del que estamos buscando.

La estructura de cubeta puede eliminarse si el índice secundario es denso y los valores de la clave de búsqueda forman una clave primaria.

El procedimiento descrito anteriormente para eliminación e inserción puede aplicarse a un archivo con múltiples índices. Cada vez que se modifique el archivo es necesario actualizar *todos* los índices.

Los índices secundarios mejoran el rendimiento de las consultas que utilizan claves que no son primarias. Sin embargo, implican un gasto extra considerable en la modificación de la base de datos. El diseñador basándose en una estimación de la frecuencia relativa de consultas y modificaciones.

Tipos de indexación.

- ✓ **Directa.** Utilizar un campo para almacenar la dirección física en el disco (nº cuenta=dirección física). No es muy eficiente, ya que hay que reservar gran espacio de memoria para el almacenamiento.
- ✓ **Por tablas índice.** Mediante índices densos. Cada uno de ellos está compuesto de dos campos:
 1. valor semántico
 2. puntero a donde apunta el valor semántico

S1	S1 ... Londres	Londres
S2	S2 ... París	París
S3	S3 ... Londres	Londres
S4	S4 ... Madrid	Madrid
S5	S5 ... París	París
S6	S6 ... Roma	Roma
S7	S7 ... Madrid	Madrid

Una Tabla índice es una lista invertida (a partir de un campo encontramos un registro). Cuando una base de datos tiene tantas tablas índices como campos, se dice que está totalmente invertida.

- ✓ **Tablas índice múltiples.** En el caso de que poseamos muchas tablas índice, podemos indexarlas de forma compuesta unas con otras. Es muy importante el orden por el que se vaya componiendo. Se las conoce como Tablas de índices múltiples.

Si nuestra tabla es muy grande, entonces necesitamos utilizar los índices no densos que se

componen de dos campos:

- * Apunta a la página donde está el registro
- * El mayor valor semántico de esa página

S# RID	TABLA	
S1 011	S3 01	Todos los Registros tienen una llave primaria, que es un Campo que nos permite diferenciar únicamente al registro.
S2 012	S5 02	
S3 013	S7 03	
S4 021		
S5 022		
S6 031		
S7 032		

- ✓ **Árboles B.** Son una serie de tablas índice, expandidas en forma de árbol, y cada rama es la composición de las siguientes, y cuyas hojas acaban en el RID.

50 82 91		
12 32 50	55 70 82	89 98 99
6 8 12 15 18 32 35 40 50 51 52 58 60 62 70 71 76 82 85 86 89 91 93 94 96 97 99		
R.I.D.		

Son árboles equilibrados, ya que para llegar a sus hojas, siempre hay que pasar por el mismo número de niveles.

Un árbol B es una estructura de árbol equilibrado, cuyos nodos son las páginas de sucesivas tablas índices y donde los nodos terminales son las propias páginas de la tabla índice.

Al conjunto de nodos no terminales se le llama conjunto índice y al conjunto de nodos terminales conjunto secuencia.

La profundidad de un árbol es el nº de niveles que tiene. El orden de un árbol B tiene que ser tal que en cada nodo haya como máximo $2n$ elementos y como mínimo n elementos.

La forma correcta de tener los nodos del árbol, es dejar un elemento vacío para poder insertar nuevos valores, sin tener que modificar todo el árbol.

Para localizar un elemento en el árbol nos situamos en la raíz y escogemos la rama cuyo índice sea menor o igual al elemento buscado (y mayor que el anterior), llegando al nodo 1 y así sucesivamente hasta llegar a la hoja, donde hacemos una búsqueda secuencial, obteniendo el RID.

A la hora de insertar pueden pasar dos cosas: Que haya sitio o que no lo haya.

40 60 80	40 60 72 80
25 30 40 45 47 60 63 70 75 80	63 70 72 75 80

Si no hay sitio, dividimos el nodo en dos, y se reestructura desde ese nivel hacia arriba.

A la hora de borrar pueden pasar dos cosas: Que se cumpla el orden del árbol, o que no se cumpla. Si no se cumple el orden, se trata de unir con su hermano a la derecha, siempre que tenga sitio, si no tiene sitio, se divide en dos y se inserta en el nodo más próximo.

- ✓ **Directo (Hashing).** A partir de una clave, y por medio de una función (f) encontrar el RID de ese registro.

Algoritmo de Direccionamiento:

$f: C \rightarrow 0, \dots, N$ $f(C_j) = n$ para cada valor no se repite

Se trata de buscar la función en la que se produzca el mínimo de sinónimos y huecos (distribuye los valores entre 0 y N).

Las más utilizadas son las de números aleatorios y son:

1. **Plegamiento:** se pasa el nº a binario y se suma la primera mitad del número con su segunda mitad y el resultado es la dirección.
2. **Cuadrados generales:** elevamos el nº al cuadrado y se cogen las cifras centrales a conveniencia.
3. **Método de congruencia:** se sigue la fórmula $C = (a * C + b) \bmod N$
4. **Desplazamiento:** se van sumando las cifras, para que quede una cifra más pequeña (en sucesivos pasos).

Para evitar los sinónimos y los huecos se puede utilizar Páginas de Overflow, pero es muy liso.

Entonces aparece el Hashing, se definen una serie de cubos para ir almacenando los valores. Se sigue la fórmula $(K \bmod 100) \bmod 2 + 1$

El resultado es el cubo donde hay que almacenarlo. Si el cálculo nos lleva a un cubo cuyos 10 registros están llenos, entonces se va a la página de overflow y se almacena allí.

Lo primero que hay que controlar es el tamaño de los cubos, que será lo suficientemente grande para no utilizar mucho la página de overflow y lo suficientemente pequeño para que no queden muchos huecos.

Los registros se pueden ordenar dentro del cubo y así las búsquedas son más fáciles. La transformación de la clave para obtener la dirección, tiene que ser la más adecuada (pocos sinónimos).

Los criterios para elegir el número de páginas de overflow (1/3 del nº de cubos).

- ✓ **Encadenada (por punteros).** Se utiliza para indexar por un campo que no es clave.

El campo escogido será un campo con pocos valores (rango) pero que se repite mucho.

S#	Ciudad	
S1	Vigo	
S2	Ourense	
S3	Lugo	
S4	Coruña	
S5	Ourense	
S6	Vigo	Se enlazan con punteros, con lo que las
S7	Coruña	inserciones son mucho más rápidas y fáciles

Esta técnica se utiliza porque al trabajar con las tablas de índices, si queremos insertar no es posible.

Entonces se utiliza la Tabla Relocalizable:

TRATAMIENTO DE INFORMACIÓN

C. Sistemas de información

I. Teoría de sistemas

1. Conceptos generales

• Conceptos de teoría general de sistemas (R)

En un sentido amplio, la Teoría General de Sistemas (TGS) se presenta como una forma sistemática y científica de aproximación y representación de la realidad y, al mismo tiempo, como una orientación hacia una práctica estimulante para formas de trabajo transdisciplinarias.

En tanto paradigma científico, la TGS se caracteriza por su perspectiva holística e integradora, en donde lo importante son las relaciones y los conjuntos que a partir de ellas emergen. En tanto práctica, la TGS ofrece un ambiente adecuado para la interrelación y comunicación fecunda entre especialistas y especialidades.

La TGS es un ejemplo de perspectiva científica. En sus distinciones conceptuales no hay explicaciones o relaciones con contenidos pre establecidos, pero sí con arreglo a ellas podemos dirigir nuestra observación, haciéndola operar en contextos reconocibles.

Los objetivos originales de la Teoría General de Sistemas son los siguientes:

1. Impulsar el desarrollo de una terminología general que permita describir las características, funciones y comportamientos sistémicos.
2. Desarrollar un conjunto de leyes aplicables a todos estos comportamientos y, por último,
3. Promover una formalización (matemática) de estas leyes.

La primera formulación en tal sentido es atribuible al biólogo Ludwig von Bertalanffy (1901-1972), quien acuñó la denominación "Teoría General de Sistemas". Para él, la TGS debería constituirse en un mecanismo de integración entre las ciencias naturales y sociales y ser al mismo tiempo un instrumento básico para la formación y preparación de científicos.

Sobre estas bases se constituyó en 1954 la Society for General Systems Research, cuyos objetivos fueron los siguientes:

1. Investigar el isomorfismo de conceptos, leyes y modelos en varios campos y facilitar las transferencias entre aquellos.
2. Promoción y desarrollo de modelos teóricos en campos que carecen de ellos.
3. Reducir la duplicación de los esfuerzos teóricos
4. Promover la unidad de la ciencia a través de principios conceptuales y metodológicos unificadores.

La perspectiva de la TGS surge en respuesta al agotamiento e inaplicabilidad de los enfoques analítico-reduccionistas y sus principios mecánico-causales (Arnold & Rodríguez, 1990b). Se desprende que el principio clave en que se basa la TGS es la noción de totalidad orgánica, mientras que el paradigma anterior estaba fundado en una imagen inorgánica del mundo.

A poco andar, la TGS concitó un gran interés y pronto se desarrollaron bajo su alero diversas tendencias, entre las que destacan la cibernetica (N. Wiener), la teoría de la información (C. Shannon y W. Weaver) y la dinámica de sistemas (J. Forrester).

Si bien el campo de aplicaciones de la TGS no reconoce limitaciones, al usarla en fenómenos humanos, sociales y culturales se advierte que sus raíces están en el área de los sistemas naturales (organismos) y en el de los sistemas artificiales (máquinas). Mientras más equivalencias reconozcamos entre organismos, máquinas, hombres y formas de organización social, mayores serán las posibilidades para aplicar correctamente el enfoque de la TGS, pero mientras más experimentemos los atributos que caracterizan lo humano, lo social y lo cultural y sus correspondientes sistemas, quedarán en evidencia sus inadecuaciones y deficiencias (sistemas triviales).

2. Sistemas de información

Un sistema de información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. En un sentido amplio, un sistema de información no necesariamente incluye equipo electrónico. Sin embargo en la práctica se utiliza como sinónimo de “sistema de información computarizada”

Un sistema de información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. El equipo computacional: el hardware necesario para que el sistema de información pueda operar. El recurso humano que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema. Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.

Entrada de Información: Es el proceso mediante el cual el Sistema de Información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos. Esto último se denomina interfaces automáticas. Las unidades típicas de entrada de datos a las computadoras son las terminales, las cintas magnéticas, las unidades de diskette, los códigos de barras, los escáneres, la voz, los monitores sensibles al tacto, el teclado y el mouse, entre otras.

Almacenamiento de información: El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos. La unidad típica de almacenamiento son los discos magnéticos o discos duros, los discos flexibles o diskettes y los discos compactos (CD-ROM).

Procesamiento de Información: Es la capacidad del Sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones, lo que hace posible, entre otras cosas, que un tomador de decisiones genere una proyección financiera a partir de los datos que contiene un estado de resultados o un balance general de un año base.

Salida de Información: La salida es la capacidad de un Sistema de Información para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, terminales, diskettes, cintas magnéticas, la voz, los graficadores y los plotters, entre otros. Es importante aclarar que la salida de un Sistema de Información puede constituir la entrada a otro Sistema de Información o módulo. En este caso, también existe una interfase automática de salida. Por ejemplo, el Sistema de Control de Clientes tiene una interfase automática de salida con el Sistema de Contabilidad, ya que genera las pólizas contables de los movimientos procesales de los clientes. A continuación se muestran las diferentes actividades que puede realizar un Sistema de Información de Control de Clientes:

Actividades que realiza un Sistema de Información:

Entradas:

- ✓ Datos generales del cliente: nombre, dirección, tipo de cliente, etc.
- ✓ Políticas de créditos: límite de crédito, plazo de pago, etc.
- ✓ Facturas (interfase automático).
- ✓ Pagos, depuraciones, etc.

Proceso:

- ✓ Cálculo de antigüedad de saldos.
- ✓ Cálculo de intereses moratorios.
- ✓ Cálculo del saldo de un cliente.

Almacenamiento:

- ✓ Movimientos del mes (pagos, depuraciones).
- ✓ Catálogo de clientes.
- ✓ Facturas.

Salidas:

- ✓ Reporte de pagos.
- ✓ Estados de cuenta.
- ✓ Pólizas contables (interfase automática)
- ✓ Consultas de saldos en pantalla de una terminal.

• **Operativos (R)**

Desde su creación, las computadoras digitales han utilizado un sistema de codificación de instrucciones en sistema de numeración binaria, es decir con los OS. Esto se debe a que los circuitos integrados funcionan con este principio, es decir, hay corriente o no hay corriente. En el origen de la historia de las computadoras (hace unos cuarenta años), los sistemas operativos no existían y la introducción de un programa para ser ejecutado se convertía en un increíble esfuerzo que solo podía ser llevado a cabo por muy pocos expertos. Esto hacia que las computadoras fueran muy complicadas de usar y que se requiriera tener altos conocimientos técnicos para operarlas. Era tan complejo su manejo, que en algunos casos el resultado llegaba a ser desastroso. Además, el tiempo requerido para introducir un programa en aquellas grandes máquinas de lento proceso superaba por mucho el de ejecución y resultaba poco provechosa la utilización de computadoras para resolución de problemas prácticos. Se buscaron medios más elaborados para manipular la computadora, pero que a su vez simplificaran la labor del operador o el usuario. Es entonces cuando surge la idea de crear un medio para que el usuario pueda operar la computadora con un entorno, lenguaje y operación bien definido para hacer un verdadero uso y explotación de esta. Surgen los sistemas operativos. Un sistema operativo es el encargado de brindar al usuario una forma amigable y sencilla de operar, interpretar, codificar y emitir las órdenes al procesador central para que este realice las tareas necesarias y específicas para completar una orden. El sistema operativo, es el instrumento indispensable para hacer de la computadora un objeto útil. Bajo este nombre se agrupan todos aquellos programas que permiten a los usuarios la utilización de este enredo de cables y circuitos, que de otra manera serían difíciles de controlar. Un sistema operativo se define como un conjunto de procedimientos manuales y automáticos, que permiten a un grupo de usuarios compartir una instalación de computadora eficazmente.

Interfaz de Línea de Comandos.

La forma de interfaz entre el sistema operativo y el usuario en la que este escribe los comandos utilizando un lenguaje de comandos especial. Los sistemas con interfaces de líneas de comandos se consideran más difíciles de aprender y utilizar que los de las interfaces gráficas. Sin embargo, los sistemas basados en comandos son por lo general programables, lo que les otorga una flexibilidad que no tienen los sistemas basados en gráficos carentes de una interfaz de programación.

Interfaz Gráfica del Usuario.

Es el tipo de visualización que permite al usuario elegir comandos, iniciar programas y ver listas de archivos y otras opciones utilizando las representaciones visuales (iconos) y las listas de elementos del menú. Las selecciones pueden activarse bien a través del teclado o con el Mouse. Para los autores de aplicaciones, las interfaces gráficas de usuario ofrecen un entorno que se encarga de la comunicación con el ordenador o computadora. Esto hace que el programador pueda concentrarse en la funcionalidad, ya que no está sujeto a los detalles de la visualización ni a la entrada a través del Mouse o el teclado. También permite a los programadores crear programas que realicen de la misma forma las tareas más frecuentes, como guardar un archivo, porque la interfaz proporciona mecanismos estándar de control como ventanas y cuadros de diálogo. Otra ventaja es que las aplicaciones escritas para una interfaz gráfica de usuario son independientes de los dispositivos: a medida que la interfaz cambia para permitir el uso de nuevos dispositivos de entrada y salida, como un monitor de pantalla grande o un dispositivo óptico de almacenamiento, las aplicaciones pueden utilizarlos sin necesidad de cambios.

Funciones de los Sistemas Operativos.

- ✓ Interpreta los comandos que permiten al usuario comunicarse con el ordenador.
- ✓ Coordina y manipula el hardware de la computadora, como la memoria, las impresoras, las unidades de disco, el teclado o el Mouse.
- ✓ Organiza los archivos en diversos dispositivos de almacenamiento, como discos flexibles, discos duros, discos compactos o cintas magnéticas.
- ✓ Gestiona los errores de hardware y la pérdida de datos.
- ✓ Servir de base para la creación del software logrando que equipos de marcas distintas funcionen de manera análoga, salvando las diferencias existentes entre ambos.
- ✓ Configura el entorno para el uso del software y los periféricos; dependiendo del tipo de máquina que se emplea, debe establecerse en forma lógica la disposición y características del equipo. Como por ejemplo, una

microcomputadora tiene físicamente dos unidades de disco, puede simular el uso de otras unidades de disco, que pueden ser virtuales utilizando parte de la memoria principal para tal fin. En caso de estar conectado a una red, el sistema operativo se convierte en la plataforma de trabajo de los usuarios y es este quien controla los elementos o recursos que comparten. De igual forma, provee de protección a la información que almacena.

Categoría de los Sistemas Operativos.

Sistema Operativo Multitareas.

Es el modo de funcionamiento disponible en algunos sistemas operativos, mediante el cual una computadora procesa varias tareas al mismo tiempo. Existen varios tipos de multitareas. La conmutación de contextos (context Switching) es un tipo muy simple de multitarea en el que dos o más aplicaciones se cargan al mismo tiempo, pero en el que solo se está procesando la aplicación que se encuentra en primer plano (la que ve el usuario). Para activar otra tarea que se encuentre en segundo plano, el usuario debe traer al primer plano la ventana o pantalla que contenga esa aplicación. En la multitarea cooperativa, la que se utiliza en el sistema operativo Macintosh, las tareas en segundo plano reciben tiempo de procesado durante los tiempos muertos de la tarea que se encuentra en primer plano (por ejemplo, cuando esta aplicación está esperando información del usuario), y siempre que esta aplicación lo permita. En los sistemas multitarea de tiempo compartido, como OS/2, cada tarea recibe la atención del microprocesador durante una fracción de segundo. Para mantener el sistema en orden, cada tarea recibe un nivel de prioridad o se procesa en orden secuencial. Dado que el sentido temporal del usuario es mucho más lento que la velocidad de procesamiento del ordenador, las operaciones de multitarea en tiempo compartido parecen ser simultáneas.

Sistema Operativo Monotareas.

Los sistemas operativos monotareas son más primitivos y es todo lo contrario al visto anteriormente, es decir, solo pueden manejar un proceso en cada momento o que solo puede ejecutar las tareas de una en una. Por ejemplo cuando la computadora está imprimiendo un documento, no puede iniciar otro proceso ni responder a nuevas instrucciones hasta que se termine la impresión.

Sistema Operativo Monousuario.

Los sistemas monousuarios son aquellos que nada más puede atender a un solo usuario, gracias a las limitaciones creadas por el hardware, los programas o el tipo de aplicación que se esté ejecutando.

Estos tipos de sistemas son muy simples, porque todos los dispositivos de entrada, salida y control dependen de la tarea que se está utilizando, esto quiere decir, que las instrucciones que se dan, son procesadas de inmediato; ya que existe un solo usuario. Y están orientados principalmente por los microcomputadores.

Sistema Operativo Multiusuario.

Es todo lo contrario a monousuario; y en esta categoría se encuentran todos los sistemas que cumplen simultáneamente las necesidades de dos o más usuarios, que comparten mismos recursos. Este tipo de sistemas se emplean especialmente en redes. En otras palabras consiste en el fraccionamiento del tiempo (timesharing).

Secuencia por Lotes.

La secuencia por lotes o procesamiento por lotes en microcomputadoras, es la ejecución de una lista de comandos del sistema operativo uno tras otro sin intervención del usuario. En los ordenadores más grandes el proceso de recogida de programas y de conjuntos de datos de los usuarios, la ejecución de uno o unos pocos cada vez y la entrega de los recursos a los usuarios. Procesamiento por lotes también puede referirse al proceso de almacenar transacciones durante un cierto lapso antes de su envío a un archivo maestro, por lo general una operación separada que se efectúa durante la noche. Los sistemas operativos por lotes (batch), en los que los programas eran tratados por grupos (lote) en vez de individualmente. La función de estos sistemas operativos consistía en cargar en memoria un programa de la cinta y ejecutarlo. Al final de este, se realizaba el salto a una dirección de memoria desde donde reasumía el control del sistema operativo que cargaba el siguiente programa y lo ejecutaba. De esta manera el tiempo entre un trabajo y el otro disminuía considerablemente.

Tiempo Real.

Un sistema operativo en tiempo real procesa las instrucciones recibidas al instante, y una vez que han sido procesadas muestra el resultado. Este tipo tiene relación con los sistemas operativos monousuarios, ya que existe un solo operador y no necesita compartir el procesador entre varias solicitudes.

Su característica principal es dar respuestas rápidas; por ejemplo en un caso de peligro se necesitarían respuestas inmediatas para evitar una catástrofe.

Tiempo Compartido.

El tiempo compartido en ordenadores o computadoras consiste en el uso de un sistema por más de una persona al mismo tiempo. El tiempo compartido ejecuta programas separados de forma concurrente, intercambiando porciones de tiempo asignadas a cada programa (usuario). En este aspecto, es similar a la capacidad de multitareas que es común en la mayoría de los microordenadores o las microcomputadoras. Sin embargo el tiempo compartido se asocia generalmente con el acceso de varios usuarios a computadoras más grandes y a organizaciones de servicios, mientras que la multitarea relacionada con las microcomputadoras implica la realización de múltiples tareas por un solo usuario

• Tácticos (R)

• Estratégicos (R)

Wisman considera a los SIS (Strategic Information Systems, en español SIE: Sistema de Información Estratégicos) como “el uso de la tecnología de la información para apoyar o dar forma a la estrategia competitiva de la organización, a su plan para incrementar o mantener la ventaja competitiva de la organización, a bien para reducir la ventaja de sus rivales”. Así, los sistemas de información de una organización pueden clasificarse como tradicionales o estratégicos. Los sistemas tradicionales son aquellos que tienen como objetivo principal cubrir las necesidades de automatización de las operaciones básicas o cumplir con la función de generar información como apoyo para la toma de decisiones (sistemas transaccionales y sistemas de apoyo a las decisiones, respectivamente).

Sus principales características son:

- ✓ Su función primordial no es apoyar la automatización de procesos operativos ni proporcionar información para apoyar la toma de decisiones. Sin embargo, este tipo de sistemas puede llevar a cabo dichas funciones.
- ✓ Suelen desarrollarse “in house”, es decir, dentro de la organización, por lo tanto no pueden adaptarse fácilmente a paquetes disponibles de mercado.
- ✓ Típicamente, su forma de desarrollo se basa en incrementos y a través de su evolución dentro de la organización. Se inicia con un proceso o función en particular y a partir de ahí se van agregando nuevas funciones o procesos.
- ✓ Su función es lograr ventajas que los competidores no poseen, tales como ventajas en costos y servicios diferenciados con clientes y proveedores. En este contexto, los sistemas estratégicos son creadores de barreras de entrada al negocio. Por ejemplo, el uso del comercio electrónico en algunas compañías que ofrecen servicios de venta a los clientes, es un sistema estratégico, ya que brindan una ventaja sobre otras compañías que ofrecen productos similares y no cuenten con este servicio. Si una compañía nueva decide abrir sus puertas al público, tendrá que dar este servicio para tener un nivel similar al de sus competidores.
- ✓ Otra característica es que las ventajas que se logran a través de estos sistemas no son “eternas”, es decir existe un periodo de vigencia similar al tiempo en que tardan los competidores en alcanzar las diferencias o ventajas obtenidas por el sistema de información estratégico (SIE). Cuando esto sucede, los beneficios generados por el SIE se convierten en estándares de la industria, como es el caso del comercio electrónico que posteriormente serán requisitos de infraestructura tecnológica que una compañía deba poseer.
- ✓ Apoyan el proceso de innovación de productos y procesos dentro de la empresa, debido a que buscan ventajas respecto a los competidores y una forma de hacerlo es innovando o creando productos y procesos.

Un ejemplo de estos sistemas de información dentro de la empresa puede ser un sistema MRP (manufacturing resource planning) enfocado a reducir sustancialmente el desperdicio durante el procesos productivo, o bien, un centro de información que proporcione todo tipo de información, como situación de créditos, embarques, tiempos de entrega, etc.

• Sistemas para trabajo en grupo (GDSS) (R)

Según DeSanctis y Gallupe, un sistema de soporte para la toma de decisiones de grupo (GDSS) es un sistema interactivo basado en computadora, el cual facilita la solución de problemas no estructurados por un conjunto de tomadores de decisiones que trabajan juntos como un grupo. Los componentes de que consta son: hardware, software, recursos humanos y procedimientos. Al unir los componentes de un GDSS es posible realizar juntas cuya temática esté relacionada con el proceso de toma de decisiones.

Características: los sistemas de soporte a la toma de decisiones de grupo deben reunir un conjunto de características para considerarse como tales. Las características principales que deben incluir son:

- ✓ Los GDSS son sistemas diseñados especialmente para apoyar las decisiones en grupo, lo que implica que no están formados por elementos de sistemas ya existentes.

- ✓ La meta de un GDSS es apoyar el trabajo de los tomadores de decisiones, por lo que el uso de este sistema mejora el proceso de toma de decisiones y decisiones resultantes.
- ✓ Un GDSS es fácil de aprender y de usar. Debe ser accesible para usuarios con diferentes niveles de conocimientos computacional y de soporte a la decisión. Generalmente los usuarios son administradores de cualquier área funcional de la empresa, tales como ventas, producción, recursos humanos, administración y finanzas.
- ✓ Un GDSS puede ser específico o general. Es específico si se diseña para un tipo o clase de problema, y es general si se diseña para tomar diversas decisiones organizacionales. Por ejemplo, si se utiliza un GDSS para apoyar el proceso de compra de materia prima del producto X, se trata de un sistema específico, por el contrario, si el GDSS apoya el proceso de la compra de cualquier materia prima y el proceso de venta de los productos, se trata de un sistema general.
- ✓ Un GDSS contiene mecanismos para evitar el desarrollo de conductas negativas en el grupo, como son los problemas de comunicación, estar de acuerdo con lo que dice la mayoría sólo por estar con el grupo o proponer ideas con intención de molestar a algún miembro del grupo.
- ✓ Un GDSS debe motivar a todos los miembros del grupo a participar de manera activa. Es importante que se cuide el anonimato de la participación.

Estas características de los GDSS dan un panorama de la aplicación de este tipo de sistemas. Un GDSS puede utilizarse por grupos de personas que están en una misma localidad y que desean tomar decisiones acerca de un problema específico, como la compra de materia prima para la elaboración de un producto, por ejemplo, y también para llevar a cabo aprovechando las ventajas de las telecomunicaciones para resolver una gran variedad de problemas como contratación de personal, ofrecimiento de productos, programación de ventas, diagnóstico de mercados y planeación estratégica.

Estos sistemas apoyan la realización de las actividades básicas necesarias en un grupo que toma decisiones: obtener, compartir y usar información, la cual se obtiene al seleccionar ciertos valores de datos que se encuentran en una base de datos o al solicitar información en general. Compartir la información incluye enviarla a todos los miembros seleccionados del grupo. El uso de la información se refiere a aplicar la tecnología del software para llegar a tomar decisiones.

Ventajas y Desventajas del uso de GDSS: El uso de un sistema de soporte a la toma de decisiones en grupo cambia el enfoque tradicional en el cual se realizaban las juntas de toma de decisiones. Las principales ventajas que se derivan del uso de esta tecnología son:

- ✓ Motiva a los miembros del grupo a trabajar juntos, ya que se pueden aportar varias ideas al mismo tiempo, lo cual elimina la situación de que pocos miembros dominen el desarrollo de la junta. En este contexto, el GDSS evita que unas cuantas personas se adueñen del “microfono” y frenen la creatividad y las aportaciones del resto del grupo.
- ✓ Mejora la etapa de preparación de la reunión de trabajo, pues debe existir una mejor planeación de las sesiones de trabajo para adecuarla a la tecnología, para así tratar de conseguir los objetivos fijados para cada una de ellas.
- ✓ Da la misma oportunidad de participación a todos los miembros del grupo, debido a que cada uno tiene su propio equipo y puede participar las veces que quiera hacerlo.
- ✓ Cuando en una junta es necesario que estén presentes muchas personas se optimiza el uso de la información que aporta cada miembro del grupo.
- ✓ Proporciona un mecanismo para enfocar al grupo en problemas clave y descartar las conductas que perjudican el desarrollo de la junta de toma de decisiones, tal como distraerse del tema central de la junta y utilizar gran parte de la sesión en tratar temas irrelevantes.
- ✓ Apoya el desarrollo de una memoria organizacional de una junta a otra, pues permite almacenar más información sobre lo que se ha logrado. El término memoria organizacional se utiliza para referirse al resumen de lo que se analizó en la junta y a toda la documentación que se generó en esta.
- ✓ Mejora la calidad de la toma de decisiones debido a que el anonimato de las contribuciones permite una mayor y mejor participación por parte de los miembros del grupo.

Como toda tecnología, el uso de los sistemas de soporte a la toma de decisiones en grupo por parte de la alta administración tiene numerosas ventajas; sin embargo, también tiene algunas desventajas. Las principales son:

- ✓ Falta de costumbre al utilizar un sistema para soportar el proceso de toma de decisiones diferente al de la forma tradicional de realizarlo. Es necesario dar cursos de inducción o capacitar a los miembros de un grupo para que utilicen de manera adecuada un GDSS.
- ✓ Resistencia al cambio por parte de los administradores, porque pueden pensar que este sistema puede desplazarlos, sobre todo a los que están acostumbrados a dirigir el rumbo de toda la junta sin dar oportunidad a que los demás miembros participen.
- ✓ La responsabilidad al tomar una decisión puede diluirse, ya que las aportaciones son anónimas y la decisión representa el consenso del grupo.

Que en el grupo no exista una cultura desarrollada de trabajo en equipo y en consenso, lo cual haga que el uso de GDSS, se realice de manera forzada.

• Sistemas de flujo (*Workflow*) (R)

CSCWP: Computer Supported Collaborative Work Processing

"Automatización de los procesos que se usan diariamente en una empresa. Una aplicación de Workflow automatiza la secuencia de acciones, actividades y tareas usadas para ejecutar los procesos, incluyendo monitoreo en cada instante del proceso, así como las herramientas para administrar el proceso mismo".

"Flujo de trabajo es la secuencia de acciones y pasos usados en procesos de negocios. Flujo de trabajo automatizado aplica tecnología al proceso, aunque no necesariamente en cada acción. En esta definición es necesario que más de una persona esté involucrada".

Flujo de trabajo en imágenes de documentos

Dos enfoques para Workflow basado en mensajes:

Aplicaciones sobre e-mail mejorado: notificaciones, recibos, formularios, reenvío basado en reglas, llenado automático, firma digital, etc. Suficientes para soportar Workflow liviano, Workflow administrativo y Workflow ad-hoc.

Aplicaciones apoyadas por e-mail pero con extensiones de Workflow: paquetes de formularios, ruteo secuencial de formularios, monitoreo del progreso en la ruta de un formulario, etc.

Workflow basado en bases de datos:

Implementado sobre un producto comercial, que generalmente tienen manejo de documentos.

Workflow pensado como asistentes del trabajador: creatividad, manejo de excepciones, procesamiento de tareas.

Workflow Ad-Hoc : Apoyan procesos que pueden cambiar y no tan intensivos en transacciones.

Ejemplos: definición de un nuevo producto, marketing de un producto existente, contratar una nueva persona. Existen fechas límites y responsables, pero éstos pueden cambiar.

Workflow Administrativo: Usualmente basado en sistemas de e-mail, maneja tareas administrativas rutinarias.

E-mail con extensiones tales como:

- ✓ Creación de formularios simples.
- ✓ Ruteo de formularios.
- ✓ Iteración para completación del formulario.
- ✓ Fechas críticas, notificaciones, alarmas, etc.
- ✓ Ayuda al procesamiento de trabajo

Al diseñar un sistema de Workflow, considerar objetos involucrados tales como dispositivos, personal, agenda de tareas, etc.

Diseñar diagramas de flujo de datos para procesos de negocios, procedimientos cooperativos, proyectos departamentales y corporativos.

• **Sistemas para la toma de decisiones (R)**

Se define como a los sistemas de soporte para la toma de decisiones como un conjunto de programas y herramientas que permiten obtener de manera oportuna la información que se requiere durante el proceso de la toma de decisiones que se desarrolla en un ambiente de incertidumbre. En la mayoría de los casos, lo que constituye el detonante de una decisión es el tiempo límite o máximo en el que se debe tomar. Así, al tomar cualquier decisión siempre se podrá pensar que no se tiene toda la información requerida; sin embargo, al llegar al límite del tiempo se debe tomar una decisión. Esto implica necesariamente que el verdadero objetivo de un sistema de apoyo a las decisiones es proporcionar la mayor cantidad de información relevante en el menor tiempo posible, con el fin de decidir lo más adecuado.

Tipos de sistemas de apoyo a las decisiones: entre los tipos de sistemas de información que apoyan el proceso de toma de decisiones se identifican los siguientes:

- ✓ **Sistemas de soporte para la toma de decisiones** (DSS: Decision Support Systems) que tienen como finalidad apoyar la toma de decisiones mediante la generación y evaluación sistemática de diferentes alternativas o escenarios de decisión mediante el empleo de modelos o herramientas computacionales. Un DSS no soluciona problemas, ya que solo apoya al proceso de toma de decisiones. La responsabilidad de tomar una decisión, de adoptarla y de ponerla en práctica es de los administradores, no del DSS. Un DSS puede usarse como apoyo durante las primeras 3 fases del modelo de toma de decisiones de Simon. También para obtener información que revele los elementos clave de los problemas y las relaciones entre ellos. Además puede usarse para identificar, crear y comunicar cursos de acción disponibles y alternativas de decisión. Asimismo, para facilitar el proceso de selección mediante la estimación de costos y beneficios que resultan de cada alternativa.
- ✓ **Sistemas de Información para ejecutivos** (EIS: Executive Information Systems), dirigidos a apoyar el proceso de toma de decisiones de los altos ejecutivos de una organización, presentan información relevante y usan recursos visuales y de fácil interpretación, con el objetivo de mantenerlos informados.
- ✓ **Sistemas para la toma de decisiones de grupo** (GDSS: Group Decision Support Systems), los cuales cubren el objetivo de lograr la participación de un grupo de personas durante la toma de decisiones en ambientes de anonimato y consenso, apoyando decisiones simultáneas.
- ✓ **Sistemas expertos de soporte para la toma de decisiones** (EDSS: Expert Decision Support Systems), los que permiten cargar bases de conocimiento integrados por una serie de reglas de sentido común para que diferentes usuarios las consulten, apoyen la toma de decisiones, la capacitación, etc.

Características de los sistemas de soporte para la toma de decisiones (DSS): Existen varias características que deben estar presentes en un sistema para poder considerarlo un DSS. A continuación se explican algunas de ellas:

- ✓ **Interactividad:** sistema computacional que puede interactuar en forma amigable y con respuestas a tiempo real con el encargado de tomar decisiones.
- ✓ **Tipo de decisiones:** Apoya el proceso de toma de decisiones estructuradas y no estructuradas.
- ✓ **Frecuencia de uso:** tiene una utilización frecuente por parte de la administración media y alta para el desempeño de su función.
- ✓ **Variedad de usuarios:** puede ser empleado por usuarios de diferentes áreas funcionales como ventas, producción, administración, finanzas y recursos humanos.
- ✓ **Flexibilidad:** permite acoplarse a una variedad determinada de estilos administrativos: autocráticos, participativos, etc.
- ✓ **Desarrollo:** permite que el usuario desarrolle de manera directa modelos de decisión sin la participación operativa de profesionales en informática.
- ✓ **Interacción ambiental:** permite interactuar con la información externa como parte de los modelos de decisión.
- ✓ **Comunicación interorganizacional:** facilita la comunicación de información relevante de los niveles altos hacia los niveles operativos y viceversa, a través de gráficas.
- ✓ **Acceso a base de datos:** tiene capacidad de acceder información de las bases de datos corporativas.
- ✓ **Simplicidad:** simple y fácil de aprender y utilizar por el usuario final.

Como complemento de estas características, los DSS integran en su mayoría un conjunto de modelos que apoyan las diferentes decisiones a las que se enfrenta el que debe tomarlas. El conjunto de modelos se conoce como base de modelos. Finalmente se requiere hacer notar que la implantación de un DSS puede realizarse en microcomputadoras o mainframes, lo cual depende de la infraestructura computacional disponible. Las ventajas que se obtienen al hacerlos a través de microcomputadoras son las siguientes:

- ✓ Menores costos
- ✓ Disponibilidad de una gran variedad de herramientas en el mercado que operan en el ambiente de microcomputadoras.

- ✓ Muy baja dependencia de personas que se encuentran fuera de control del tomador de decisiones.

Sin embargo, los inconvenientes pueden ser:

- ✓ Falta de integridad y consolidación para administrar la información. Debido a que cada usuario maneja su propia información y es posible que la modifique según sus necesidades, lo cual puede ocasionar problemas al tratar de consolidar la información de toda la compañía.
- ✓ Problemas de seguridad de la información, lo que se debe a que la información está disponible para cualquier persona que utilice una microcomputadora, no obstante que en ocasiones sea confidencial o que no debe ser acezada sin previa autorización.
- ✓ Perdida del control administrativo por parte del área de informática, debido a que cada usuario maneja en forma independiente el DSS y la información.

II. Análisis y diseño de sistemas de información

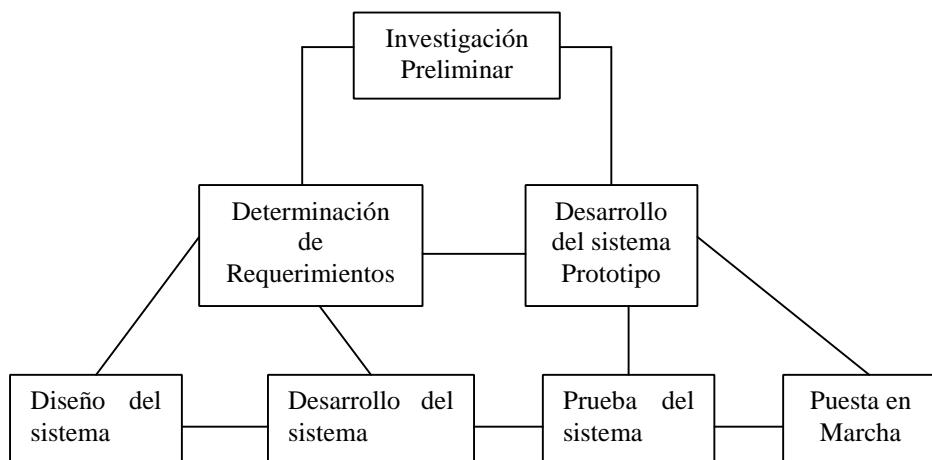
1. Ciclo de vida de sistemas

El desarrollo de sistemas es un proceso que consiste en dos etapas principales de análisis y diseño de sistemas; comienza cuando la gerencia, o en algunas ocasiones el personal de desarrollo de sistemas, se da cuenta de cierto sistema del negocio necesita mejorarse.

El ciclo de vida del desarrollo de sistemas es el conjunto de actividades de los analistas, diseñadores y usuarios, que necesitan llevarse a cabo para desarrollar y poner en marcha un sistema de información. Se debe tener presente que en la mayoría de las situaciones del negocio, las actividades están íntimamente relacionadas y son inseparables.

El ciclo de vida del desarrollo de sistemas consiste en las siguientes actividades:

1. Investigación preliminar
2. Determinación de requerimientos
3. Desarrollo de sistema prototipo
4. Diseño de sistema
5. Desarrollo de software
6. Prueba de los sistemas
7. Puesta en marcha



• Análisis preliminar (R)

Investigaciones preliminares: ¿Cuántas veces se está en situaciones en donde se pregunta si no existe una mejor manera de hacer algo? Por ejemplo, abrir una tienda departamental adicional que creará una necesidad para nuevos procedimientos de facturación, cuando un alto porcentaje de clientes utiliza la cuenta de crédito de esta compañía de esta compañía y compra en todas las tiendas. Duplicar el número de clientes para agrandar las instalaciones y la introducción de muchos nuevos productos, puede traer nuevos requerimientos de pago e cuentas. Un cambio en las áreas de los gerentes departamentales puede guiarlos hacia nuevas formas para registrar las ventas, con implicaciones para el sistema de entrada de pedidos basado en computadora. Una compañía en crecimiento, puede contemplar los sistemas de información computarizados como una forma para hacer posible el crecimiento continuo, sin tener dificultades en el proceso de los pedidos de los clientes.

Se puede iniciar una petición por muchas razones, pero la clave es que alguien, ya sea gerente, un empleado o un especialista de sistemas, inicie un requerimiento para recibir ayuda de un sistema de información. Cuando ese requerimiento se realiza, la primera actividad de sistemas, es decir, la investigación preliminar, se inicia. Esta actividad tiene tres partes: clasificación de requerimiento, estudio de la factibilidad y aprobación del requerimiento. El resultado será aprobar el requerimiento para la atención posterior o rechazarlo como no factible para un desarrollo futuro.

Clarificación del requerimiento: En las empresas muchos requerimientos de los empleados y usuarios no están establecidos claramente; por lo tanto, antes de que pueda considerarse la investigación del sistema, el proyecto requerido debe examinarse para determinar para determinar precisamente lo que desea la empresa. Una simple llamada telefónica puede ser suficiente si la persona que requiere el servicio tiene una idea clara, pero no sabe cómo establecerla. Por otro lado, la persona que hace el requerimiento puede estar simplemente pidiendo ayuda sin saber qué es lo que está mal o por qué existe un problema. La clarificación del problema es este caso, antes de poder llegar a otro paso, el requerimiento de proyecto debe estar claramente establecido.

Estudio de Factibilidad: Un resultado importante de la investigación preliminar es la determinación de que el sistema requerido es factible. Existen tres aspectos en el estudio de factibilidad de la investigación preliminar:

1. Factibilidad técnica. ¿Puede realizarse el trabajo para el proyecto con el equipo actual, tecnología de software y el personal disponible? Si se requiere nueva tecnología, ¿qué probabilidades hay de que pueda desarrollarse?
2. factibilidad económica. ¿Existen suficientes beneficios en la creación del sistema para hacer que los costos sean aceptables? O, en forma inversa, ¿son tan altos los costos como para que el proyecto no deba llevarse a cabo?
3. Factibilidad operativa. ¿Se utilizará el sistema si se desarrolla y pone en marcha? Habrá resistencia de los usuarios, que los posibles beneficios reducirán del sistema.

El estudio de factibilidad se lleva a cabo con un pequeño grupo de gente, familiarizada con las técnicas de los sistemas de información, que entienden la parte de la empresa que será afectada por el proyecto y tienen los conocimientos suficientes del proceso de análisis y diseño de sistemas.

Aprobación del requerimiento: No todos los proyectos requeridos son deseables o factibles. Sin embargo, aquellos que son tanto factibles como deseables deben anotarse para tomarlos en cuenta. En algunos casos, el desarrollo puede comenzar inmediatamente, pero en la mayor parte, los miembros del departamento de sistemas están ocupados en otros proyectos que se encuentran en marcha. Cuando esto sucede, la gerencia decide que los proyectos son más importantes y entonces los programas. Después de que se aprueba la requisición de un proyecto, se estima su costo, la prioridad, el tiempo de terminación y los requerimientos del personal que se utilizan, para determinar qué lista existente los proyectos se incluirán.

Posteriormente, cuando se terminan algunos proyectos anteriores, puede iniciarse el desarrollo de la aplicación propuesta. En este momento, comienza la recabación de datos y la determinación de los requerimientos.

• Análisis detallado (R)

Determinación de requerimientos: El punto clave de análisis de sistemas se consigue al adquirir un conocimiento detallado de todas las facetas importantes dentro del área de negocios que se investiga. (Por esta razón, a menudo esta actividad se conoce como *investigación detallada*.) Los analistas, al trabajar con los empleados y gerentes, deben estudiar el proceso que actualmente se efectúa para contestar estas preguntas clave:

1. ¿Qué se está haciendo?
2. ¿Cómo se está haciendo?
3. ¿Qué tan frecuentemente ocurre?
4. ¿Qué tan grande es la cantidad de transacciones o decisiones?
5. ¿Qué tan bien se lleva acabo la tarea?
6. ¿Existe algún problema?
7. ¿Si el problema existe, qué tan serio es?
8. ¿Si el problema existe, cuál es la causa principal?

Para contestar estas preguntas, los analistas de sistemas hablarán con diferentes personas para recabar los

detalles en relación con el proceso, así como sus opiniones sobre las causas por las cuales suceden las cosas de esa manera y algunas ideas en relación a modificarlas. Se utilizan cuestionarios para recopilar esta información, aplicándolos a grandes que no pueden entrevistarse en forma individual. Las investigaciones detalladas también requieren el estudio de manuales y reportes, la observación real de las actividades de las actividades de trabajo y algunas veces la recabación de formas y documentos para entender completamente el proceso.

Conforme se recopilan los elementos, los analistas estudian los requerimientos de datos para identificar las características que tendrá el nuevo sistema, incluyendo la información que el sistema debe producir y las características operativas, como son controles de procesamiento, tiempos de respuesta y métodos de entrada y salida.

• Diseño preliminar (R)

Desarrollo del sistema prototípico: La **preparación de prototipos** es el proceso de crear, desarrollar y refinar un modelo funcional del sistema final. Se puede crear un **modelo prototípico preliminar** durante la etapa de definición del problema. Un miembro del equipo de reconocimiento -suponga que se trata de un especialista en el procesamiento de datos- puede construir un modelo de este tipo que muestre la composición de las pantallas y los formatos de los informes. Durante una sesión de requerimientos, otros miembros del equipo y usuarios del futuro sistema examinan esta muestra en la forma con el constructor del modelo entiende en principio el problema y los resultado que debe producir el sistema. En este momento puede iniciarse un proceso de refinación si los usuarios señalan omisiones y equivocas.

Durante este proceso de refinación, cuyo objetivo es definir la necesidad que existe, uno o más miembros del equipo pueden utilizar una computadora personal y un **paquete de programas de prototipos** a fin de crear una serie de pantallas en la computadora personal. Estas pantallas no son las salidas que producen los programas ya terminados, pero pueden parecerse mucho a esos resultados. Es posible exhibir en el monitor de la computadora, como una secuencia de diapositivas, menús de captura de datos, la interfaz con el usuario debe servir para buscar, consultar y manipular datos y el formato de los informes de salida. Por ejemplo, se pueden simular los resultados de una serie de selecciones hechas en menús para que los usuarios tengan una idea más clara de la forma como el constructor o los constructores del sistema están interpretando el problema. Si los usuarios no están convencidos de lo que se exhibe define con precisión sus necesidades, pueden modificar fácilmente las plantillas prototípico hasta que estén satisfechos. La creación de un modelo preliminar de prototípico en este punto produce varios beneficios: los usuarios pueden ver que se está avanzado, se les motiva para que participen activamente en la definición del problema, se mejora la comunicación entre todas las partes interesadas y se aclaran los equívocos en una etapa temprana del estudio de sistemas, antes de que se conviertan en costosos errores.

Como se acaba de ver, puede ser necesario un proceso repetitivo (o *interactivo*) para terminar el paso de definición del problema. No existe un procedimiento definido que se deba seguir antes de que se pueda iniciar el análisis detallado del sistema. Un alto ejecutivo puede creer que existen diferencias de información. Puede preparar una declaración general de los objetivos y nombrar a un gerente para que realice un reconocimiento. Pueden realizarse varias sesiones de requerimientos para traducir los deseos generales a objetivos más específicos. Asimismo, pueden crearse y refinarse modelos preliminares de prototípico; se puede ampliar o reducir el alcance del estudio y es posible también que cambien los objetivos conforme se reúnan los datos. Una vez que parezca haberse logrado la aprobación en cuanto a la definición del problema, el equipo de reconocimiento deberá poner la definición detallada *por escrito* y enviarla a todas las personas interesadas, las cuales deberán aprobarla también por escrito. Si persisten diferencias, deberán resolverse en sesiones adicionales de requerimientos. Hay quienes se impacientan con los "retrasos" en el desarrollo del sistema causados por estas sesiones adicionales. Sin embargo, las personas más prudentes saben que los retrasos verdaderamente largos y costosos se presentan cuando los usuarios descubren, ya muy avanzados el proceso del desarrollo, que el sistema diseñado no es satisfactorio por haberse pasado por alto algunos requerimientos.

• Diseño detallado (R)

Diseño del sistema: El diseño de un sistema de información produce los elementos que establecen cómo el sistema cumplirá los requerimientos indicados durante el análisis de sistemas. A menudo los especialistas de sistemas se refieren a esta etapa como en *diseño lógico*, en contraste con desarrollo del software de programas, que se conoce como *diseño físico*.

Los analistas de sistemas comienzan por identificar los informes y otras salidas que el sistema producirá. A continuación los datos específicos con éstos se señalan, incluyendo su localización exacta sobre el papel, la pantalla

de despliegue u otro medio. Usualmente, los diseñadores dibujan la forma o la visualización como la esperan cuando el sistema está terminado.

El diseño del sistema también describe los datos calculados o almacenados que se introducirán. Los grupos de datos individuales y los procedimientos de cálculo se describen con detalle. Los diseñadores seleccionan las estructuras de los archivos y los dispositivos de almacenamiento, como son discos magnéticos, cintas magnéticas o incluso archivos en papel. Los procedimientos que ellos escriben muestran cómo se van a procesar los datos y a producir la salida.

Los documentos que contienen las especificaciones de diseño utilizan muchas formas para representar los diseños, diagramas, tablas y símbolos especiales, algunos de los cuales el lector puede haber utilizado ya y otros que pudieran ser totalmente nuevos. La información del diseño detallado se pasa al grupo de programación para que pueda comenzar el desarrollo del software.

Los diseñadores son responsables de proporcionar a los programadores las especificaciones completas y escritas con claridad, que establezcan lo que debe hacer el software. Conforme comienza la programación, los diseñadores están pendientes para contestar preguntas, esclarecer ideas confusas y manejar los problemas que confronten los programadores cuando utilicen las especificaciones de diseño.

• Construcción (R)

Desarrollo del Software: Los desarrolladores del software pueden instalar o modificar; por ejemplo, software comercial que se haya comprado, o pueden escribir programas nuevos diseñados a la medida. La decisión de qué se va a hacer depende del costo de cada una de las opciones, el tiempo disponible para describir el software y la disponibilidad de programadores. En forma usual, en las grandes empresas los programadores de computadoras (o la combinación de analistas-programadores) son parte del grupo profesional permanente. Las compañías más pequeñas en donde los programadores permanentes no se han contratado, pueden obtener servicios externos de programación con base en un contrato.

Los programadores también son responsables de documentar el programa e incluir los comentarios que expliquen tanto cómo y por qué se utilizó cierto procedimiento conforme se codificó de cierta forma. La documentación es esencial para probar el programa y darle mantenimiento una vez que la aplicación se ha puesto en marcha.

• Pruebas (R)

Prueba de los sistemas: Durante la prueba, el sistema se utiliza en forma experimental para asegurar que el software no falle; es decir, que corra de acuerdo a sus especificaciones y a la manera que los usuarios esperan que lo haga. Se examinan datos especiales de prueba en la entrada del procesamiento y los resultados para localizar algunos problemas inesperados. Puede permitirse también a un grupo limitado de usuarios que utilice el sistema, de manera que los analistas puedan captar si tratan de utilizarlo en forma no planeadas. Es preferible detectar cualquier anomalía antes de que la empresa ponga en marcha el sistema y dependa de él.

En muchas compañías la prueba se lleva a cabo por personas diferentes a aquellos que los escriben en forma original; es decir si se utilizan personas que no conocen como se diseñaron ciertas partes de los programas, se asegura una mayor y más completa prueba, además de ser imparcial, lo que da a un software más confiable.

• Instalación (R)

Puesta en marcha: Cuando el personal de sistemas verifica y pone en uso el nuevo equipo, entrena al personal usuario; instala la nueva aplicación y constituye los archivos de datos que se necesiten, entonces el sistema está puesto en marcha.

De acuerdo con el tamaño de la empresa que empleará la aplicación y el riesgo asociado con su uso, los desarrolladores del sistema pueden escoger una prueba piloto para la operación del sistema solamente en un área de la compañía; por ejemplo, en un departamento o sólo con una o dos personas. A veces correrán en forma paralela tanto el sistema anterior como el nuevo para comparar los resultados de ambos; en otras situaciones, los desarrolladores pararán por completo el sistema anterior un día y al siguiente empezarán a utilizar el nuevo. Como se puede apreciar, cada estrategia para la puesta en marcha tiene sus méritos, que dependen de la situación del negocio considerado. Sin importar la estrategia para la puesta en marcha que se haya utilizado, los desarrolladores tendrán que asegurarse que el uso inicial del sistema esté libre de problemas.

Una vez instalada, con frecuencia la aplicación se utiliza por muchos años; sin embargo, tanto la empresa como los usuarios cambiarán, y el medio ambiente será diferente también a través del tiempo. Por lo tanto, la aplicación indudablemente necesitará mantenimiento; es decir, se harán cambios y modificaciones al software, y a los archivos o procedimientos para cubrir los requerimientos nuevos de los usuarios.

Los sistemas de la empresa y el medio ambiente de los negocios están en continuo cambio. Los sistemas de información deben mantenerse de la misma forma; es este sentido, la propuesta en marcha es un proceso continuo.

- **Mantenimiento (R)**

III. Administración de sistemas de información

1. Organización y administración

- **Administración de acceso a la información (R)**

A fin de efectuar una gestión coherente de los recursos y optimizar su adscripción a los proyectos hay que tener en cuenta un buen número de problemas de dirección y otros aspectos de interés. De entre ellos se puede resaltar los siguientes:

- ✓ El sistema de información está compuesto de un gran número de componentes o elementos técnicos (programas, equipo de comunicaciones, ordenadores personales, hardware del mainframe (computadora central) y software del sistema) que configuran un conjunto que puede considerarse un activo que hay que gestionar como cualquier otro.
- ✓ El sistema desempeña funciones y presta apoyo a procesos empresariales. Estos últimos están siempre sujetos a la posibilidad de cambiar, y el sistema que los apoya debe ser sensible a estas necesidades cambiantes.
- ✓ Hay una documentación que describe los sistemas y sus componentes, y que se constituye en punto de referencia cuando queremos modificarlos o mejorarllos; la documentación (especialmente para los sistemas clave para las operaciones) ha de estar siempre meticulosamente actualizada al mismo nivel de cambio que el sistema que describe.
- ✓ El personal de los sistemas de información necesita una formación y una actualización continuas. Por ello tenemos que organizar y asignar de forma óptima los escasos recursos disponibles de personal y establecer unas prioridades adecuadas para llevar a cabo los nuevos trabajos de desarrollo y mantenimiento.
- ✓ La seguridad de nuestros sistemas es una cuestión clave. Algunos de sus componentes son extremadamente vulnerables a los resentimientos y a los actos de mala fe, especialmente los datos que se almacenan en soportes intercambiables. Es esencial ser capaz de superar cualquier tipo de desastre.
- ✓ A medida que los sistemas se dispersan físicamente, se hace más relevante la problemática de su cohesión y compatibilidad: que componentes funcionan de forma específica con otros. Si hay algo que funciona en un computador, funcionará también con otro, o es probable que no lo haga. Cuantas versiones tenemos de un programa o de especificaciones de un sistema. A medida que se difunda y se diversifique el trabajo en red, se hará más vital la necesidad de mantenerla en buena forma.
- ✓ La capacidad del hardware informático instalado (como es el caso de los grandes mainframes y las redes) es muy compleja de estimar y de gestionar para dar un buen servicio. La gestión de la capacidad es una habilidad especial que no se encuentra fácilmente.
- ✓ El mantenimiento de los sistemas en uso es, una tarea considerable que suele consumir más de la mitad del presupuesto del SI.

La seguridad: es una de las cuestiones de mayor importancia en la actualidad a medida que los sistemas se hacen más globales, más dependientes de las redes y, por lo tanto, más accesibles para quienes tienen malas intenciones. La detección de estos es difícil, ya que los datos legibles para la máquina se pueden copiar con facilidad. La compleja naturaleza de los sistemas de información informatizados hace aún más difícil la detección de actividades delictivas. Casi siempre salen a la luz por accidente y muchas organizaciones prefieren no llevar los delitos informáticos al juzgado. Lo peor de todo es que es posible perpetuar un delito informático sin necesidad de entrar en ninguna oficina o dependencia corporativa, cualquier ordenador que esté conectado a la red telefónica pública incluso a una red privada, pero con transmisiones externas, corre el riesgo de sufrir accesos no autorizados.

La empresa moderna depende tanto de su tecnología de la información que las presiones a corto plazo pueden ser abrumadoras. La gestión de los sistemas de información operativos se realiza sobre la marcha sobre la marcha,

según se van presentando las pequeñas crisis del funcionamiento cotidiano. Sin embargo, los intereses operativos a largo plazo, tales como la seguridad y la necesidad de recuperarse de un desastre importante, se suelen dejar en segundo plano, poniendo a toda la organización innecesariamente en una situación arriesgada. Este es otro aspecto sobre el que la dirección debe ejercer su habilidad para equilibrar lo global con lo particular.

Comparación con otras situaciones en que se gestionan recursos: gran parte del reto de la gestión de recursos en el terreno de los sistemas de información se puede considerar como similar a lo que ocurre en cualquier otra parte. Consideremos la gestión de la producción. Las mismas actitudes y habilidades que tienen éxito en la gestión de una sección de almacén que contenga componentes de valor será probablemente de utilidad para administrar el “inventario” de Sistemas de Información con igual éxito. Pero la naturaleza del producto que estamos viendo es única y eso significa que:

- ✓ Dado que es abstracta, no hay una base común concreta de cuáles son las diferentes partes de una organización que pueden compartir fácilmente sus puntos de vista sobre lo que es un sistema.
- ✓ Es posible modificar el software muy rápidamente, y no dejar evidencia alguna de ello;
- ✓ Es muy compleja, el sistema más sencillo se basa en miles de componentes elementales y los grandes sistemas se basan en millones de ellas;
- ✓ El contexto dentro del que concebimos y construimos los sistemas puede cambiar rápidamente.

- **Asignación y administración de recursos para el sistema (E)**

- **Políticas de respaldo y seguridad del sistema (E)**

- **Soporte a los usuarios del sistema (R)**

GRAFICACIÓN, INTELIGENCIA ARTIFICIAL E INTERACCIÓN HUMANO-COMPUTADORA

A. Graficación

I. Aspectos generales

1. Introducción a la graficación

- **Definición y objetivos de la graficación (R)**

Definición y objetivos de la graficación.

El uso adecuado y provechoso de la tecnología han hecho de la computadora una dispositivo poderoso para producir imágenes en forma rápida y económica. Actualmente en todas las áreas es posible aplicar gráficas por computadora con algún objetivo, por ello se ha generalizado la utilización de gráficas por computadora. De igual modo las gráficas por computadora se utilizan de manera rutinaria en diversas áreas, como en la ciencia, ingeniería, empresas, industria, gobierno, arte, entretenimiento, publicidad, educación, capacitación y presentaciones gráficas.

Breve historia.

Las computadoras se han convertido en una herramienta poderosa para producir imágenes, interpretar información o mejorar la calidad de visualización de las mismas en forma rápida y económica. Debemos aclarar que los métodos que se utilizan en las gráficas por computadora y en procesamiento de imágenes tienen características similares pero no son iguales es decir, las dos áreas realizan, en forma fundamental operaciones distintas. Las herramientas para graficación por computadoras, se utilizan para crear una o mas imágenes. Por otro lado, en el procesamiento de imágenes se aplican técnicas para modificar o interpretar imágenes existente como fotografías y rastreos de televisión. El interés por los métodos de tratamiento o procesamiento digital de imágenes deriva de dos áreas principales:

Objetivo

Estudiar los principios y metodologías necesarias para la representación, manipulación y despliegue de figuras e imágenes en dos y tres dimensiones, considerando los dispositivos de hardware con características específicas para procesos de graficación.

• **Aplicaciones (multimedios, ambientes inmersivos, visualización, cartografía, CAD/CAM, etcétera) (R)**

Diseño asistido por computadora

El método de diseño asistido por computadora, conocido por lo general como CAD (Computer Assisted Design), ahora se utilizan de forma rutinaria en el diseño de construcciones, automóviles, aeronaves, embarcaciones, naves espaciales, computadoras, telas y muchos productos. En el caso de algunas aplicaciones de diseño, los objetos se despliegan primero en forma de armazón mostrando la forma general y sus características internas. Los despliegues del armazón permiten ver a los diseñadores con rapidez los efectos de ajustes interactivos para diseñar formas. Regularmente, los paquetes de software de aplicaciones de CAD ofrecen al diseñador un entorno con ventanas múltiples; estas diversas ventanas desplegables muestran secciones amplificadas de vistas de diferentes objetos. Los circuitos y las redes para comunicaciones, abastecimientos de agua y otros servicios públicos se construyen a través de la colocación repetida de algunas formas gráficas. Las formas usadas en un diseño representan los diversos componentes del circuito o de la red. Con el paquete de diseño se ofrecen formas estándar para circuitos eléctricos, electrónicos y lógicos. Para otras aplicaciones, un diseñador puede crear símbolos personalizados empleados necesariamente para construir la red o el circuito. Así, se diseña el sistema colocando sucesivamente los componentes en el esquema, con el paquete de gráficas ofreciendo de manera automática las conexiones entre los componentes. Esto permite al diseñador experimentar rápidamente con esquemas de circuitos alternativos para reducir al mínimo el número de componentes o el espacio para el sistema. Con frecuencia se utilizan las animaciones en las aplicaciones del CAD. Las animaciones en tiempo real que emplean despliegues de armazones en un monitor de video son útiles para probar el comportamiento de un vehículo o un sistema. Cuando no desplegamos objetos con superficies presentadas, pueden realizarse con rapidez los cálculos correspondientes a cada segmento de la animación para así crear un movimiento suave de tiempo real en la pantalla. Igualmente, los despliegues de armazones permiten al diseñador ver el interior del vehículo y observar el comportamiento de los componentes internos durante el movimiento. Las animaciones en entornos de realidad virtual se utilizan para determinar la forma como influyen ciertos movimientos en los operadores de vehículos. Por ejemplo, el operador de un tractor con ayuda de un dispositivo montado sobre la cabeza que presenta una vista estereoscópica del cucharón del cargador frontal o del retro excavador, manipula los controles como si se encontrara en el asiento del tractor. Esto permite al diseñador explorar diversas posiciones del cucharón o del retro excavador que pudieran obstruir la visión del operador. Cuando los diseños de objetos están completos o casi completos, se aplican modelos de iluminación realista y presentaciones de superficies para producir despliegues mostrando la apariencia del producto final. También se crean despliegues realistas para la publicidad de automóviles y otros vehículos mediante efectos especiales de iluminación y escenas de fondo. El proceso de manufactura también se asocia con la descripción por computadora de objetos diseñados para automatizar la construcción del producto. Por ejemplo, se puede convertir el esquema de un tablero de circuitos en una descripción de los procesos individuales necesarios para elaborar el esquema. Algunas partes mecánicas se fabrican por medio de la descripción de cómo se deben formar las superficies con herramientas. Luego, se ajustan las herramientas controladas de manera numérica para fabricar la parte de acuerdo con estos planos de construcción. Los arquitectos utilizan métodos gráficos interactivos para proyectar plantas arquitectónicas donde se muestra la disposición de habitaciones, ventanas, escaleras, anaqueles, barras de cocina y otras características de la construcción. A partir del despliegue del plano de una construcción en un monitor de video, un diseñador eléctrico puede experimentar con instalaciones para cableado, conexiones eléctricas y sistemas de alarma de incendios. Del mismo modo, aplicando paquetes para el esquema de instalaciones se determina la utilización del espacio en una oficina o en una planta de fabricación. Despliegues realistas de diseños arquitectónicos permiten a los arquitectos y a sus clientes estudiar la apariencia de una construcción particular o de un grupo de ellas, como un campus universitario o un complejo industrial. Con los sistemas de realidad virtual, los diseñadores pueden simular un "recorrido" por las habitaciones o alrededor de construcciones para apreciar mejor el efecto general de un diseño particular. Además de presentar despliegues de fachadas realistas, los paquetes de CAD para arquitectura ofrecen medios para experimentar con planos interiores tridimensionales y la iluminación. Muchas otras clases de sistemas y productos se diseñan usando ya sea paquetes de CAD generales o software de CAD desarrollado en forma especial.

Arte por computadora

Los métodos de gráficas por computadora se utilizan en forma generalizada tanto en aplicaciones de bellas artes como en aplicaciones de arte comercial. Los artistas utilizan una variedad de métodos computacionales, incluyendo hardware para propósitos especiales, programas artísticos de brocha de pintar del artista (como Lumena), otros

paquetes de pintura (como PixelPaint y SuperPaint), software desarrollado de manera especial, paquetes de matemática simbólica (como Mathematica), paquetes de CAD, software de edición electrónica de publicaciones y paquetes de animaciones que proporcionan los medios para diseñar formas de objetos y especificar movimientos de objetos. La idea básica del programa paintbrush (brocha de pintar) permite a los artistas "pintar" imágenes en la pantalla de un monitor de video. En realidad, la imagen se pinta por lo general de manera electrónica en una tableta de gráficas (digitalizador) utilizando un estilete, el cual puede simular diferentes trazos, anchuras de la brocha y colores. Los creadores de bellas artes emplean diversas tecnologías de computación para producir imágenes. Con el propósito de crear pinturas el artista utiliza una combinación de paquetes de modelado tridimensional, diagramación de la textura, programas de dibujo y software de CAD. En un ejemplo de "arte matemático" un artista utilizó una combinación de funciones matemáticas, procedimientos fractales, software de Matemática, impresoras de chorro de tinta y otros sistemas con el fin de crear una variedad de formas tridimensionales y bidimensionales, al igual que pares de imágenes estereoscópicas. Otro ejemplo arte electrónico creado a partir de relaciones matemáticas es la obra de un compositor que está diseñada en relación con las variaciones de la frecuencia y otros parámetros en una composición musical para producir un video el cual integra patrones visuales y auditivos. También se aplican estos métodos en el arte comercial para crear logotipos y otros diseños, distribuciones de página que combinan texto y gráficas, anuncios publicitarios por televisión y otras áreas. Para muchas aplicaciones de arte comercial (y películas, al igual que otras aplicaciones), se emplean técnicas fotorrealistas para presentar imágenes de un producto. Las animaciones también se utilizan con frecuencia en publicidad y los comerciales de televisión se producen cuadro por cuadro, donde cada cuadro del movimiento se presenta y graba como un archivo de imagen. Se simula el movimiento al mover ligeramente las posiciones de los objetos con respecto a las del cuadro anterior. Una vez presentados todos los cuadros de la secuencia de animación, se transfieren a película o se almacenan en un búfer de video para hacer una reproducción. Las animaciones en película requieren 24 cuadros por cada segundo de la secuencia de animación. Si se reproduce en un monitor de video, se requieren de 30 cuadros por segundo. Un método común de gráficas que se utilizan en muchos comerciales es la transformación (morphing), donde se transforma un objeto en otro (metamorfosis). En televisión se ha empleado para transformar una lata de aceite en un motor de automóvil, un automóvil en un tigre, un charco en una llanta y el rostro de una persona en otro.

Entretenimiento

Es muy común utilizar métodos de gráficas por computadora para producir películas, videos musicales y programas de televisión. En ocasiones, se despliegan sólo imágenes gráficas y otras veces, se combinan los objetos con los actores y escenas en vivo. Por ejemplo, en una escena gráfica creada para la película Start Trek - The Wrath of Khan, se dibujan en forma de armazón el planeta y la nave espacial y se sombrean con métodos de presentación para producir superficies sólidas. Al igual que pueden aparecer personas en forma de armazón combinadas con actores y una escena en vivo. Los videos musicales aprovechan las gráficas de muchas maneras, se pueden combinar objetos gráficos con acción en vivo, o se pueden utilizar técnicas de procesamiento de imágenes para producir una transformación de una persona o un objeto en otro (morphing).

Visualización

Científicos, ingenieros, personal médico, analistas comerciales y otros con frecuencia necesitan analizar grandes cantidades de información o estudiar el comportamiento de ciertos procesos. Las simulaciones numéricas efectuadas en supercomputadoras a menudo producen archivos de datos que contienen miles e incluso millones de valores de datos. De modo similar, cámaras vía satélite y otras fuentes acumulan grandes archivos de datos más rápido de lo que se pueden interpretar. El rastreo de estos grandes conjuntos de números para determinar tendencias y relaciones es un proceso tedioso e ineфicaz. Pero si se convierten los datos a una forma visual, es frecuente que se perciban de inmediato las tendencias y los patrones. Por lo regular, la producción de representaciones gráficas para conjuntos de datos y procesos científicos de ingeniería y de medicina se conoce como visualización científica.

Clasificación y alcances del software:

- ✓ *Alias Wavefront's Maya*: Es quizás el software más popular en la industria, por lo menos hasta 2003. Es utilizado por muchos de los estudios de efectos visuales más importantes en combinación con Renderman, el motor de rénder fotorrealista de Pixar. Última versión a Septiembre de 2003: Maya 5.
- ✓ *Discreet's 3D Studio Max*: Originalmente escrito por Kinetix (una división de Autodesk) como el sucesor de 3D Studio. Kinetix luego se fusionó con la última adquisición de Autodesk, Discreet Logic. La versión actual (a Febrero de 2003) es la 5.1. Es el líder en el desarrollo de 3D en la industria de juegos y usuarios hogareños.
- ✓ *Newtek's Lightwave 3D*: Fue originalmente desarrollado por Amiga Computers a principios de la década de 1990. Más tarde evolucionó en un avanzado y muy usado paquete de gráficos y animación 3D. Actualmente

disponible para Windows, Mac OS y Mac OS X. La versión actual es 7.5. El programa consiste en dos componentes: Modelador y Editor de escena. Es el favorito entre los entusiastas, y es utilizado en muchas de las mayores productoras de efectos visuales como Digital Domain.

- ✓ Avid's Softimage XSI: El contrincante más grande de Maya. En 1987, Softimage Inc, una compañía situada en Montreal, escribió Softimage|3D, que se convirtió rápidamente en el programa de 3D más popular de ese período. En 1994, Microsoft compró Softimage Inc. y comenzaron a reescribir SoftImage|3D para Windows NT. El resultado se llamó Softimage|XSI. En 1998 Microsoft vendió Softimage a [Avid](#). La versión actual (hasta mediados de 2003) es 3.5.

II. Hardware para graficación

1. Medios

- Dispositivos de entrada y de salida (características y ejemplos) (R, E, RP)

Teclados: Un teclado alfanumérico en un sistema de gráficas se utiliza sobre todo como un dispositivo para capturar cadenas de texto, es eficiente para capturar datos no gráficos. Las teclas de control de cursor y las de función son características regulares en los teclados de uso común. Las teclas de función permiten que los usuarios capturen operaciones empleadas con frecuencia con un solo golpe de tecla y las teclas de control de cursor pueden emplearse para seleccionar objetos desplegados o posiciones de coordenadas al poner en posición el cursor de la pantalla. Otros tipos de posicionamiento del cursor, como la bola palmar o la palanca de control, se incluyen en algunos teclados. A menudo se incluye un teclado numérico en el teclado para la captura rápida de datos numéricos.

Bola palmar y esfera de control: Como su nombre indica, la bola palmar es una bola que se puede hacer girar con los dedos o la palma de la mano, para producir movimiento en el cursor de la pantalla. Con potenciómetros que se conectan a la bola, se mide la cantidad y la dirección de la rotación. Este dispositivo es de posicionamiento bidimensional. Una esfera de control ofrece seis grados de libertad, y en realidad no se mueve.

Guante de datos: El guante está construido con una serie de sensores que detectan los movimientos de la mano y los dedos. Se emplea un acoplamiento electromagnético entre antenas transmisoras y antenas receptoras para proporcionar información acerca de la posición y la orientación de la mano. Cada una de las antenas transmisoras y receptoras puede estar estructurada con un conjunto de tres retículas mutuamente perpendiculares que forman un sistema de coordenadas cartesianas. La entrada del guante de datos puede emplearse para poner en posición y manipular objetos en una escena virtual.

- Arquitecturas para graficación (aceleradores, buses, coprocesadores, etcétera) (R, E, RP)

III. Conceptos, técnicas y algoritmos

1. Modelado

- Transformaciones y proyecciones en dos y tres dimensiones (R, E, RP)

Transformaciones

Las transformaciones, que matemáticamente se relacionan con multiplicaciones de matrices, incluyen operaciones de modelado, vista y proyección. En este grupo de operaciones se encuentran las rotaciones, traslaciones, escalados, reflexiones, proyecciones ortográficas y proyecciones en perspectiva. Habitualmente se utiliza una combinación de varias transformaciones para representar una escena.

Gráficos 3D

Imágenes que representan simulaciones del espacio tridimensional.

- ✓ El objetivo básico de la graficación 3D es provocar la sensación visual de profundidad.
- ✓ La principal barrera del desarrollo de gráficos 3D consiste en representar escenas tridimensionales en un dispositivo bidimensional

- Representación de objetos primitivos y objetos compuestos: líneas, curvas, superficies paramétricas (R, E, RP)

- Modelos implícitos y modelos volumétricos (R)

2. Síntesis de imagen

- Superficies ocultas (R, E, RP)

- Sombreado, iluminación local, iluminación global (*ray-tracing* y radiosidad) y textura (R)

Como se ha podido apreciar, dicha actividad se circunscribe en un orden intelectual, técnico y manual, aspectos indivisibles a la hora de planificar sus trabajos.

Aunar criterios establecidos desde el ámbito cultural, para confrontarlo con los aportes tecnológicos y proyectarlos a un objetivo en común, serán los requisitos básicos, a fin de establecer un nuevo aporte en esta materia y delimitar sus alcances.

A su vez, es preciso tomar conciencia de que cada programa específico de diseño requiere de una formación integral, que ayudan a comprender la esencia misma de esta disciplina, con nuevos criterios que consolidan, aún más, su autonomía.

Es posible establecer una analogía con las principales relaciones entre las funciones operativas y un uso consensual del sistema.

PARATEXTO VIRTUAL	PARATEXTO GRÁFICO	VISIÓN CRÍTICA
Fuentes	Aplicación de estilo, tamaño y nombre de las fuentes, así como también los efectos.	Destacar títulos, palabras claves, organización discursiva de citas, etc.
Bordes y Sombreados	Permiten la utilización de cuadros sobre la base de estilos predefinidos.	Remarcando carátulas, separación de párrafos, encabezados, organización de tablas, etc.
Viñetas / números	Aplicación de fuentes especiales y símbolos.	Destacar elementos en una dispersión genérica, o bien, en una sucesión jerárquica.
Columnas	Se establecen como espacios interdependientes (ancho y separación) para la lectura del texto.	Para el sistema occidental, un texto se lee desde la primera columna de arriba hacia abajo y se pasa a la siguiente hasta llegar al final.
Tablas	Generan espacios independientes de identificación global.	El usuario establece un modo de organización discursiva independiente para su posterior aplicación de formatos.
Títulos/ rótulos	Genera una aplicación práctica de convenciones que permiten cambiar fuentes y posición de títulos básicos.	Los títulos se pueden identificar en un orden jerárquico (principal y accesorios)
Índice	Mediante la aplicación de títulos, se puede generar un índice automático.	Sugieren dos modelos analíticos de índice: el onomástico y el general.
Dibujos	Se pueden incluir dibujos en un texto de párrafo o de imagen.	Se sugieren los modos ClipArt establecidos como metaarchivos de Windows (.wmf).
Imágenes	Selección de imágenes de ClipArt y personales para su inclusión en el texto (formatos: JPG, BMP, etc.)	Una imagen tiene expresa significación con el contenido textual.
Otras funciones Complementarias	Se establecen las funciones de ortografía, sinónimos, guardar, salir, copiar y pegar, dividir ventanas, entre otras.	Permiten al usuario una perfección de su actividad y el ahorro de tiempo (copiados intensos de párrafos, reproducción de imágenes, organizar una disposición de varios archivos, trabajar alternadamente, etc.)
Código ascii	Ubicación de la tecla Alt con una	Incorporación de caracteres especiales: ñ, Ñ,

	combinación numérica. Se utiliza en el caso de no hallar algún signo o letra en el teclado.	¿,¡, etcétera.
Encabezados y Pie de página	Habilita un nuevo espacio de escritura (fuera del área de página).	Permite crear un texto repetitivo para otras páginas que organizan un contenido sistemático con funciones automáticas (rótulos, números de página, fecha y hora, etc.)
Número de página	Incluye en las páginas una sucesión de números en forma automática.	Disposición de los elementos, tanto en el encabezado como en el pie de página.
Texto de párrafo	Organización de un texto base de acuerdo con una estructura convencional.	Incorporar guiones, sangrías, alineaciones, etc.
Texto artístico	Incorporar un texto base con relación a un efecto imagen	Cambios globales de disposición y de efectos. Ejemplo: WordArt.
Letra capital	Ampliar la primera letra de un párrafo.	Cambio de fuente y su ubicación en el párrafo.
Guiones	Distribución de funciones básicas y de efectos.	Galería de efectos Preset y Scripts.
Colores y Rellenos	Aplicación de colores básicos y estructuras de rellenos interactivos.	Modos de enunciación disponibles: tramas, degradados y formas complejas de combinación.
Efectos comunes de texto	Escribir un texto de párrafo con modos de visión unívoca y alternada.	Se pueden establecer: negrita, subrayada, tachada, subíndice, superíndice, etcétera. Por ejemplo: m ² , H ₂ O, texto, entre otros.
Efectos especiales	Organización de elementos sobre la base de un determinado modo de visión.	Incorporar efectos en 2 y 3 dimensiones, de imagen, etc.

3. Técnicas de animación

Técnicas de animación

La animación es la simulación de un movimiento, creada por la muestra de una serie de imágenes o cuadros. Un ejemplo de esto son las caricaturas, que pertenecen a la animación tradicional.

Una característica importante de la animación por computadora es que nos permite crear escenas “realmente” tridimensionales. Esto quiere decir que, a diferencia de la animación dibujada a mano, en una escena animada por computadora es posible cambiar el ángulo de la cámara y con esto, ver otra parte de la escena.

Algunas aplicaciones: típicas de la animación generada por computadora son el entretenimiento (películas y dibujos animados), publicidad, estudios científicos y de ingeniería, capacitación y educación. A pesar de que tendemos a considerar que la animación implica movimientos de objetos, el término animación por computadora por lo regular se refiere a cualquier secuencia de tiempo de cambios visuales en una escena. Además de cambiar las posiciones de los objetos con traslaciones y rotaciones, una animación por computadora podría desplegar variaciones de tiempo en el tamaño, el color, la transparencia o la textura de la superficie de los objetos. Con frecuencia, las animaciones publicitarias realizan la transición de la forma de un objeto en otra.

Este planteamiento estándar para dibujos animados se aplica también en otras aplicaciones de la animación, aunque hay muchas aplicaciones especiales que no siguen esta secuencia. Las animaciones por computadora de tiempo real que producen os simuladores de vuelo, por ejemplo, despliegan secuencias de movimiento en respuesta a las especificaciones de los controles de la aeronave. Y aplicaciones de visualización se generan mediante las soluciones de los modelos numéricos. Para una animaci6ncuadro por cuadro, se genera y almacena por separado cada cuadro de la escena. Luego, pueden grabarse los cuadros en la película desplegarse de manera consecutiva en modo de "pista de tiempo real".

El guión es una descripción de la acción. Define la secuencia de movimiento como un conjunto de eventos básicos que deben ocurrir. De acuerdo con el tipo de animación que se debe producir, el guión podría consistir en un conjunto de borradores o ser una lista de las ideas básicas para el movimiento.

En sistemas de rastreo, podemos generar animación de tiempo real en aplicaciones limitadas al utilizar operaciones de rastreo. Como estudiaremos un método simple para la traslación en el plano de xy es transferir un bloque rectangular de valores de pixel de una posición a otra. También es sencillo realizar rotulaciones bidimensionales en múltiples de 90°, aunque podemos girar bloques rectangulares de pixeles a través de ángulos arbitrarios al emplear procedimientos de antillas.

Lenguajes de animación por computadora

El diseño y el control de las secuencias de animación se manejan con un conjunto de rutinas de animación. Con frecuencia, para programar las secuencias de animación se utiliza un lenguaje de uso común, como C, Lisp, Pascal o FORTRAN, pero se han desarrollado varios lenguajes de animación especializados. Las funciones de animación incluyen un editor de gráficas, un generador de cuadros clave, un generador de cuadros intermedios y rutinas gráficas estándar. El editor de gráficas nos permite diseñar y modificar las formas de los objetos al utilizar superficies de spline, métodos de geometría sólida constructiva u otros esquemas de representación. Una tarea común en la especificación de una animación es la descripción de la escena, la cual incluye el posicionamiento de objetos y fuentes de luz, define los parámetros fotométricos (intensidades de las fuentes de luz y propiedades de la iluminación de la superficie) y establece los parámetros de la cámara (posición, orientación y características del lente). Otra función estándar es la especificación de la acción. Esta función comprende la distribución de las trayectorias de movimiento para los objetos y la cámara. Se requieren las rutinas gráficas estándar, transformaciones de vista y de perspectiva, así como transformaciones geométricas, a fin de generar movimientos de los objetos como una función de especificaciones de aceleraciones o de trayectoria cinemática, identificación de superficies visibles y las operaciones de presentación de superficies.

Los sistemas de cuadro clave son lenguajes especializados para animación que se diseñan sólo para generar los cuadros intermedios a partir de los cuadros clave que el usuario especifica.

Transformación (Morphing)

La conversión de las formas de los objetos de una a otra forma se denomina transformación (morphing), que es un término para expresar una metamorfosis. Se pueden aplicar métodos de transformación en cualquier movimiento o transición que implique un cambio de forma. Al dar dos cuadros clave para la transformación de un objeto, primero ajustamos la especificación del objeto en uno de los cuadros, de modo que el número de aristas de polígono (o el número de vértices) sea igual para los dos cuadros.

• Animación por cuadros clave (keyframes) (R, E, RP)

Un cuadro clave es un diseño detallado de la escena en un momento determinado de la secuencia de animación. En cada cuadro clave se sitúa cada objeto de acuerdo con el tiempo para ese cuadro. Algunos cuadros clave se seleccionan en posiciones extremas en la acción; otras se espacian de modo que el intervalo de tiempo entre cuadros clave no sea muy extenso. Para movimientos complicados, se especifican más cuadros clave que para movimientos simples con variaciones lentas.

Los cuadros intermedios son los cuadros intermedios entre los cuadros clave. El número de cuadros intermedios que se necesita se determina por los medios que se van a utilizar para desplegar la animación. La película requiere 24 cuadros por segundo. Por lo regular, los intervalos de tiempo para el movimiento se establecen de modo que haya de tres a cinco cuadros de intermedio por cada par de cuadros clave.

Sistemas de cuadro-clave

Generamos cada conjunto de cuadros intermedios a partir de la especificación de dos (o más) cuadros clave. Las trayectorias del movimiento se pueden dar con una descripción cinemática como un conjunto de curvas spline o los movimientos pueden ser con base en las características físicas al especificar las fuerzas que actúan sobre los objetos que se deben animar.

Para escenas complejas, separamos los cuadros en componentes u objetos individuales que se llaman diapositivas de celuloide, un término para hacer referencia a la animación de dibujos. Dadas las trayectorias de la animación, podemos interpolar las posiciones de objetos individuales entre cualquiera de dos tiempos. Con transformaciones de objetos complejas, las formas de los objetos pueden cambiar con el tiempo. Como ejemplos, podemos citarlas telas, rasgos faciales, detalles amplificados, formas en evolución, objetos que explotan o se desintegran y la transformación de un objeto en otro. Si todas las superficies se describen con enlaces de polígonos, entonces el número de aristas por polígono puede variar de un cuadro al siguiente. Por tanto, el número total de segmento de línea puede cambiar en cuadros diferentes.

- **Animación por cinemática inversa, por captura de movimiento, basada en física y mediante agentes autónomos (R)**

GRAFICACIÓN, INTELIGENCIA ARTIFICIAL E INTERACCIÓN HUMANO-COMPUTADORA

C. Interacción humano-computadora

I. Aspectos generales

1. Introducción a la interacción humano-computadora

- **Orígenes y definiciones (R)**

IHC se define como el entendimiento, diseño, evaluación e implementación de sistemas interactivos para el uso por humanos.

“IHC es el diseño de sistemas computacionales que apoyan a personas para que puedan llevar a cabo sus actividades de manera eficiente y segura.” [Preece et al., 1994]

“IHC es el estudio y práctica de usabilidad. Es sobre el entendimiento, y creación de software y otras tecnologías que la gente querrá utilizar, será capaz de utilizar y encontrará efectivo al usarla.” [Carroll, 2002]

“No debemos de enfocarnos tanto en la interacción humano-computadora, sino más en la **interacción entre humanos a través de la computadora.**” [comentario de Terry Winograd]

La IHC se lleva a cabo en un entorno (contexto) social y organizacional.

Diferentes tipos de aplicaciones se requieren para diferentes propósitos y se necesita cuidado para dividir las tareas entre humanos y máquinas, asegurando que esas actividades creativas y no-repetitivas sean asignadas a los humanos mientras que las otras sean asignadas a las máquinas.

Son necesarios conocimientos de las habilidades psicológicas y fisiológicas, y son especialmente importantes sus limitaciones.

II. Aspectos humanos

1. Características, capacidades y limitaciones cognitivas y perceptuales

Tipos de factores que intervienen en ihc:

- ✓ Hay varios factores a considerar en la HCI, no solo los computacionales.
- ✓ Si no se consideran los factores en su totalidad, un buen producto de software puede resultar ser finalmente inútil.
- ✓ Los factores son de diverso tipo, pero se relacionan.
- ✓ Hay factores relativos al usuario, relativos a su ambiente y relativos al desempeño del sistema en general.
- ✓ Factores relativos al ambiente.
- ✓ Factores relativos al usuario.
- ✓ Factores relativos a los beneficios de la organización.

Factores de las capacidades y procesos cognitivos del usuario:

- ✓ Motivación
- ✓ Gusto a las actividades

- ✓ Satisfacción
- ✓ Experiencia
- ✓ Capacidad de aprendizaje

Factores de la interfaz del usuario:

- ✓ Dispositivos de entrada
- ✓ Dispositivos de salida
- ✓ Estructura de los diálogos humano-computadora
- ✓ Uso de colores
- ✓ Iconos
- ✓ Comandos
- ✓ Gráficas
- ✓ lenguaje natural
- ✓ Realidad virtual
- ✓ Multimedios

Factores referentes a las limitaciones:

- ✓ Limitantes de Costo
- ✓ Limitantes de Tiempo
- ✓ Limitantes de Personal
- ✓ Limitantes de Equipo
- ✓ Limitantes de Espacio físico.

• **Memoria y percepción (R, E)**

Memoria:

La memoria es el reflejo de lo que existió en el pasado. Este reflejo está basado en la formación de conexiones temporales suficientemente firmes (fijación en la memoria) y en su actualización o funcionamiento en el futuro (reproducción y recuerdo)

Modelo De La Memoria Humana:

Memoria Sensorial:

- ✓ Información del mundo externo se registra por elementos de memoria sensorial específicos para cada uno de **nuestros sentidos** (tactil, visual, auditivo).
- ✓ Dichos elementos de memoria sensorial específicos son como buffers que mantienen una representación **directa** de información percibida.
- ✓ Se mantiene durante **décimas de segundo**.

Memoria a Corto Plazo:

- ✓ Se le conoce también como **memoria de trabajo**.
- ✓ Lugar donde se procesa el contenido proveniente de la memoria sensorial y que puede pasar a la memoria a largo plazo.
- ✓ Es el área de trabajo temporal para que se lleven acciones de comparación y selección de respuesta.
- ✓ Tiene limitaciones de tiempo y espacio.
- ✓ El **número mágico 7 +- 2** identificado por Miller en 1956.

Memoria a Largo Plazo:

- ✓ La información que llega se considera permanente.
- ✓ Implica una estructura de **representación de conocimientos**.

Memoria de Episodios:

Memoria referente a eventos, en **una secuencia y un contexto**.

Memoria Semántica:

Necesaria para la **interpretación de símbolos**.

Percepción:

La percepción es un componente del conocimiento en donde el sujeto aplica el interactuar con el mundo objetivo al percibirlo. Por lo que la percepción está ligada al lenguaje y es entonces un elemento básico en el desarrollo cognitivo.

Psicología Cognitiva.

- ✓ Capacidades y limitaciones del usuario.
- ✓ Interacción en grupo (distributed cognition).
- ✓ Estructura y funciones de la organización.
- ✓ Apoyo a la toma de decisiones.

Procesos cognitivos.

"Modos en que trabajan nuestras mentes, como influyen en nuestra comprensión del mundo, recuerdos, percepciones, pensamientos, emociones y motivaciones guían nuestras acciones". Actuaremos con nuestra base de cómo creemos que es el mundo.

La perspectiva cognitiva

El marco dominante que ha caracterizado la IHC ha sido cognitivo.

La cognición es el proceso a través del cual se adquiere conocimiento incluye: entender, recordar, razonar, atención estar consciente, adquirir capacidades, crear nuevas ideas.

También se le conoce como los procesos referentes a nuestra interacción con los objetos del mundo que nos rodea y el cómo conocemos. Incluye:

- ✓ Comprensión
- ✓ Memoria
- ✓ Razonamiento
- ✓ Atención
- ✓ Conciencia (estar consciente de algo)
- ✓ Adquirir habilidades
- ✓ Crear nuevas ideas

Uno de los principales objetivos de IHC ha sido entender y representar como humanos interactúan con los sistemas en términos de cómo se transmite el conocimiento entre los dos

¿Cómo los humanos logran sus objetivos?

Esta actividad orientada a metas comprende diferentes tareas cognitivas, entre ellas el procesamiento de la información.

• Ergonomía (R, E, RP)

La palabra *ergonomía* se deriva de las palabras griegas "ergos", que significa trabajo, y "nomos", leyes; por lo que literalmente significa "leyes del trabajo", y podemos decir que es la actividad de carácter multidisciplinario que se encarga del estudio de la conducta y las actividades de las personas, con la finalidad de adecuar los productos, sistemas, puestos de trabajo y entornos a las características, limitaciones y necesidades de sus usuarios, buscando optimizar su eficacia, seguridad y confort.

La ergonomía es básicamente una tecnología de aplicación práctica e interdisciplinaria, fundamentada en investigaciones científicas, que tiene como objetivo la optimización integral de Sistemas Hombres-Máquinas, los que estarán siempre compuestos por uno o más seres humanos cumpliendo una tarea cualquiera con ayuda de una o más "máquinas" (definimos con ese término genérico a todo tipo de herramientas, máquinas industriales propiamente dichas, vehículos, computadoras, electrodomésticos, etc.). Al decir optimización integral queremos significar la obtención de una estructura sistémica (y su correspondiente comportamiento dinámico), para cada conjunto interactuante de hombres y máquinas, que satisfaga simultánea y convenientemente a los siguientes tres criterios fundamentales:

- ✓ Participación: de los seres humanos en cuanto a creatividad tecnológica, gestión, remuneración, confort y roles psicosociales.
- ✓ Producción: en todo lo que hace a la eficacia y eficiencia productivas del Sistema Hombres-Máquinas (en síntesis: productividad y calidad).
- ✓ Protección: de los Subsistemas Hombre (seguridad industrial e higiene laboral), de los Subsistemas Máquina (siniestros, fallas, averías, etc.) y del entorno (seguridad colectiva, ecología, etc.).

III. Interfaces

1. Diseño de interfaces

• Metodologías de desarrollo (R)

Hasta este punto habrá quedado claro que el rol de IHC en el de diseño de sistemas es el de mejorar la calidad de la interacción entre humanos y computadoras. Para ello se requiere de la aplicación sistemática de conocimiento sobre las metas humanas, capacidades y limitaciones junto con el conocimiento sobre las capacidades y limitaciones de la tecnología. Este conocimiento debe relacionarse al entendimiento de los aspectos sociales, organizacionales y físicos del entorno de trabajo del usuario.

El reto para los que desean diseñar computadoras o sistemas que la gente pueda usar consiste en:

- ✓ Saber cómo hacer la transición de los que se puede hacer (funcionalidad) a cómo debe hacerse para cubrir los requerimientos del usuario (usabilidad) en el ambiente de trabajo.
- ✓ En el nivel físico esto significa seleccionar los dispositivos adecuados de entrada y salida, determinar el mejor esquema de interacción (formas, lenguaje natural, GUI's, etc.).

• Herramientas y técnicas para la construcción de interfaces (E, RP)

El proceso de diseño de la interfaz :

- ✓ El origen del diseño es una necesidad no satisfecha.
- ✓ Esa necesidad no es necesariamente la de los usuarios, sino a veces es la necesidad de los administradores.
- ✓ El diseño es un proceso que implica diversos intereses y negociación.
- ✓ El diseño es una actividad social.

• Principios y guías de diseño (E)

• Usabilidad (E, RP)

Un concepto clave en la HCI: Desarrollar sistemas fáciles de aprender y fáciles de usar.

Según Donald Norman , hay dos principios claves para una buena interacción humano-computadora: Visibilidad y Provisión.

- ✓ *Visibilidad (visibility):* Los componentes de una interfaz deben:
 - Ser visibles, pero no sobrecargar al usuario.
 - Tener un buen mapeo con los efectos que producen.
- ✓ *Feedback:* relación entre los objetivos del usuario, las acciones que debe hacer y los resultados.
- ✓ *Provisión (affordance):* Los componentes de una interfaz deben proveer de manera natural su funcionalidad. Se refiere a las propiedades de los objetos que sugieren su uso.

2. Parádigmas de interacción

• De lenguajes de comandos a manipulación directa (R)

• Ambientes virtuales e inmersión (R, E)

• Navegación (R, E)

• Visualización (R, E)

• Agentes (R, E)

• **Interfaces multiusuario (groupware) (R, E, RP)**

El término Groupware es una contracción de las palabras group = grupo y software. Pretende englobar a todas las aplicaciones computacionales que sirven a más de un usuario, bajo un ambiente compartido. Este término fue utilizado inicialmente para referirse a " un sistema basado en computador más los procesos sociales de grupos ", sin embargo, fue restringido a sistemas basados en computador, pues " se considera que los sistemas y los grupos, son entidades que interactúan íntimamente.

Posteriormente fue definido formalmente por C.Ellis, S.Gibbs y G. Rein en "*Communications of the ACM*" como: " sistemas basados en computador que soportan a grupos de personas relacionadas en una tarea común y que provee una interfaz a un ambiente compartido".

Estos sistemas difieren de los tradicionales, al permitir que los usuarios interactúen directamente entre ellos " ya sea preparando un documento, consultando una base de datos, etc., utilizando al computador como herramienta de interacción, en vez de ser sistemas en los que el usuario interactúan independientemente con el computador.

La comunicación que pueden obtener los grupos, al utilizar aplicaciones Groupware, se puede clasificar de acuerdo a estos tres escalafones:

- ✓ *Coordinación*: es una actividad en sí misma, que ayuda a la organización de las actividades de los miembros del grupo. (por ej.: se coordina una reunión).
- ✓ *Colaboración* : es la realización de actividades individuales, en forma coordinada, que permitirán de una tarea común. Una vez complementadas las actividades, no es distingible el trabajo individual. (por ej. :se colabora si cada miembro del grupo escribe un capítulo de un libro que será editado en conjunto).
- ✓ *Cooperación*: es la realización de una actividad en forma coordinada y conjunta. (por ej. :si en la edición de un documento, cada miembro puede entregar sus opiniones y observa el resultado del conjunto de cambio que se hacen con los otros miembros).

capedsc@hotmail.com

El desarrollo de esta guía esta protegido bajo la ley del DERECHO DE AUTOR a favor de la Sociedad Civil CAPED S.C.
Copyright ° México 2009

Registro: 24-76022/14
5/01/2009
CAPED S.C.

Ayúdanos a proteger este registro, informándonos si esta guía te ha sido vendida de cualquier instancia ajena a CAPED S.C.

E-MAIL: capedsc@hotmail.com

Matemáticas básicas

- 1) Si tenemos una ciudad cuya población se ha duplicado en 10 años, en cuantos años se triplicará con el mismo índice de crecimiento
(según yo, se resuelve con una ecuación conocida como Ley de Malthus, con ecuaciones diferenciales)
- 2) Si tenemos una matriz A y tenemos además una matriz B que queremos multiplicar por A y obtener la transpuesta de la multiplicación, cual ecuación es equivalente?
$$(AB)^T = BTAT$$
- 3) Si el determinante de la matriz A es igual a cero, que podemos deducir de ese hecho?
Que la matriz A no es invertible

Matemáticas discretas

- 4) Si se tienen 10 pelotas rojas, 5 azules y 2 amarillas. ¿Cuál es la posibilidad de que al extraer dos pelotas (con reemplazo) ambas sean rojas?
- 5) Se tienen 7 libros distintos, 3 de matemáticas y 4 de informática, que se quieren colocar en una estantería. ¿De cuantas formas distintas se pueden colocar si los libros de cada materia deben ir juntos?

Números complejos

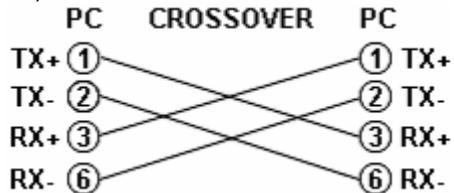
- 6) Cual es el resultado de $(3 + 2i) - (4 + 1i)$
$$(3 - 4) + (2 - 1)i = (12 - 2) + (3 + 8)i = (10 + 11)i$$

Teorema: $(a+bi) \cdot (c+di) = (ac-bd) + (ad+bc)i$.

Redes y telecomunicaciones

- 7) Primitivas mínimas de servicio
Request, Indication, Response, Confirm
- 8) Cual es el mejor protocolo de ruteo?
ARP, OSPF, IGRP, GGP, EGP, BGP, RIP, ICMP, IPX
(Nota: estos son algunos protocolos de ruteo, pero no se cual es el mejor)
- 9) Cual de los siguientes es un método de acceso al medio?
Respuesta = CSMA/CD ("Carrier Sense Multiple Access Collision Detect"),
- 10) Cómo funciona el protocolo X.25?
Por conmutación de paquetes
- 11) Tipo de transmisión que puede transmitir en ambas direcciones, pero no en forma simultánea half-duplex

12) Cual es la normativa de cableado para conectar dos computadoras sin utilizar un hub?



13) Pregunta sobre un filtro pasa baja (no recuerdo el contexto de la pregunta)

Un filtro pasa baja es un arreglo de componentes electrónicos que solo deja pasar las frecuencias menores a la frecuencias de corte.

$$f_c = \frac{1}{2\pi RC}$$

Donde: f_c es la frecuencia de corte en Hz
R es la resistencia en ohms, y
C es la capacitancia en faradios.

14) Determine la relación señal ruido de un canal telefónico de 30dB y un acho de banda de 3000 Hz

Teorema: Capacidad de Shannon $C = B \log_2(1+S/N)$ bps

$$C = 3,000 \log_2(1+1000) = 30,000 \text{ bps}$$

NOTA: (1001) es igual al logaritmo natural de $\ln(1001)/\ln(2)$ y es igual a 9.97

Ing SW

15) Que quiere decir CASE?

Ingeniería Asistida por Computadora

16) ¿Que es ingeniería inversa?

La ingeniería inversa es un método de resolución. Aplicar ingeniería inversa a algo supone profundizar en el estudio de su funcionamiento, hasta el punto de que podemos llegar a entender, modificar, y mejorar dicho modo de funcionamiento. Este término no sólo se aplica al software de protección. Se considera ingeniería inversa también al estudio de todo tipo de elementos, por ejemplo equipos electrónicos, microcontroladores, etc..., siempre y cuando el resultado de dicho estudio repercuta en el entendimiento de su funcionamiento.

Lógica de predicados

17) Cual es la traducción correcta para el siguiente enunciado?

$$(\forall \text{ ave}) \Rightarrow (\sim \text{ vuela})$$

Todas las aves vuelan

Algunas aves no vuelan

Ningún ave vuela

Algunas aves vuelan

(NOTA: no recuerdo muy bien si el planteamiento, pero tiene que ver con los cuantificadores universal (\forall) y existencial (\exists))

Sistemas Digitales

18) Cual circuito integrado tiene de 10 a 100 componentes por chip?

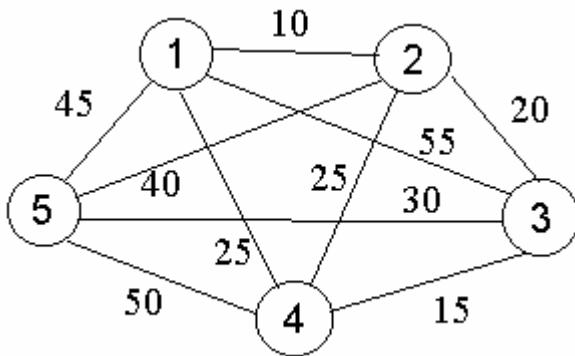
MSI

Teoria de grafos

19) Cual de los siguientes diagramas contiene un circuito de Euler?

(presenta varias imágenes para determinar cual(es) figuras cumplen la condición de ser circuito de Euler

20) ¿Cual es el peso del recorrido de distancia mínima del grafo que se presenta?



Lenguajes de programación

21) Como se denominan los elementos ejecutables de que se compone prolog?

Predicados

Graficación e Interacción humano-computadora

22) ¿Cuántos grados de libertad necesita un brazo de robot para mover las piezas de un tablero de damas chinas?

(NOTA: Un robot tiene grados de libertad o DOF (del inglés Degree Of Freedom). Si el robot sólo se pudiera mover sobre una línea tendría 1 DOF, 2 si se pudiera mover en un plano, etc. Es evidente que cuanto mayor sea el número de grados de libertad mayor serán las posibilidades de movimiento que se puedan obtener.)

23) Cual es la disciplina que se ocupa de los sistemas de control y comunicación en las personas y en las máquinas, estudiando y aprovechando aspectos y mecanismos comunes
Cibernética

24) ¿Con cuantos bits puede representarse una imagen con 4 tonos de gris?

2^n = 4 tonos de gris, n= numero de bits

Ejemplos: 2^2 = 4 tonos de gris, 2^4 = 16 tonos de gris

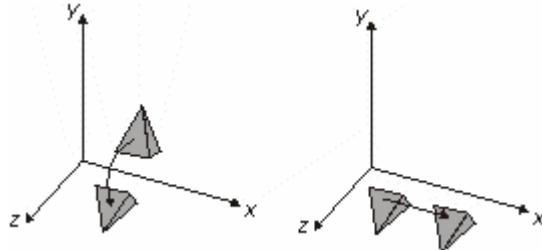
25) ¿Cuántos MB mide una imagen de 3"x5" en RGB a una resolución de 300 p.p. ?

$$\text{Tendríamos la resolución} = (3 \times 300) \times (5 \times 300) = 1,350,000 \text{ pixeles}$$

Para RGB (24 bits por píxel, cada píxel se anota con 3 bytes), entonces

$$1,350,000 \text{ pixeles} \times 3 = 4,050,000 \text{ bytes} = 3.86 \text{ MB}$$

26) Cuál de las siguientes es una matriz para el movimiento de un cuerpo que rota sobre el eje x y se traslada después sobre el mismo eje x



Rotación seguida de traslación:

$$\mathbf{T}((\mathbf{x}, \alpha), \mathbf{p}) = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & \cos \alpha & -\sin \alpha & p_y \\ 0 & \sin \alpha & \cos \alpha & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

NOTA: Es posible combinar rotaciones y translaciones utilizando las matrices correspondientes, pero estas transformaciones no son conmutativas

Rotar y después trasladar \Leftrightarrow Trasladar y después rotar

EL ÁREA DE INFORMÁTICA

Tipo de canales de proceso de reclutamiento : **B. Interno y externo**

El dueño de una empresa tiene 3 pb: contabilidad retrasada, inventario no actualizado y nóminas. A cual de estos pb puede darle solución más pronto: **B. Inventarios**

Para definir el problema en la planeación de un proyecto de programación es necesario

I Desarrollar un enunciado definitivo del problema a resolver

II Técnicas de estimación que se utilizan A. I, II

La Auditoria_ tiene la finalidad de realizar una misión de una opinión profesional sobre operatividad eficiente y observación, normas establecidas, ideas para operaciones y sist aplicados **C . Informática**

En la solicitud de USR para el desarrollo de un sist es recomendable que se incluya inf que facilite su atención ¿ Cual de la inf siguiente NO debe incluirse? **C. Programa de actividades**

Son datos que se deben de proporcionar al proveedor en la propuesta para la adquisición de equipo de cómputo o de SW respecto del soporte al usr, con EXCEPCION de: **D. Experiencia previa por parte del personal de soporte**

Son ventajas de un sist para el trabajo en grupo, con EXCEPCION de: **D. Dividir la responsabilidad en la toma de decisiones**

Ventajas de la grafica de Gantt en la plantación de proyectos Informáticos: **B. Muestra el tiempo real y el estimado**

Diferencias entre auditoria lineal y la de staff es que la lineal es: **B. Espontánea y natural, derivada de la personalidad del individuo**

Que regla tradicional de la admón de la organización ya no se aplica actualmente en la reingeniería. **B. Las organizaciones se deben adaptar a las necesidades del cliente**

La seguridad de los sist en las organizaciones se fundamenta en el estricto control de los accesos a las redes, sean locales, externos, o internos. Existen dos áreas de la organización que se vinculan estrechamente con el área informática quien tiene derecho a usar una clave de acceso en la organización y esta son: **C. Recursos humanos y administración**

LAS ORGANIZACIONES

Formula para obtener los costos de venta es: **A. Inventario Inicial + Compras netas – Inventario Final**

Que características tiene el autofinanciamiento: **D. Paga intereses pero no moratorios**

En la etapa de control, el principio de_ señalar “El control debe implicarse antes que se efectué el error”
B. Oportunidad

Cuales de las opciones siguientes no corresponden a presupuestos: **B. Verificar fallos o éxitos**

Cuales de los siguientes no es un proceso contable que delimita e identifica: **A. Entidad**

El balance general como estado financiero de una empresa permite. **D. Saber cual es la situación financiera actual**

Cual es el orden lógico del proceso administrativo **A, Planeación Organización Dirección Control**

Los / Las __ son documentos detallados que contienen inf acerca de la organización de la empresa en forma ordenada y sistematica: **A. Manuales**

Son técnicas de presupuestación de capital que no consideran el valor del dinero en el tiempo

II Valor presente neto

III Tasa Promedio de rendimiento

B. II III

El análisis de puestos esta conformado por: **A. El perfil de puestos y el perfil de la persona**

Quien propone como fases del proceso administrativo la planeación, organización, integración, de personal Dirección y Control **C. Harold koontz**

Los_ son el resultado del proceso de planeación y se pueden definir como diseños o esquemas detallados de lo que habrá de hacerse en el futuro y las especificaciones necesarias para realizarlas:

Planes

La teoría de la admón de Taylor esta enfocada principalmente a: **B. Las personas**

La diferencia entre pb y un conflicto es que el problema involucra: **A. Sentimientos**

Considera la declaración de misión sig: "Pensamos que nuestra mayor responsabilidad es que ante los médicos, enfermeras, pacientes, madres y todos aquellos que utilizan nuestros productos y servicios" ¿Cuál es el elemento que especifica el interés esencial de la compañía? **A. Clientes**

La aplicación del Método de Valuación de inventarios Primeras Entradas Primeras Salidas causa que los inventarios queden valuados el costo: **C. Más actual**

Cual es la semejanza entre un manual de organización y un manual de procedimientos **D. Son utilizados como técnicas de admón.**

En cual de los casos siguientes un proyecto de inversión NO se debe aceptar

A, El valor presente neto del proyecto es igual a cero 0

Considere a una persona que tiene \$ 100 en una cuenta de ahorros y una tasa de interés de 8% anual, utilizando el interés compuesto y el factor de capitalización ¿Cuánto valdrán los \$100 al final del 3er año? **A.125.97**

La dirección general de una empresa solicita que se invierta en automatización, existen 4 problemas. Para tomar la decisión correcta hay que considerar dos aspectos: el tiempo en el que se tendrá implantado el sistema y la capacitación del personal responsable de la operación. El periodo contable se terminara en seis meses.

D. Costo 250, 000 dólares consta de procesos en línea a través de red interna, Intranet, internet y extranet, capacitación en línea a los operadores desde el inicio del proyecto y con OUTSOURCING mensual del proveedor durante los primeros seis meses

Para el presupuesto de SW y HW del siguiente año hay que considerar el cambio de cómputo central el cual será tomado por el proveedor al 25 % del valor del nuevo equipo. Adicionalmente había que considerar un 12 % del costo del equipo a precio de lista para gastos de instalación. El costo de actualizar el SW será de 5 % del valor del equipo según precio de lista. El área de finanzas solicita el presupuesto de inversión total; si se sabe que el equipo nuevo tiene precio de lista de 85 750 dólares que la inflación esperada sería el 15% y además se tendrá que adecuar la acometida eléctrica con una instalación nueva con un costo de 7 500 dólares ¿Cuanto será el valor del presupuesto de inversión en pesos a un tipo de cambio de \$9.50 por dólar: **D. \$943,810.46**

NORMATIVIDAD JURÍDICA

De acuerdo a la LFT no es una consecuencia de los riesgos de trabajo

B. Pensión laboral

El derecho moral de un autor con relación a su obra es

I Inalterable, III Irrenunciable, IV Imprescriptible, VI Perpetuo

D. I, III, IV, VI

De acuerdo con la LFT a falta de estipulaciones expresas, la relación de trabajo será por

D. Tiempo Indeterminado

El Art. 18 inciso F de la ley federal de derechos de Autor establece que.

D. El programador de un programa de computo puede tener una copia para uso exclusivo como archivo de respaldo

Que delitos dan como resultado actividades ilícitas en el área de informática y No están estipulados en la legislación:

A. Atípicos

Para evitar un ilícito se requieren un control preventivo y correctivo

¿Cuál no corresponde a preventivo? **B. Aplicación de las disposiciones jurídicas penales**

Conforme la ley de propiedad industrial no se consideran invenciones

B. Los modelos de utilidad que, como resultado de una modificación en su disposición, configuración, estructura o forma presenten una función diferente o ventajas en cuanto a su utilidad

De acuerdo con la LFT No es un día de descanso obligatorio **A. 5 de Mayo**

El Art 2^a de la convención de Berna define a una obra como: **B. Cualquier producción del entendimiento de ciencias, letras o artes**

Cuando se usa una marca que pueda ser confundida con otra marca registrada, para amparar los mismos o similares productos o servicios, constituye un delito si. **D. Se encuentra esta conducta prevista como delito en el código penal**

Al Art 6 constitucional el derecho de la organización esta garantizado a todo individuo con la limitación de que

I NO provoque algún delito,,II NO ataque derechos de 3ros,,IV NO perturbe orden publico,,V No ataque la moral A. I, II, IV V

La promoción del desarrollo intermático racional requiere instalaciones de coordinación que garanticen la participación de las diversas instituciones públicas con atribuciones vinculadas a la infraestructura. ¿Qué dependencia de admón publica federal registra protección de datos públicos y de particulares? **C. SEC COM y FOM IND**

La ley federal del trabajo es un reglamento del artículo: **C. 123 apartado A Constitucional**

El marco legal en que deben desenvolverse todas las personas de nuestro país es: **C. Constitución política**

Son derechos de Autor de un programa de computo con Excepción de. **B. Estar registrado como autor ante la dirección general de derechos de autor**

De acuerdo con la LFT cual de los enunciados sig es verdadero: **C. los mayores de 16 años pueden prestar libremente sus servicios, con las limitaciones que marca la ley**

TRATAMIENTO DE LA INF.

Que técnica no corresponde a la etapa de recolección de inf. durante el análisis detallado:

D. Revisión de registros

Son datos que se deben proponer el proveedor en respuesta para la adquisición de equipo de cómputo o sw respecto del soporte del usr, adquiere con excepción de: **B. Lista de Oficinistas de Ventas**

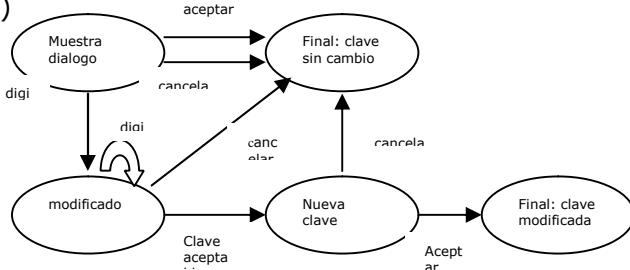
Dos caract importantes de un método de acceso a una tabla generada por dispersión (hashing) son acceso ____ y tiempo de acceso ____ : **B, Por llave parcial, lineal**

Suponga que se tiene información organizada a través de índices dispersos y que algunas llaves son buscadas con mayor frecuencia el propósito es optimizar las búsquedas en la información a través de un mecanismo que proporcione: **B. La ubicación de la i-esima**

Llave de acuerdo a una distribución uniforme de las llaves en el área de almacenamiento principal. Un sist de inf operativo esta orientado a la solución de problemas operativos: **A. Estructurados**

En la interfaz de un sistema aparece un cuadro de dialogo que muestra el valor de la variable clave y 2 botones ACEPTAR Y CANCELAR consisten en 6 dígitos y puede aceptarse o modificarse por otros dígitos si se modifica para que el cambio tenga efecto se debe oprimir ACEPTAR si se oprime CANCELAR en cualquier momento la operación termina y el valor y el valor queda sin cambio elija el diagrama de transición de estados que representa la operación del cuadro de dialogo, suponga teclado numérico.

D)



Son caract de los sist para la toma de decisiones:

I flexibilidad, adaptabilidad y respuesta rápida

IV permite que los usuarios incrementen y controlen el insumo del producto.

B. I, IV

En la teoría general de sistemas el atributo de retroalimentación de información es indispensable para. **C. Iniciar un proceso de computación para la toma de decisiones**

Que aspectos se deben consideran en el estudio de viabilidad técnica. **C. Evaluación de las características físicas del proyecto y determinar si es posible o existen herramientas para tener éxito en el desarrollo**

El estudio de Factibilidad técnica de un sistema de información determina si: **A, Si se puede o no desarrollar con el personal, equipo y software**

La / el ____ registro de información en una computadora solo accesible para lectura solo ente autorizado. La / el ____ requiere que los recursos de la computadora sean modificados la ____ requieren que sean disponibles a los ente autorizados: **D. Secreto, integridad, disponibilidad**

Se dice que la búsqueda es interna cuando los datos a buscar están: **B. Dentro de la memoria RAM**

MATEMÁTICAS

$$\frac{dy^2}{dt} + 4 \frac{dy}{dt} + y = \frac{dx}{dt} x$$

Es un sistema Inestable. **D.**

Considera una función 5 que permite calcular un determinante de 3×3 para resolver un sistema de ecuaciones lineales de 4×4 con una solución única. ¿Cuántas veces se ejecuta la función? **C. 20**

$$A = \begin{bmatrix} 1 & 1 \\ a & 1 \end{bmatrix}$$

¿Cuál es el cuadrado de A? **B.**

Existen 10 rollos de película en una caja y 3 son defectuosos. ¿Cuál es la probabilidad de seleccionar un rollo defectuoso seguido por otro también defectuoso? **A. 9/100**

Si en una ciudad 24 % de la población tiene sangre de tipo B y si tomamos una muestra de sangre a 20 personas de esa población. ¿Cuál es la probabilidad de que exactamente 3 tengan sangre de tipo B? **A. 0.01516**

70% de los hombres y 20% de las mujeres poseen cierta característica. Supongamos que a igual # de hombres y mujeres. Si seleccionamos al azar una persona con esta característica. ¿Qué probabilidad hay que sea hombre? **C. 77.7%**

$$S = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Obtener una base 2×2 **B.**

El arco de la parábola $y = 5x^2$ se hace girar en torno al eje y de (1,5) a (2,20). ¿Cuál es el área? **B. 146.92**

$$\int_1^2 \int_{0}^{2x} x(xy) dy dx$$

C. 9

Sea una matriz de orden $n \times n$ si el determinante A es diferente de cero, entonces se puede afirmar sobre la solución de la ecuación $Ax = b$, donde x es un vector de variable y b es un vector constante. **D. Para algunos b, el conjunto de solución es más de un punto**

Una función es: **D. una representación matemática que asigna a cada valor de entrada (x) uno y solo uno de salida (y)**

Sea una matriz $n \times n$ donde el iésimo renglón son ceros: **D. La inversa de A existe si A contiene valores 1 en la diagonal excepto [i,i]**

Sea $AT = A$ y B de orden $n \times n$ $(AB)^T = B^T A^T$ **D. BT AT**

Dos rectas en el plano cruzan si y solo si poseen uno y solo un punto en común

$$R1 : a_1 x + b_1 y + c_1 = 0$$

$$R2 : a_2 x + b_2 y + c_2 = 0$$

$$\text{D. } a_1 b_2 - a_2 b_1 \neq 0$$

$$\int \sin^2(3x) \cos(3x) dx$$

A. Sen 3 (3x) / 9

La raíz de multiplicidad 2 de la ecuación $(x+2)^2(x^2-4)(x-2)=0$ es: **A. 2**

Dada A encontrar una matriz S tal que $S^{-1}SA = D$ donde D es una matriz diagonal

B. $S = \begin{bmatrix} 1 & 1 \\ 2-i & 2+i \end{bmatrix}$

La función que en un experimento aleatorio asigna a cada elemento en el espacio muestral un numero real es **B. Variable aleatoria**

Para la sol aproximada de Ec con una variable mediante el algoritmo de bisección es necesario

D. Conocer los valores de la variable independiente que ocasionen un cambio de signo a la variable dependiente

Para la solución aproximada de EC con una variable. **D. Conocer dos valores de la variable independiente que ocasionan un cambio de signo de la variable dependiente**

Que no puede concluir de la afirmación $P[A \cap B] = 0$ donde A y B son cualquier par de eventos no vacíos.
D. Ninguna de las anteriores, porque existen eventos no ajenos cuya intersección tiene probabilidad igual a cero

Para calcular el área de una región situada entre las graficas de las curvas de dos funciones F y G y las líneas $X = A$ y $X = B$ de tal forma que $F(x) > g(x)$ para toda X que pertenece al intervalo $[a,b]$ esta dada por: **B. $\int_a^b (f-g)(x)dx$**

Lim $\frac{6x+1}{x+2}$ $x \rightarrow \infty$ **D. 2**

Si F es continua en la región: $R = \{(x,y) | a < x < b, c < y < d\}$ entonces esta dada por las integrales definidas en la región que cumplen **D. $\int_a^b \int_c^d f(x,y)dy dx = \int_a^b \int_c^d f(x,y)dx dy$**

Cual de las siguientes ecuaciones es una ecuación diferencial, lineal de primer orden. **C. $X^2 Y' - \operatorname{Sen}XY = \operatorname{Tan}X$**

Cual de las siguiente relaciones no es funcion real de variable real? **D. $\{(x,y) / y = \sqrt{1-x^2} = 0; -1 < x < 1\}$**

Derivar $y = \cos^2(x)$ **C. $-2\cos x \operatorname{sen}x$**

Corresponde a la energía ____ y U corresponde a la energía _____ por lo tanto W corresponde a la energía ____ **C. cinética, de potencia gravitacional, total**

La distancia focal de una lente delgada se define como:

I La distancia entre la lente y el punto definido X el eje óptico y el punto de objeto

La reflexión total interna solo ocurre cuando **B. La intensidad de un haz de luz es mayor a la intensidad de la luz del otro lado de la superficie en que incide**

Dado el vector $x = (1, -4, 3)^T$ que norma vectorial produce un mínimo? **A. $|x|$**

La grafica $4x^2 - 20x - 20y + 97 = 0$ **C. Parábola**

En una caja se tienen 3 bolas rojas, 2 azules, 5 amarillas. Se sacan 2 al azar con sustitución ¿cuál es la probabilidad de obtener exactamente una roja? **A. 21/100**

Considere $x = \{1, 2, 3, 4\}$ y la relacion $R = \{(1,1), (2,2), (3,3), (4,4), (2,1)\}$ en $X \times X$ R NO es una relación de equivalencia porque NO es: **C. Simétrica**

Considera la función $f(x) = 2 \operatorname{sen}x (x-2) (1/2x)$ una función modificada para la aplicación del método del punto fijo en la obtención de raíces **A. X_{n+1} = 4 sen (X_n -2)**

Desviación estándar de: 2 3 6 8 11 **A. 3.29**

$$W = y \cos v, u = x^2 + y^2, v = xy, \frac{dw}{dx} = \mathbf{D. 2x \cos xy - y(x^2 + y^2) \operatorname{sen}x}$$

Átomos de los materiales aislantes poseen _____ electrones de valencia y los átomos de conductores _____ electrones de valencia. **C. 5 a 8, 1 a 3**

Suponga que la ecuación Einstein $E = mc^2$, la masa esta en Kilogramos y la velocidad de la luz en metros (m)s / seg cual es el valor de e en unidades del sistema interno. **B. Kg m²/s²**

La energía que adquieren los cuerpos al estar en movimiento es: **A. Cinética**

La segunda ley de Newton se aplica tanto a una pelota de Beis Ball como a un ec debido a que se satisface la ley **C. de la gravitación universal**

$$\int \pi/3\pi/4 d\theta \csc v \quad \mathbf{A. (V^2 - 1)/2}$$

En las formulas siguientes $w = U + (1/2) m v^2$ (o tal vez la u se v) $U = -e^2 / R$ $(1/2) m v^2$

El fenómeno óptico que ocurre cuando una onda incide en una superficie que separa dos medios ópticos como el aire y el vidrio provocando que una parte de la onda sea transmitida al segundo medio se conoce: **B. Refracción**

Sean A y B dos sucesos entonces la $P(B/A)$. La probabilidad condicional.... Que ha ocurrido es: **A. P(B/A) = P(A ∩ B) / P(A)**

Considera la función una función modificada para la aplicación del método. Punto Fijo en la obtención raíces es:

$$F(x) = 2 \operatorname{Sen}(x-2) - (1/x) x$$

$$F(x) = 2 \operatorname{Sen}(x-2) - x$$

$$F(x) = \operatorname{Sen}(x-2) - x$$

$$\text{Si } F(x) = r, s / s = e^{2r}$$

$$F(s) = (r, s / 2 \operatorname{Sen}(x-2) - (1/2) Xr)$$

$$2 \operatorname{Sen}(x-2) - Xr / 2 = 0$$

$$2 \operatorname{Sen}(x-2) = -xr / 2$$

$$\mathbf{A. X_{n+1} = 4 \operatorname{Sen}(X_n - 2) - X_n}$$

Cual es la formula general para poder simular valores de las variables aleatorias (con distribución exponencial).

$$\mathbf{C. x = -1/2 \ln r, 0 <= r <= 1}$$

En una facultad, 25% de los estudiantes reprobaron inglés, 15% reprobaron química, 10% reprobaron ambas materias, si seleccionamos a un estudiante al azar cual es la probabilidad de que haya reprobado inglés y química.

A. 30%

Una empresa desea conocer la lealtad de sus empleados, Si les ofrecieran otro trabajo mejor. Para lo cual cuenta con la siguiente información:

	Número de empleados			
	0 - 1	2 - 5	6 - 10	10 o mas
Se queda	10	30	5	75
No se queda	25	15	10	30

¿Cuál es la probabilidad de seleccionar un ejecutivo que se quedaría y tenga 10 años de servicio?

D. 0.714

Cuales son los vértices de la región factible del problema de programación lineal.

Max Z= $3X_1 + 2X_2$

$$2X_1 + X_2 < 6$$

$$X_1 + 2X_2 < 6$$

$$X_1, X_2 > 0$$

- A. **A(0 , 1/2) C(3 , 0) E(0 , 3) B(1/3 , 0) D(2 , 2)**

Si una secretaria comete un promedio de 3 errores al mecanografiar cada hoja ¿Cuál es la probabilidad de que al teclear dos hojas cometa por lo menos 2 errores? Suponga la dist. De Poisson para el numero de errores.

B. 0.224

La ec lógica siguiente es equivalente a:

$$a \cdot b \cdot c + a \cdot b \cdot (\neg c) + a \cdot (\neg b) \cdot (\neg c) \text{ donde } \cdot = \text{y logico}, + = \text{o logico} \text{ y } \neg = \text{negacion}$$

A. a

Un dado esta cargado de tal forma que la probabilidad de salir una cara dada es proporcional al numero de puntos que tiene la cara. ¿Cuál es la probabilidad de obtener un numero menor que a 4? **B. 0.2857**

5 hombres y 4 mujeres en una fila de modo que las mujeres ocupen sitios pares ¿De cuantos formas pueden sentarse? **D. 5!4!**

Cual de las reglas siguientes se conoce como la ley de Morgan

$$\text{C. } \neg(A \cup B) = \neg A \cap \neg B$$

4 libros de matemáticas, 6 de física, y 2 de geometría todos ellos diferentes se colocan en el librero ¿De cuantas maneras posible ordenarlos si los libros de cada asignatura deben estar unidos?

C. 207, 360

Cual es la moda? 9 8 9 7 6 5 8 8 7 6 8 1 0 **C. 8**

Cual es la mediana de los siguientes datos? **B. 8**

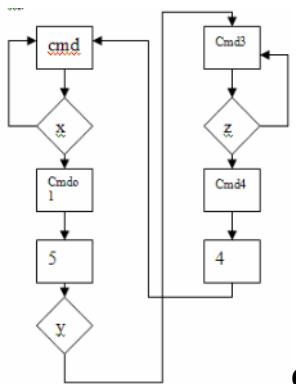
Forma normal disyuntiva **C. $(\neg P \cap Q \cap \neg R) \cup (P \cap \neg Q \cap R)$**

La mediana es una medida de: **A.Tendencia central**

ESTRUC DE DATOS Y ALGORITMICA

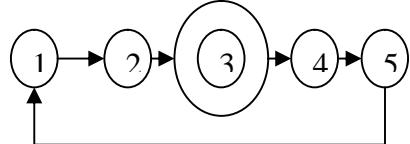
Una / Un ___ acepta un lenguaje tomando como entrada un símbolo Σ produciendo un cambio y solo un estado siguiente. **D. Autómata finito determinístico**

Para la maquina de estado siguiente (representada con ASM) Cuantas entradas hay:



C. 9

El siguiente autómata reconoce



C. pares de unos o 2 + 15 unos

Que algoritmo no permite visitar todos los nodos de un grafo no dirigido conexo. **A. Dijkstra**

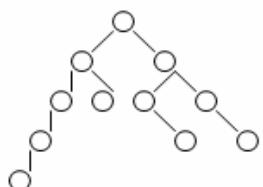
En la teoría los árboles, la longitud de la ruta del nodo raíz T a un nodo ni (donde $i=1,2,\dots,j$) se conocen como: **A. Altura**

Al ordenar paso a paso el arreglo de números $A=\{22, 7, 14, 84, 2, 43, 29, 3, 65, 24, 54, 32, 11, 6\}$ que método de ordenamiento se utilizó si el proceso es de la siguiente manera:

6,7,14,84,2,43,29,3,65,24,54,32,11,22
2,7,14,84,6,43,29,3,11,24,54,32,65,22
2,7,14,84,6,22,29,3,11,24,54,32,65,43
2,7,14,84,6,22,29,3,11,24,54,32,65,43
2,7,14,3,6,22,29,32,11,24,54,84,65,43
2,3,7,14,6,22,29,32,11,24,54,84,65,43
2,3,6,7,14,22,29,32,11,24,54,84,65,43
2,3,6,7,11,14,22,29,32,24,54,84,65,43
2,3,6,7,11,14,22,24,29,32,54,65,84,43
2,3,6,7,11,14,22,24,29,32,43,54,65,84

B. shell sort

Al terminar el nodo 7º del árbol binario de búsqueda siguiente, que forma puede tomar?



Sol. D

Cual es la complejidad de un algoritmo que partitiona un pb en pequeños pb que son resueltos independientemente y al combinar las soluciones de los subpb resuelve el problema original **A. N2**

Elementos relacionados con la complejidad de algoritmos **B. decisiones e iteraciones**

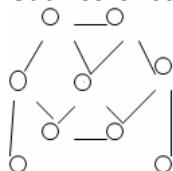
En el modo jerárquico un nodo padre cuantos apuntadores hijos debe de tener. **D. todos los que necesite**

Un pb Np completo tiene la propiedad de que cualquier problema NP puede ser _ a el:
Reducido en tiempo polinomial

Ejemplo de lista invertida **C. Índices secundarios**

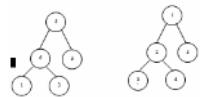
Uno de los requerimientos para que un archivo se pueda organizar por acceso directo es que los contenidos de los registros. **A, Sean de igual longitud**

Cual es el costo del árbol abarcador mínimo del grafo sig.



D. 28

Cual de los siguientes árboles no es un heap máximo



sol. D

Considera un archivo que utiliza un árbol B+ como método de acceso ¿cuál es el recorrido optimo para obtener una lista secuencial de todas las llaves que componen el archivo? **C. Secuencial**

La complejidad temporal del NET de Bus binario es: **A. log 2n**

Recorriéndole nodo el izquierda a derecha ¿Cuántos nodos se visitan hasta llegar a k, si se hace una búsqueda en profundidad (depth-firts)



D. 12

Árbol de un orden izquierda, derecha Cual es el recorrido de búsqueda en anchura en el árbol siguiente.
Sol D

La pila es una estructura de datos también llamada **C. LIFO**

La estrategia para intercalación polifásica para k vías requiere el uso de __ cintas
C. k+1

Cual de los grafos siguientes tiene un circuito de Euler



Características de los árboles b+ excepto
B. Para K datos hay k+1 apuntadores

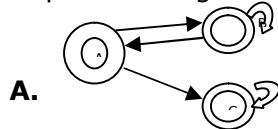
En un modelo padre e hijo (VPH) una ocurrencia consiste en
A. 1 registro padre y 0 o mas hijos

El almacenamiento con base en la secuencia primeras entradas primeras salidas se conocen como: **A.**

Cola

Que diagrama de transición corresponde a la gramática

A →	aB ^I bc
B →	abIaAIb
C →	aCIa



Las técnicas básicas para codificación cifrada son

II Modificar el esquema excepto los seguros de control de accesos

III Sustitución reemplazando los caracteres de texto con caracteres de un alfabeto en clave

C . II, III

Operaciones básicas para realizar la comunicación de un árbol binario de búsqueda no vacío en un árbol AVL

D. Rotaciones simples y dobles

124 byte 512 byte sector->bloque

B. 1

Cual es el orden ascendente de las complejidades siguientes

I log n

II 2n

III n log n

IV n²

I,III,II y IV

La expresión que mejor define a un apuntador es:

D. Variable que contiene la dirección de otro objeto del programa

Una de las dificultades importantes que se presentan en el acceso de archivos por medio de tablas de dispersión es que:

B. con cierta frecuencia a dos registros diferentes les toca la misma dirección

La forma de representar datos almacenados en un archivo binario es igual a la usada por

B. Memoria de la maquina

El algoritmo recomendado para ordenar una gran cantidad de inf en memoria, considerando que esta se encuentra totalmente desordenado es: **C. quick-sort**

Dentro de un relación el grado y cardinalidad: **C, Grado estático, cardinalidad dinámica**

REDES

Tipos de multiplexajes que usa N subcanales para transmisión simultanea de información

D. FDM

En el modelo de red un tipo de conjunto es: **A. conjunto de registros**

Cual es el protocolo mas efectivo para encontrar una ruta de una computadora a otra a través de una copia de intered **D. RIP.**-

Routing Information Protocol.- es un protocolo de routing de vector distancia, es un protocolo abierto. Métodos de transmisión que usa un bit de sincronización por c/carácter transmitido al receptor.

C. Asincrónica

Multiplexaje que aprovecha mejor el ancho de banda del canal. **C. FDM división de frecuencia**

Cual es la configuración del cable UTP nivel 5 de pares para conectar 2 PC's en una red tipo Punto a Punto **B. (1,8) (2,7) (3,6) (4,5) (5,4) (6,3) (7,2) (8,1)**

¿Cuál es la configuración del cable UTP nivel 5 de 4 pares para conectar dos PC's en una red punto a punto, sin utilizar un concentrador? **(1,3)(2,6)(3,1)(4,4)(5,5)(6,2)(7,7)(8,8)**

Un puente de transporte que opera en una subcapa MAC (MAC Brigd) no analiza:

B. Topología del uso de la red

Un ruteador con dos interfaces de red tiene los tipos de direcciones siguientes

C. Dos físicas y una lógica

En una red OSI el direccionamiento lógico es tarea de la capa **A. Red**

El tamaño de un canal(bus) de direcciones de por lo menos ___ bits para poder acceder a 106 celdas de memoria. **B. 20**

En una red TCP/IP la configuración en la transmisión de información se proporciona por capas de: **C. Transporte**

Función de la capa de presentación en el modelo OSI mapea en la capa de ___ de TCP/IP
A. Red

Para efectos de servicios de nombres cual es la definición de domino

D. Conjunto de computadoras relacionadas por partes o una misma organización que comparten la misma red

Las funciones de descubrimiento de rutas son tareas de la capa **A. Red**

El protocolo responsable de la traducción de las direcciones lógicas con las direcciones físicas en una red TCP / IP es: **A. ARP**

Que clase de red mínima es necesaria para conectar a 300 hosts con TCP/IP **A. B**

Cual es el protocolo mas efectivo para encontrar una ruta de una computadora a otra a través de una capa intered **A. RIP**

Cantidad de Hosts de una red clase **B: B. 65,534**

La dirección de Internet 120.1.2.3 pertenece a una red clase: **A. A**

El nodo 148.248.20.156 corresponde a una clase **A. B**

Cual es la dirección valida para un anfitrión TCP / IP **B. 137.222.1.2**

Protocolo interno de ruteo mas utilizado es: **D. IGRP**

Que mascara le corresponde a una red tipo C si se divide en 6 subredes: **B. 255.255.255.224**

Cual de las afirmaciones es cierta para el protocolo TCP **C. Controla el intercambio de datagramas con confiabilidad garantizada**

Capa del modelo OSI responsable de la traducción de las direcciones lógicas en direcciones físicas **A. RED**

Cual estándar se emplea en la capa física del modelo OSI para comunicaciones en serie **D. RS -232**

X.25 es un protocolo orientado a la comunicación de **D. Paquetes por circuitos virtuales**

No es una tarea de la capa de sesión **C. El control de la secuencia de datos**

Si una red existente se separa en N segmentos y estos se intercomunican con un switch con el uso de switch con N puertos se logra

A. Extender la cobertura de la red N veces

Cual de los protocolos siguientes es un mecanismo distribuido de acceso al medio de comunicación **D. Token passing**

Las primitivas básicas que permiten interactuar a las distintas capas del modelo OSI son **Solicitud, indicación y confirmación**

Cuantos apuntadores dependientes inmediatos puede tener un nodo en el modelo de red **A. uno por nodo**

Cual de las configuraciones de red local parece estrella pero opera como anillo

C. Token Ring.- combinación de estrella y anillo

En caso de que estuviéramos en la misma red local y quisieramos enviar un correo a la cuenta abc@correo.universidad.mx cual es la dirección mas corta de correo que pudiéramos usar.

B. abc@correo

Los protocolos de bajo nivel orientados a bit, el mecanismo usado para controlar el flujo entre el transmisor y el receptor es.

A. Sincronización de bits

Un conmutador switch de una red LAN opera a nivel de la capa

B. Física

Los protocolos de terminal virtual tienen varias fases de operación, cual es el propósito

Determinar las características del dialogo entre los extremos de la conexión

Cual es la diferencia entre el protocolo FTP y TFTP

TFTP no realiza validación del usuario

Cual de las características siguientes corresponde a una central privada (PBX) de una red digital de servicios integrados (ISDN) **B. Puede transmitir voz y datos sobre la misma red**

Son características de la topología estrella a excepción de **A, El desempeño de la red esta garantizado**

En cual capa del modelo OSI se establece el método de acceso al medio **D. Enlace de datos**

Por que no se utiliza la configuración MANCHESTER para transmisión a larga distancia.

A. Exige mas ancho de banda

Cual de las tecnologías de conmutación es la única que usa ancho de banda fijo para la transmisión de datos **B. Mensajes**

Una caract que usa _ es el retardo que se puede presentar durante la transmisión de datos

D. Conmutadores de paquetes

En el modelo OSI ¿cuál es la capa que permite especificar los niveles de tensión de 1 señal para ser trasmisida en 1 red? **B. Física**

Cambiar una red Ethernet por una fast Ethernet, el cuello de botella se encuentra en: **A. Servidor de la red**

En la trama del protocolo orientado a bit después del campo de control pertenece al

C. Encabezado a nivel de enlace

Direccionamiento en una WAN

D. Vector distancia

Red publica en el que el ancho de banda es de 300 hz, 20 db ¿cuál es la máxima velocidad que alcanza?

A. 1135

No es función de Firewall

A. Permitir acceso FTP

Los nodos conectados de acuerdo al modelo OSI deben cumplir las siete capas

A. Si

Características de una red Fast Ethernet

II Topología de estrella

IV Utilizar prioridad

A. II y IV

Los códigos de corrección de errores se utilizan en lugar de la detección para errores por

A. Ancho de banda

NO es tarea de la capa de sesión

C. Control de la seguridad de datos

Los tipos de agresión a la seguridad de sistemas de comunicación o redes se caracterizan viendo a la función de sistema como proveedor de información cuando un ente no autorizado se llamo.

A. Deducción, fabricación

Un mecanismo para enviar al nodo receptor llave compartida encriptada en

C. Kerberos

El firewall es una

A. Filtro de paquetes para protección de trafico de internet

Cual de los tipos de Firewall siguiente opera a nivel de la capa de transporte (OSI):

A. Relevador de Circuitos (circuit relay)

De acuerdo con los principios y guías de diseños cual de las interfaces siguientes muestra un diseño adecuado.

Sol C

Que es un emulador: **A. Se construye para trabajar tal como lo haría el dispositivo original**

CLUSTER es un conjunto de _ en distintos platos : Co **D. Sectores de igual posición**

La interferencia constructiva entre dos o mas ondas tiene lugar cuando

C. La diferencia de las trayectorias es un múltiplo entero de la longitud de onda

Características de los main frames

I Sist de enfriamiento especial

II Procesamiento de entrada / salida como front / end

III Instalación especial IV Procesador de un circuito integrado

D. I, II, III, IV

Proceso	Quantum asignado	No. de veces h entra CPU	Tiempo total
D	5	2	34
B	3	3	23
A	5	4	46
C	4	3	27

De las siguientes cadenas de bit cuales son los datos que corresponden para ser paridad par:

1101100

1010011

0101100

1000110

C. 0111

Para detectar un error durante la transmisión se ha optado por el polinomio $X^4 + x + 1$ cual es la representación en bits: **A. 10011**

Función principal del GATEWAY **A. Fungir como puerta lógica**

SW DE BASE

Cuales son los 4 elementos que administran el sistema operativo **D. Memoria, procesos, archivos, dispositivos de E/S**

La primera generación de S.O. se caracterizo por que su programación fue con: **B. Lenguaje de maquina**

Fue diseñado para automatizar el trabajo del operador y el primer paso en el desarrollo de una nueva maquina:

D. Sistema Operativo

Suponga un sistema multitareas y que en cierto momento la situación de los procesos en estado de lista es como sigue:

Considere además que en ese momento no hay ningún proceso en ejecución y los procesos no realizan ninguna operación de e/s ¿Qué proceso ha sido atendido por el CPU menos tiempo?

B

Considere la matriz de acceso siguiente ¿Qué puede utilizar el archivo de juego?

Admón.	Impresora	Modem	Juego	Hoja de Calculo
Admón.	Lectura escritura	y	Lectura	
Contabilidad	Lectura escritura	y	Escritura	Lectura escritura
Dirección general	Lectura escritura	y	Lectura escritura	y
Sara		Lectura		Lectura
Pedro		Lectura	Lectura escritura	y
Claudia				

B. Solo si pertenece al grupo de dirección general

Un ejemplo de SO estructurado en capas es **A. UNIX**

Cual de los elementos siguientes forman parte del SHELL **A. Editores**

Si A, B, C son variables difusas ¿Cómo ceder la operación siguiente C = A and B? **D. La suma del mínimo entre A y B con uno**

Cual es la semántica de la expresión siguiente: X: ave (x)→vuela (x) **C. Si es ave entonces vuela**

Cual es la relación entre entidad y atributo? **C. Atributo es característica o propiedad de la entidad**

Son características de las gramáticas tipo 2 o libros del contexto de acuerdo chomky

B. Lado izq. Formado por un solo elemento no terminal

La clase A hereda atributo y funcionalidad de la superficie B. Para utilizar una función con el mismo nombre en ambas clases ¿Qué característica del paradigma orientado objetos se utiliza? **A. Polimorfismo**

Consideré el conjunto de reglas de producción siguiente:

R1: if A and B

 Then D

R1: if A and C

 Then E

R1: if B and C

 Then F

R1: if D and G

 Then H

R1: if F and G

 Then I

Consideré que se tienen los siguientes hechos en la memoria de trabajo [1 :A, 2 : B, 3 : C, 4: G] donde el numero representa una etiqueta de t, por lo que A se conoció antes que B, C y G, B antes que C y G y C antes que G.

También consideré que la estrategia de conflictos es preferir hechos más ¿ Cual es el orden de las reglas de producción?

A. R1, R2, R3, R4, R5

De las gramáticas sig regulares libros de contexto sensibles del contexto sin restricciones Que tipo corresponde a chomsky? **D. 1,2,3,4**

ARQUITECTURA DE COMPUTADORAS

Capacidad en MB del disco duro si se tienen 1,002 cilindros, 8 cabezas y 32 sectores.

A. 131

Parte de un CPU que sincroniza las actividades dentro de este **D. Unidad de control**

Un medidor Kilowatt – hora de una casa tiene tres discos numerados del 0 – 9 cual es la cantidad en bits para digitalizar un medidor **A. 10**

Cuantos cabezas de lectura/escritura existen en un paquete de discos de 5 platos
10

A que tipo de procesador se está refiriendo

- I Tres a seis nodos de direccionamiento
- II Juego de instrucciones limitado
- III Instrucciones ejecutadas en un ciclo de reloj

C. RISC

Propósito de la memoria virtual es extender la capacidad de memoria

B. RAM

Una característica no propia de los discos duros

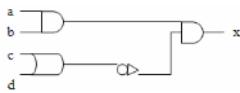
A, Siempre están girando

El teclado de Dvorck es un dispositivo:

D. ergonómico de texto, cuyos símbolos se han dispuesto de manera que su uso sea mas flexible.

SIST DIGITALES Y ELECTRONICA

Considere el diagrama siguiente que valores deben ser asignado a las variables de entrada para que el resultado sea uno

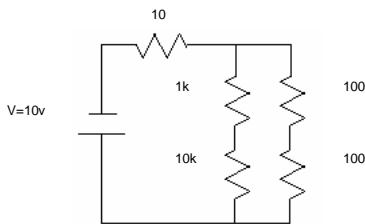


C. A = 1; B = 1, C = 0, D = 0

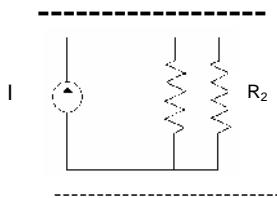
Que función es equivalente a la función sig $g = a'(a+b'c) + a'b + a'b'$

D. $g = a'(a+b'c) + 2a'b$

Obtenga la forma normal disyuntiva a partir de la formula proporcional



B. 48mA



B. $\left(\frac{R_1}{R_2 + R_1} \cdot 1 \right)^2 \cdot R_2$

¿Cuál es la expresión booleana que permite obtener la salida w del circuito convertidor de código? **B. A+B(C+D)**

Si la señal analógica tiene un rango de 0 a 10 V, y se codifica en 8 bits, el error de codificación en el peor caso es lo mas cercano en mV a: **A. 39.06**

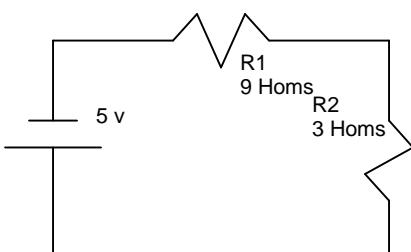
Cual operador logico es necesario para completar la expresión $39 _ 50 = 21$. **C. NAND**

No. Total de entradas que habilita E=0 y S=1 **C.4**

Supóngase que se tiene una unidad de procesamiento central (CPU) formado por los siguientes registros: Un PC (Contador de programa), un MAR (registro de direcciones de memoria), un MBR (registro interno de memoria), un IR (registro de instrucciones), una ALU y una memoria principal principal con una organización de 64K por B. De que tamaño seria los registros siguientes?
PC_____bits.,MAR_____bits;IR_____bits,MBR_____bits **D. 16 16 16 16**

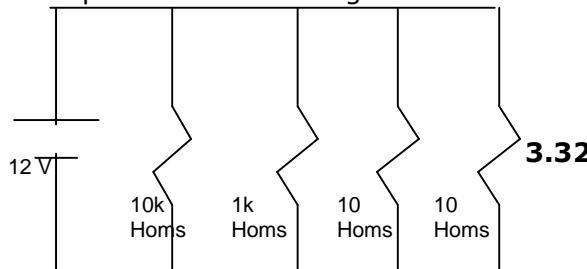
Reducir $f(x,y,z,w) = x'yz + yzw' + xyzw + x'y'zw + w + yz' + x'y'zw' + x'y'zw$ **C. $y' + w' + xz'$**

Calcule la potencia de R2 expresado en Watts



2.27 watts

Calcule la potencia total del siguiente circuito



La atenuación de un canal de datos disminuye con el aumento de:

D. Frecuencia de señal

Que no se ocupa en la señal con modulación en banda base.

C. División de tiempo

La modulación ___ es la resistente en ruidos electromagnéticos.

D. Psk

Circuito de memoria cuyo comportamiento es parecido al de un PLA

B. ROM

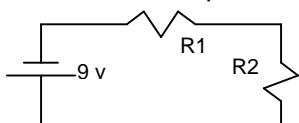
Un CI que contiene en una sola pastilla la alu, la unidad de control, los registros internos, la memoria del programa, la memoria de datos y los puertos de e/s se llama:

D. Microprocesador

El ___ convierte las señales electricas asociadas al dato a otra forma de energia en el caso de una salida y viceversa en el caso de una entrada.

B. Multiplexor

Cual es el complemento a dos del numero hexadecimal (7398)16 con una precisión de 16 bits B. 8C68



$$\text{B. } R_2/(R_1 \cdot R_2) = R_2$$

Cuales son las funciones que pueden realizar el registro de corrimiento universal de 3 bits siguiente: integrado por multiplexores y flip flopsD

C)

F1	F2	FUNCION
0	0	Mantener la información de salida en paralelo
0	1	Carga de información en paralelo
1	0	Entrada salida serie corrimiento a la derecha
1	1	Entrad salida de salida en paralelo

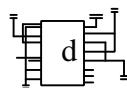
H = 300 Hz Ruido = 20 db ¿Cuál es la máxima velocidad de frecuencia? **C. 19.974**

Una señal digital se puede transmitir por línea telefónica **A, Falso**

Un técnico utiliza un contador como divisor de frecuencias si se alimenta al contador con una frecuencia de 60 Hz que modulo debe ser el contador **D. 144**

Ruido que es producido por variables propias de la transmisión **B. Impulsivo**

Cual es la función lógica en la salida, expresa, en mini términos implementada por el multiplexor siguiente figura.



A. $Y = m_0 + m_1 + m_2 + m_3 + m_4$

Cual es el # de codificadores de 4 a 16 lineas y de 2 a 4 lineas para diseñar un decodificador de 6 a 64 lineas. Considere que los decodificadores tienen entradas de habilitación

C. 4 decodificadores de 4 a 16 líneas y 1 decodificador de 2 a 4 líneas

Si se tiene una memoria EPROM con una capacidad total de 128y un numero de 8 bits por palabra Cuantas lineas de... tiene dicha linea? **C. 16**

Un contador binario completamente sincrono que cuenta en forma natural de $(0000)_2$ hasta $(1111)_2$ dispone de una entrada (LD) para cargar un dato que se presenta en las 4 lineas de datos (D3 D2 D1 D0). Para contar de $(0011)_2$ hasta $(1010)_2$ con este dispositivo debe codificar la cuenta actual y cargar en el contador el número. ¿Cuáles son la cuenta y el número?

C. Cuenta = $(0011)_2$

Numero = $(1010)_2$

Las ecuaciones que representan a cada una de las salidas del circuito codificador en la figura son:

A. $A_1 = D_2 + D_3$

$V = D_0 + D_1 + D_2 + D_3$

$A_0 = D_3 + D_1 D_2$

Dos antenas de micro ondas a 75m. De superficie con un factor K= 4/3 ¿Cual es la distancia máxima a la que se puede trasmisir? **B. 71.4Km**

Los circuitos integrados _ contienen internamente entre 10 y 100 computadoras o una complejidad similar, algunos ejemplos se éstos son los decodificadores, sumadores, comparadores, convertidores de código. **B. MSI**

Modos de direccionamiento de un RISC

I Registro

II Registro indirecto

III Registro inmediato

IV Registro relativo

A. I

Cual es la secuencia de paso para atender una interrupción generada por una tecla de paro de emergencia en una guillotina automática.

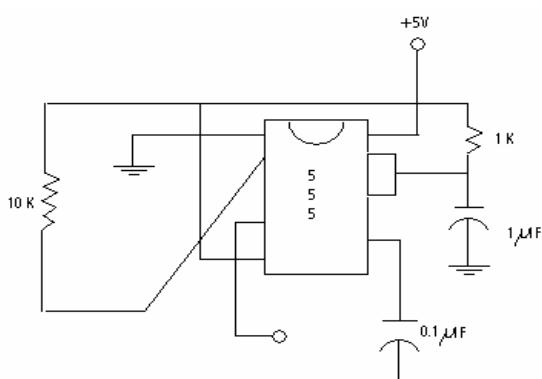
I. Aplicar los registros que se destruyen

II. Ir una rutina de retardo para evitar rebote de teclas

III. Para el motor de bajada

A)I, II, III

El siguiente circuito opera como multivibrador monoestables, el ancho del pulso es de:



D. 1seg

PROG E ING DE SW

Son actividades que se llevan a cabo durante la liberación de un sistema con excepción de:

B. Diagrama de contexto

Heap

15	12	9	7	8	4	
----	----	---	---	---	---	--

D. 12 12 10 7 8 4 9

Propiedades de las transacciones **B. Dinamica, Estructurada, Objetiva e iterativa**

Si en C se define arreglo y estructura

float datos [20]

struct recurso (

 float medida;

 float * acceso)

y luego se define una variable

struct recurso informe;

¿Qué estructura NO es valida? **A. Informe acceso = datos**

Que estructura permite construir una lista enlazada cuyos campos son: nombre, # de seguro social y fecha de nacimiento.

C. Typedef Struct (int día, int mes, int año) date

Struct elem {char nom [25],

Long ns,

Date fnac ;

Struct elem * psig ;

Class punto (int x, y, public punto (int a, int b)

(x=a ; b=b ;))

¿ En que función es necesario incluir en la definición de la clase para sobrecarga del operador +, para que sume 2 objetos de esta clase?

B. Punto operador + (Punto P)

{return punto (x + px, y + p.y)}

La ingeniería inversa consiste en: **D. Un proceso de recuperación de diseño de datos y procedimientos entre otros de un sistema**

Sean estudiantes una relación con esquema (no_mat, semestre, nombre, dirección, estado)

Considere los fragmentos horizontales r1 y r2

r1 = estado = Coahuila (estudiante)

r2 = estado = Yucatan (estudiante)

D. r1 INTERSECCION r2

Cual es el obj de replicar una BD distribuida **A. Mantener independencia entre distintos nodos de la red que componen BD distribuida**

Para que surgieron los Lenguaje de 4º generación) 4GL

Para apoyar el desarrollo de prototipos

Algunos de los elementos para asegurar en un DBMS son:

I Personal autorizado

II Datos

III Esquemas A. I, II, III

Cual es la diferencia de utilizar ODBC y conexión directa a cualquier BD
B, ODBC agiliza las consultas al tener una capa adicional de SW

Para poder concluir que un número telefónico no esta en un archivo secuencial desordenado con 50000 hay que acceder a _____ registros. **C. 50 000**

Cuál es la salida del segmento de programa en C siguiente, suponiendo que las direcciones de memoria de p, q y r son 1001, 1003 y 1005?

```
Int p, *q, **r;
p = 100;
q = &p;
r = &q;
printf ( %6d %6d %6d %6d %6d , p, q, r, *q, *r, **r) ;
C. 100 1003 1005 1003 1005 100
```

Cuál prototipo de función se puede crear con la plantilla en lenguaje C + + siguiente?

```
Template < class T, class T1> T1 muestra_arreglo (T *array, T1contador)
{ T1 1;
for (1 = 0; 1< contador; 1+ +)
cout << arreglo[1] << << end 1 ;
return (T)
}
```

D. char muestra_arreglo (char * , long);

El modelo entidad – relación:

II. Especifica las relaciones que existen entre los objetos de un sistema.

IV. Especifica las relaciones que existen entre las actividades de una organización.

II, IV

Una llave primaria es un: **C. Atributo a conjunto de atributos que relacionan una región con otra en una relación**

Cuando se desarrolla una biblioteca de enlace estático no es conveniente incluir **A. Funciones de uso frecuente**

Una característica del mantenimiento preventivo es que :

A, Utiliza ingeniería inversa y reingeniería

El mantenimiento adaptativo es una actividad que:

B, Modifica el SW para que interactúe adecuadamente con su entorno cambiante

Salida:

Begin

Integer entrada, divisor=2, cociente, residuo, indice=1

String salida

Dividendo=entrada

Do

Cociente=dividendo div divisor

Residuo= dividendo mod divisor

Salida[indice]= chr[residuo]

Indice= indice+1

 Until (dividendo=0 or dividendo=1)

 End

D. Cadena invertida de la entrada entre 2

Prototipo de función que se puede crear con la plantilla en C++

Template < class T, class T1>

{T1=1;

For (i=0; i<contador; i++)

Cout <<arreglo[1] << " end I;

Return (T);}

D.char muestra_arreglo (char *, long)

Considera la declaración en C "float a[12][25]" y la posición inicial de la memoria del arreglo. ¿A partir de que posición de la memoria se almacena la asignación siguiente a[11][23]=31?

A. 229610

El esquema de una relación lo conforman a,b,c,d,e,f, , el {} de dependiendo funcionales que se cumplan son:

ab → c

c → a

d → e

de → f

e → d

e → f

Cual es la cerradura de d? **D. adef**

F o V

SI (x_1, \dots, x_n) son independientes lo mismo para cualquier subconjunto

si (x_1, \dots, x_n) son linealmente independientes

(x_a, \dots, x_n) genera el espacio vectorial = la dimensión de x es n

D. FVV

Aumentar la precisión de un tipo de datos de punto flotante requiere al menos de un dato adicional a ella

Mantisa

Exponente

Exponente o mantisa

Cuales de los subconjuntos en r3 son espacios vectoriales

I Planos de los vectores con 1a comp. b1=0

II Planos de los vectores b con b1=1

III (0,0,0)

A. I,II,III

Lenguaje Estructurado. Las estructuras de control de flujo del programa

A. while, do while, for

Modelo que describe las transformaciones de valores que ocurren dentro de un sistema.

A. Dinámico

Salida:

Int x=5;

Funcion(x);

Printf ("%d",x);

Void Funion(int x)

X++;

Printf ("%d",x);

C. 6, 5

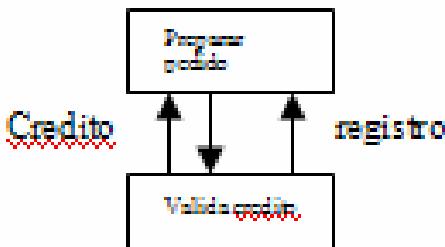
Salida:

```
Float x=0;  
While (x<=1)  
Printf ("% 5,2f", 3*x-1);  
B. 1.00 0.00 1.00 2.00
```

Cuales son los elementos del esquema siguiente

A. 1 Objeto, 2 Datos, 3 Método, 4 Mensaje

Cuál es el tipo de acoplamiento que se presenta en la comunicación entre los módulos mostrados:



A. De contenido

Un pb Asociado con la eliminación de registros en los archivos hash es: **B. Solo se pude borrar las llaves de tipo numérico**

Por ordenamiento según el modelo relacional cual seria el par ordenado para un dominio
B. (columna, tupla)

Dentro de una relación en grados la cardinalidad tiene la caracteristica de: **B. Ambos son dinamicos**

Como se buscan 50000 registros con búsqueda secuencial **50000**

Se esta desarrollando un sist de captura y seguimiento de fugas de agua. Este sist deberá registrar para c/reporte los datos siguientes: fecha y hora del reporte, nombre y domicilio de la persona que reporta la fuga y la ubicación de la fuga.

Una vez registrada la fuga se le asigna # de reporte y se canaliza a una de las cuadrillas disponibles, quienes se encargan de su sol, para seguridad se deben registrar # de Cuadrilla a la que fue asignado el reporte, fecha de asignación y el edo de la fuga ¿cuál es una def de almacén de fugas dentro del diccionario de datos del sist?

C. Num_Rep + Fecha + Hora + Nombre + Domicilio + Ubica_fuga + Num_Cuadrilla

Cual es el tipo de cohesión NO deseada en el modulo siguiente

Iniciar conteo
Abrir archivo
Inicializar reporte

Funcional

Que asociaciones de cardinalidad son incorrectas en el siguiente diagrama: **D. AI – Beca y AI – Profesor**

Son propiedades de las relaciones de las BD

I.-Existencia de alguna tupla repetida

II.-Las tuplas no son ordenadas

III.-Los atributos están ordenados de izq. A derecha

IV.-Ausencias de tuplas repetidas

V.-Valores de los atributos son atómicos

D. III,IV,V

Las características básicas para codificar cifras son:

I.-transposición de letras de texto original se entrelazan de acuerdo a una clase

II.-Modificar el esquema excepto los seguros de control de acceso

III.-Sustitución reemplazando los caracteres de texto con caracteres de un alfabeto clave

IV.-Mostrar, crear o cambiar de control de acceso

B. I,IV

¿Cuales son las funciones y objetos principales de un SGBD? **B. Fun: Buscar, añadir, suprimir y modificar los datos OBJ:**

Mantener integridad , coherencia, concurrencia, seguridad y calidad/cantidad de datos

Un sist de inf para la toma de decisiones tiene la finalidad de ayudar a la toma de decisiones conducente a. **B, Entrenar probabilidades NO concurrentes**

Los sist de inf que se enfocan a la automatización de procesos operativos en una organización se denominan sist **D. De Información transaccionales**

Un sist experto es la aplicación de los sist de inf que **D. Captan y codifican en una computadora el conocimiento de un especialista para simular su forma de pensar**

Que técnica no permite asegurar el control de acceso en la inf de un sist de inf **D. Sentencias SQL para mantenimiento de integridad**

Cual es el propósito de un sist de inf estratégico en una organización: **A. Cambiar de metas**

Durante en ciclo de vida de los sist de inf en que etapa planea la solución **B. Análisis**

Cual de los siguientes NO es un objetivo de las interfaces de usuario

D. Ocultar información que no se quiere que no se sepa a través de códigos del diseñador

Son caract de los sist para la toma de decisiones con EXCEPCIÓN **C. Toman decisiones automáticamente**

Cual de los siguientes es la mejor definición de un sistema **D. Conjunto de entidades interrelacionadas o interdependencia con un objetivo en común.**

Son actividades que se llevan acabo durante la liberación de un sist excepto: **D. Construcción de los archivos de datos**

En la construcción del SW puede incurrirse en un error al momento de la transacción del diseño este error se conoce como. **E. Interpretación de requerimientos**

Que interfaces son ideales para usuarios no expertos debido a que estos no interactúan con la computadora en un lenguaje común **B. Lenguaje natural**

El tipo de usuario adecuado para la prueba final de campo de una interfaz es:

C. Inexperto en el manejo de sistemas de cómputo

Los componentes de un diagrama de flujo de datos (DFD): **A. Proceso, flujo y archivo**

Cual de los tipos de diagramas sig expresa las transformaciones que sufre la inf conforme se mueve desde la entrada hasta la salida, dentro del modelo de análisis estructurado del sist.

D. De flujo de datos

La metodología _ de diseño de SW, se centra en la identificación de las entidades de inf y de las acciones que se les aplican, proporcionan técnicas pragmáticas para transformar los datos en estruc de prog y consiste en una serie de pasos que cubre el análisis y diseño : **A. De Jackson**

Técnica O.O. de Polimorfismos : A. Objetos que se comportan en diferentes formas
CASE **D. Ingeniería de SW asistida por computadora**

DFD, DD, mini especificaciones, diagrama de transición de estados se utilizan en el
D. Análisis

Los análisis candidatos a desarrollarse mediante el análisis de prototipo son aquellos que
I No se tiene claridad acerca de cuales son los requerimientos del sistema

II Los requerimientos deben evaluarse

III Alto costo y alto riesgo del sistema

V Requieren nueva tecnología

A. I,II,III,V

Una librería ha decidido desarrollar un sistema que sea capaz de recibir los pedidos de los clientes en línea a través de Internet. Para determinar los requerimientos del sistema se consultara con diversos tipos de usuarios. Que técnica de recolección se recomienda para obtener información de los clientes de la empresa. **B. Cuestionarios**

Son actividades que se llevan a cabo durante la fase de pruebas de un sistema de información con excepción de:

C, Instalar equipo nuevo y entregar a los usr

¿Cuál de los términos sig no pertenece a la definición de objeto? **A. Rol**

Cuales de las funciones siguientes corresponden a un diccionario de datos

I Escribir el significado de los flujos de datos de los DFD

II Describir la composición de agregado de datos que se mueven en los flujos de DFD

IV Especificar valores y unidades relevantes en los flujos de datos

B. I, II, IV

Cuando el volumen de los datos a procesar alcanza el mayor orden posible la forma de accesarlos y almacenarlos mas eficientemente es a través de: **A, indización multinivel**

En una BD distribuida existen 3 localidades dispersas geográficamente (L1,, L2,L3)

El 80% de los accesos de L1 a L3 son actualización

El 30% de los accesos de L3 a L2 son los accesos de recuperación

¿Cómo deben replicarse las bases de datos en L1,, L2,L3?

B. de L2 en L1

Etapas de un ciclo de desarrollo en espiral **A, Planificación, Análisis de riesgo, diseño, Ingeniería, Evaluación**

El nivel mas alto de un DFD (diagrama de Flujo de Datos) que contiene un solo proceso que representa el sistema completo se conoce como: **A, Diagrama de Contexto**

Que es la homeostasis **A, La tendencia de un sist para equilibrarse**

Que aspectos se deben de considerar en un estudio de viabilidad económica

A, análisis y diagnostico de un proyecto, en términos de inversión y recuperación económica

Cual no es una fase de ciclo de vida : **B. Tiempo Compartido**

Una def de groupware es: **D. Grupo de herramientas de sw que permite la comunicación entre grupos de usuario**

Considera un sist de inf en donde el SW ha sido probado y se encuentra en uso en otras compañías en 4 sitios diferente y una capacitación a 25 personas por sitio (incluye operarios y empleados gerenciales)
¿qué método de conversión conviene implantar? **C. En paralelo**

Si se tiene un entorno cambiante en los sistemas de información como aumentos constantes e indicadores antiinflacionarios, cambio de políticas fiscales, etc. ¿Cuál es la clase de mantenimiento que debe emplearse? **B. Adaptativo**

Opción que representa condiciones del operador JOIN en algunas relaciones
C. Condiciones para establecer vínculos entre dos tablas

El lenguaje que utiliza un SMBD para crear un esquema de una base de datos es
D. Creación de Esquemas

Técnica Clásica de Planificación. **B. Strips**

Cual es la instrucción para representar una operación de restricción y proyección: **B. GRANT**

En C++ (private) puede ser manipulado: **B. misma clase**

Un programa PROLOG **C. Un predicado de primer orden**

El diseño preliminar lleva a cabo: **A. La transformación de los requisitos en los datos y en la arquitectura del software**

En que etapa es mas conveniente medir la productividad de un equipo de desarrolladores de SW? **B. Durante el desarrollo del software**

El diagrama de _ es una herramienta de análisis que representa el flujo de información entre procesos. **B. flujo de datos**

Constituye la base para la elaboración de los planes de prueba que han de aplicarse a un sistema: **B. Especificaciones de procesos**

No corresponde el modelo de cascada: **B. Análisis de Riesgo**

Al atacar el desarrollo de un sistema, en el cual los requerimientos son muy volátiles existen riesgos en el manejo de la tecnología y existe la necesidad de contar operable del sistema en el menor tiempo posible. Cuál de los ciclos de desarrollo será más efectivo?: **D. Prototipo rápido**

El diagrama de flujo de datos (DFD) es un/una: **D. Herramienta de modelado de sistemas que visualiza a un sistema como una red de procesos funcionales**

GRAFICACIÓN, IA, I H-C

Una imagen digital después de aplicar filtro pasa bajas quedan: **B. Borrosa**

Con cuantos bits por píxel se puede representar una imagen un b y n con tonos de grises **B. 2**

Ciencia que trata de la comprensión de la inteligencia y del diseño de máquinas inteligentes, es decir el estudio de las simulaciones de las actividades intelectuales del hombre **D. Inteligencia Artificial**

Sensores que informan sobre el estado interno del robot **C. Propioceptivos**

Mínimo grado de libertad que requiere un robot para jugar damas **B. 3**

Como se conoce el diseño detallado de las escenas en un momento determinado de la acción **B. Cuadro clave**

De acuerdo con estudios realizados sobre el procesamiento de la inf por el humano se estima que en la memoria de corto plazo el común de las personas puede retener ___ elementos. **C.7+- 2**

La teoría constructivista de la percepción afirma que: **D. La percepción implica el proceso activo en que la visión del mundo se forma a partir de la información del medio ambiente y del conocimiento obtenido previamente.**

Las ventajas de interpolar vectores normales (sombreado de Phong) sobre interpolar intensidad o color (sombra de gouraud) son: **C. El SG puede omitir brillos que no ocurren en los vértices, Phong los considera.**

La finalidad de la computación ubicua es: **C. Hacer invisible la metáfora de la interfaz y lograr así que el uso de las computadoras requieran un mínimo esfuerzo**

Uno de los paradigmas de interacción humano-computadora es la manipulación directa, que se caracteriza por el uso de ___: **D. Objetos representados gráficamente seleccionados mediante un dispositivo de entrada para indicar acciones sobre ellos.**

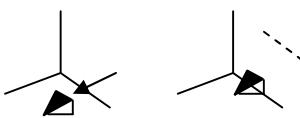
Cuales son los beneficios de la usabilidad? **D. Minimizar tiempo, errores frustración y entretenimiento.**

En un acelerador gráfico se puede tener HW especializado para manejo geometría y de pixeles. El que acelera la geometría generalmente a ___ agiliza el recorte. **C. Aplica transformaciones y proyecciones**

El termino ___ se refiere a cualquier curva compuesta que se forma con secciones polinómicas que satisfacen condiciones específicas de continuidad en la frontera de las piezas. **B. Curva de spline**

Cual es la principal ventaja que presentan los sistemas de vectores aleatorios sobre los de barrido. **B. Su capacidad para trazar líneas con gran precisión.**

Cual es la matriz resultante del par de transformaciones siguientes.



$$\begin{array}{cccc} \mathbf{B.} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ & \mathbf{0} & \mathbf{\text{Cos } \theta} & \mathbf{\text{Sen } \theta} & \mathbf{0} \\ & \mathbf{0} & \mathbf{-\text{Sen } \theta} & \mathbf{\text{Cos } \theta} & \mathbf{0} \\ & \mathbf{0} & \mathbf{0} & \mathbf{-Tz} & \mathbf{1} \end{array}$$

Las ventajas de interpolar vectores normales (sombreado de Phong) sobre interpolar Cuales son los pasos a seguir para realizar un programa de adquisiciones de bienes informáticos de una organización **B. Analizar las necesidades de equipo de computo ver con cuanto dinero se cuenta, elegir el proveedor y adquirir el equipo de computo**

La aplicación de principios de interacción H-PC tiene como meta principal **D. La mejoría en la productividad y seguridad del usuario al emplear sist. Computacionales**

¿Cuál de los enunciados siguientes no es una técnica para ganar la atención del usuario sobre un txt?
Colocar el txt en la parte superior derecha de la pantalla

Cual de las siguientes no es una técnica de presentación de información para atraer la atención del usuario. **B. Claves especiales y temporales**

OTROS

Propiedades de las transacciones. **C.atomicidad, integridad,durabilidad,cohesión.**

Características del aotufinaciamiento.**D.pagar el minimo de intereses.**

Considera parida par, complete con el bit de paridad las secuencias de bits siguientes:1101100---,0101100---,1011011---,1000110---, **c. 0111**

84-cual no esta definido en $x=0,y=2$ en $xy^2-3x^2y^2$. **d.honesta definido.**

68-cual es verdadero. **a.x2=x1/2.**

78- $ax+by+c=0$. **a.recta.**

83-si un numero complejo seleccione cuales son los valores de $z=re i\theta$. **b.**

ANEXOS

Supóngase un sistema multitareas

Proceso tamaño no de veces cpu tiempo total

D	5	2	34
B	3	3	23
A	5	4	46
C	4	3	27

Considera además q en el no hay procesamiento en ejecución cual fue la q atendió al cpu
A,a,b,b.c,c,d,d

Un servo mecánico esta formado

- a. una articulación y su actuador**
- b. un sistema mecánico y un microporcesador**
- c. un sensor de posición con un comparador, un contador y un actuador.**

El crecimiento de registro q agrupa pixel a subregiones dentro del registro mas grande. Un método comienza con un conjunto de puntos grandes a partir de las q van creciendo la 1era diferencia del método.

- a. toma la región a su brillo**
- b. la selección de los generadores iniciales del tamaño de la región.**
- c. la selección del generador inicial y la selección de las propiedades.**
- d. el tamaño a la selección de las propiedades adecuadas de la región.**

Histograma corre a alto contraste. b



respuesta 150

r=30

r=r1+r2+r3.



b)-r2/r1 v

el

a.Cis,Ris,Cis

b.CISC,Cis,Ris

ley de lentz

a.efecto foto electrico de eintein

b.paralelismo atino

cual es el mejor ajuste de 2k

a.primer



b.segundo

c.tercero

d.cuarto

arbol por el metodo de anchura

resp-letras

pila hola

b.aloh

la luz se emite alrededor de una orbita

b.distracción

que dedujo eistein de efecto foto electrico

a.luz algo de ..

b.luz tiene ondulatorias.

c.luz propiedades de la luz y su energis sale de la electricidad.

1.ocupa 2 bytes, reconoce pseudoinstrucciones ORG,DB,DW para definir el origen del programa y reservar uno o dos bytes.

UNO: DB 01H

DOS: EQU 02H

TRES: DW 20H

X: MOV REG.DOS

Y: MOV REG,(UNO)

Z: EQU 0AH

W: MOV REG.Z

a.nombre valor

uno - 0050

dos - 0052

tres - 0054

x - 0056

y - 0058

z - 005A

w - 005C

2.para el siguiente programa encuentra la tabla de símbolos generada por el ensamblador si ORG,EGU,DB, definir el programa, constantes y reserva el byte.

```
ORG 10H
AB EQU 05H
BC EQU 07H
X DB AH
IF(X*DB)<40H
Y DB BC
ELSE
Y DB AB
d.AB 0005
BC 0007
X 0005
Y 0007
```

3.suponga la siguiente macro

```
IF INST ELSE INST
    SUM REST DEF MACRO A,B,C
        MOV REG ,(A)
        IF (A<B) ABD REG,(B)
        ELSE SUB REG,(B)
        MOV(C), REG
    END MACRO
C.MOV REG,(X)
```

```
IF(X<Y)
ADD REG,(Y)
ELSE
SUB REG,(Y)
MOV(Z), REG
```

4.considere el sig programa

5.que linea es la entrada.

a.3.2

EXAMEN GUIA

Lenguajes, Compiladores y Sistemas Operativos

1. Consideré la siguiente gramática:

$$S ::= AB$$

$$A ::= a$$

$$A ::= BaB$$

$$B ::= bbA$$

¿Cuál de las siguientes frases es **FALSA**?

- A La longitud de toda cuerda producida por la gramática es par.
- B Ninguna cuerda producida por la gramática tiene un número impar de b's consecutivas.
- C Ninguna cuerda producida por la gramática tiene tres a's consecutivas.
- D Ninguna cuerda producida por la gramática tiene cuatro b's consecutivas.
- E Toda cuerda producida por la gramática tiene al menos tantas b's como a's.

2. ¿Cuáles de las siguientes frases son ciertas para un intérprete?

- I. Genera código objeto.
- II. Maneja variables de tipo estático.
- III. Maneja variables de tipo dinámico.

- A Ninguna
- B I
- C I y II
- D III
- E I, II y III

3. Una definición particular de “word” está dada por las siguientes reglas:

```
<word> ::= <letter> | <letter> <pairlet>| <letter> <pairdig>
<pairlet> ::= <letter> <letter> | <pairlet> <letter> <letter>
<pairdig> ::= <digit> <digit> | <pairdig> <digit> <digit>
<letter> ::= a | b | c | ... | z
<digit> ::= 0 | 1 | 2 | ... | 9
```

¿Cuáles de las siguientes entidades léxicas pueden ser derivadas a partir <word>?

- I. word
- II. words
- III. c22

- A Ninguna
- B I y II únicamente
- C I y III únicamente
- D II y III únicamente
- E I, II y III

4. ¿Qué es cierto respecto a la verificación estática y la verificación dinámica de tipos?
- I. En general los programas verificados estáticamente corren más rápido que los verificados dinámicamente.
 - II. La verificación dinámica es más útil para encontrar errores que la estática.
 - III. La verificación de tipos es parte del análisis semántico.
- A** Ninguna **B** I y II **C** II **D** I y III **E** I, II y III
5. El analizador léxico de Pascal lee carácter por carácter, desde un cierto punto, hasta que se da cuenta que reconoce un “token”. Supongamos que los “tokens” de Pascal son: identificadores, constantes, palabras clave (key words) y operadores.
- ¿Para cuál de las siguientes secuencias de caracteres el analizador léxico de Pascal puede determinar que ha visto el “token” completo sin ver el siguiente carácter?
- I. [
 - II. 3.1416
 - III. while
- A** I **B** II **C** I y II **D** I y III **E** II y III
6. ¿Cuál de las siguientes características de un lenguaje requiere el uso de “stacks” (pilas) en lugar de manejo de variables de tipo estático?
- A** Parámetros por referencia.
B Funciones de valor entera.
C Arreglos bidimensionales.
D “goto’s” arbitrarios.
E Rutinas recursivas.
7. Dos estaciones de trabajo están conectadas a una red local. Una de las estaciones tiene acceso a los archivos vía red de un servidor de archivos, el tiempo promedio para accesar de esta manera a una página de un archivo es de 0.1 seg.
- La otra estación de trabajo accesa los archivos de un disco local con el tiempo promedio de 0.05 seg. por página. Una compilación particular requiere de 30 seg. de tiempo de procesamiento más 200 accesos a páginas de un archivo.
- ¿Cuál es la proporción del tiempo total requerido para la compilación por la primera estación (sin disco) al tiempo total requerido por la segunda (con disco), suponiendo que el procesamiento no se traslape con accesos a los archivos?
- A** 1/1 **B** 5/4 **C** 5/3 **D** 1/2 **E** 3/1

8. Suponga que la protección de los archivos de un sistema está representado por una matriz A de derechos de acceso, donde $A(i, j)$ denota el conjunto de derechos que el usuario i tiene sobre el archivo j . Los usuarios están divididos en grupos y pueden pertenecer a más de un grupo. Existen tres modos distintos de accesar los archivos: (R)ead, (W)rite y (E)xecute.

El sistema tiene tres tipos de archivos: Correo, Texto y Binario. Considere el siguiente conjunto de políticas de seguridad:

- Cada usuario tiene acceso R y W a todos los archivos que le pertenecen, más acceso E a los archivos Binarios que le pertenecen.
- Usuarios en un grupo tiene acceso E a los archivos Binarios de todos los usuarios en ese mismo grupo, acceso R a los archivos Correo de todos los usuarios en ese mismo grupo, y R y W a los archivos Texto de todos los usuarios en ese mismo grupo.
- Un super usuario tiene acceso a todos los archivos del sistema como si fueran los archivos que le pertenecen.

María, Juan y Alicia son tres usuarios del sistema. María y Alicia están en el mismo grupo. Juan es el super usuario. ¿Cuál de las siguientes frases es **INCONSISTENTE** con las políticas anteriores?

- A Juan tiene acceso W a los archivos Correo de Alicia.
- B Juan tiene acceso E a los archivos Correo de Alicia.
- C María tiene acceso R a los archivos Texto de Alicia.
- D Juan tiene acceso E a los archivos Binarios de María.
- E María tiene acceso R a los archivos Correo de María.

9. ¿Cuáles de las siguientes frases son correctas?

- I. El cargador de arranque normalmente reside en ROM.
- II. El paso de parámetros por valor no afecta a los valores del parámetro formal en la subrutina invocada.
- III. Un ensamblador sólo permite el uso de macros y un compilador sólo permite el uso de subrutinas.

- A Ninguna
- B I y II
- C III
- D I
- E II y III

10. ¿Cuáles de las siguientes frases describen funciones de un sistema operativo?

- I. Administración de memoria y disco.
- II. Carga de programas compilados a memoria.
- III. Responde a interrupciones y “traps”.

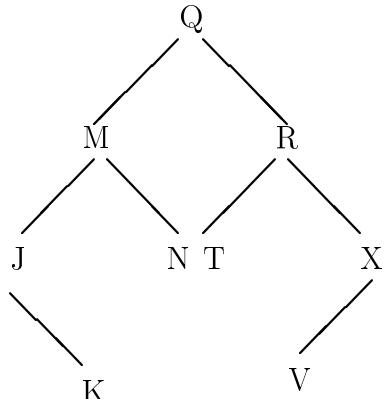
- A Ninguna
- B I y II
- C III
- D II y III
- E I, II y III

Estructuras de Datos y Programación

1. Suponga que utilizamos una lista simplemente ligada para representar una pila e indicamos la base de la pila con un apuntador nulo y guardamos un apuntador al tope de la pila. ¿Cuánto tiempo se tardan las operaciones meter (push) y sacar (pop)? Suponga que la pila tiene en un momento dado n elementos.

- A meter es $O(1)$, sacar es $O(n)$.
- B meter es $O(1)$, sacar es $O(n^2)$.
- C meter es $O(n)$, sacar es $O(n)$.
- D meter es $O(1)$, sacar es $O(1)$.
- E meter es $O(\log(n))$, sacar es $O(1)$.

2. ¿Qué nodo está fuera de lugar para que el árbol resultante sea un árbol binario de búsqueda?



- A J
- B M
- C T
- D R
- E X

3. Convertir de notación infija a notación posfija la expresión: $(A + B + C)/(D + E * F)$

- A $ABC + DEF * +/+$
- B $AB + C + DEF * +/$
- C $ABC ++DEF * /+$
- D $AB + C + DEF + */$
- E $ABC + D + EF * +/$

4. En una árbol binario de búsqueda balanceado por la altura, la diferencia entre las alturas de los subárboles derecho e izquierdo es 0 o 1. ¿Qué es verdadero para un árbol de este tipo que tiene N nodos?

- I. La búsqueda en el peor caso es proporcional al $\log(N)$.
- II. La búsqueda en el caso promedio es proporcional al $\log(N)$.
- III. La búsqueda en el mejor de los casos es proporcional al N .
- IV. La altura del árbol es logarítmica en el número de nodos.

- A I y III
- B II y III
- C II y IV
- D I, II y IV
- E I, III y IV

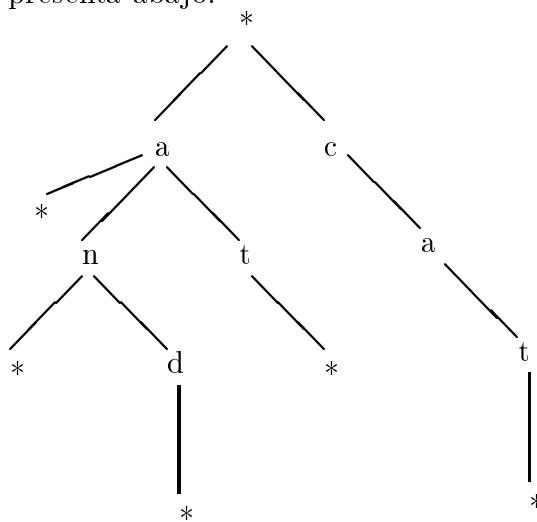
5. ¿Qué tipo de árboles de búsqueda están más orientados para trabajar en memoria secundaria y que particularmente emplean el concepto de página?

- A Árboles B
- B Árboles binarios
- C Árboles AVL
- D Árboles 2 – 3
- E Árboles de Huffman

6. Un “trie” es una clase de árbol que puede ser usado para representar palabras (i.e., cadenas de caracteres arbitrariamente largas pero finitas) como sigue:

1. Cada nodo se etiqueta con un carácter. La raíz y todas las hojas tienen por etiqueta el símbolo *.
2. Cada camino no vacío del “trie”, desde la raíz, representa la cadena de caracteres de la secuencia encontrada en dicho camino.
3. Cada nodo tiene a lo más una hoja entre sus hijos, cada hoja es el hijo más a la izquierda de su parente, y cada nodo no-hoja aparece, de izquierda a derecha, en el orden alfabético de su etiqueta.

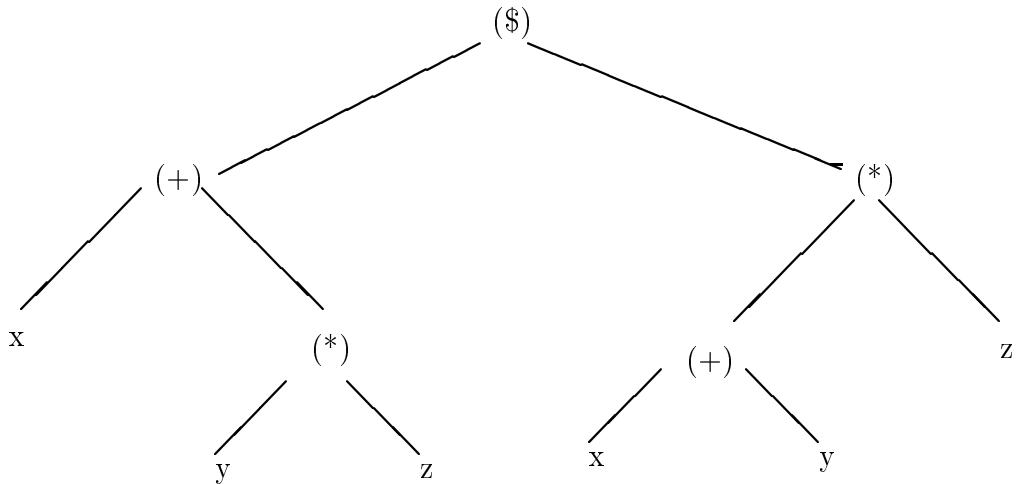
Dado el orden alfabético para los caracteres, cada conjunto de palabras determina de forma única un “trie”. Por ejemplo, el “trie” para “a”, “an”, “at”, “and” y “cat” se presenta abajo.



¿Cuántos nodos, incluyendo los *, hay en un “trie” para las palabras “do”, “dog”, “door” y “doors”?

- [A] 8 [B] 9 [C] 10 [D] 11 [E] 12

7. Considere el siguiente árbol binario:



\$ denota exponentiación. Si el árbol se recorre en preorden, ¿Cuál de las expresiones siguientes se obtiene?

- A** $+x * z\$ + xy * z$
- B** $*yz + x\$ + xy * z$
- C** $\$ + *xyz * +xyz$
- D** $\$ + x * yz * +xyz$
- E** Ninguna de las anteriores

8. Considere el siguiente código pseudo-Pascal:

```

var i, j : integer;
procedure P(k, m : integer);
begin
  k := k - m;
  m := k + m;
  k := m - k;
end;
i := 2;
j := 3;
P(i, j);
  
```

Si los dos parámetros de P se pasan por referencia, ¿Cuáles son los valores de i y j al final del fragmento de programa?

- A** $i = 0, j = 2$
- B** $i = 1, j = 5$
- C** $i = 2, j = 3$
- D** $i = 3, j = 2$
- E** Ninguno de los anteriores.

9. Consideré el siguiente código pseudo-Pascal:

```
p:= 1; k := 0;
while k < n do
begin
  p := 2 * p;
  k := k + 1;
end;
```

Para el fragmento de programa anterior, que involucra a los enteros p , k y n , cuál de las siguientes condiciones es invariante a iteraciones, i.e., que es verdadera al inicio de cada ejecución de la iteración y al final de la ejecución de la iteración:

- [A] $p = k + 1$
- [B] $p = (k + 1)^2$
- [C] $p = (k + 1)2^k$
- [D] $p = 2^k$
- [E] $p = 2^{(k+1)}$

10. ¿Cuántos renglones de salida produce el siguiente programa?

```
#include <stdio.h>
main()
{
    float sum = 0.0, j = 1.0, i = 2.0;
    while (i/j > 0.001)
    {
        j = j + j;
        sum = sum + i/j;
        printf(``%f\n'', sum);
    }
}
```

- [A] 0-9
- [B] 10-19
- [C] 20-29
- [D] 30-39
- [E] Más de 39

Matemáticas para la Computación

1. ¿Qué propiedades cumple la relación R definida sobre dos nodos x, y de un árbol binario del siguiente modo: x está relacionado con y si x es hijo o descendiente de y ?

- I. Reflexiva
- II. Simétrica
- III. Transitiva

A I **B** II **C** III **D** II y III **E** I y III

2. ¿Cuáles de los siguientes conjuntos de parejas (a, b) representan a una función?

- I. $\{(-1, 0), (1, 0), (2, 1), (1, 1)\}$
- II. $\{(2, 2), (1, 1), (3, 3), (0, 0)\}$
- III. $\{(1, 2), (2, 3), (3, 4), (4, 5)\}$

A I y II **B** II y III **C** I y III **D** Ninguno **E** I, II y III

3. ¿Cuáles de las siguientes formas sentenciales son tautológicas?

- I. $P \supset \neg P$
- II. $(P \supset Q) \supset Q$
- III. $(P \cdot Q) \supset P$

A I **B** II **C** III **D** Ninguna **E** I, II y III

4. Aplicando la inducción diga cuáles de los siguientes expresiones son verdaderas ($n \in \mathbf{N}$) :

II. $n^2 < 2^n$
 I. $\sum_{i=1}^n i^2 = \frac{n(n+1)(n+2)}{6}$

III. $\underbrace{1 + 4 + 7 + \dots + (3n - 2)}_{n \text{ elementos}} = \frac{n}{3n-1}$

- A** Ninguna **B** I y II **C** I y III **D** II y III **E** I, II y III

5. ¿Cuál es el mínimo de la función $f(x_1, x_2) = x_1 - 2x_2$ sujeta a las restricciones:

$$\begin{cases} x_1 - \frac{x_2}{2} \geq -2 \\ 0 \leq x_2 \leq 2 \\ x_1 \leq 0 \end{cases} ?$$

- A** 0 **B** -2 **C** -25 **D** 2 **E** -5

6. Hay 5 amigos que deciden sentarse en la mesa redonda. Cada uno de ellos elige al azar una silla entre 5 sillas alrededor de la mesa. ¿Cuál es la probabilidad de que 2 amigos, Carlos y Ricardo, sean vecinos?

- A** $\frac{2}{5}$ **B** $\frac{1}{5}$ **C** 1 **D** 2.01 **E** 0.1

7. ¿Cuánto vale $\lim_{x \rightarrow 0} \frac{2x^2}{\sin^2 x}$?

- A** ∞ **B** 0 **C** 2 **D** 1 **E** no existe

8. ¿Cuáles de las siguientes funciones representan la solución de la ecuación diferencial $y'' + 3y' + 2y = 0$?

- I. $\sqrt{x} + 1$
- II. $3e^{-x}$
- III. $2e^{-2x} - 4e^{-x}$

- [A]** I y II **[B]** I y III **[C]** II y III **[D]** Ninguna **[E]** I, II y III

9. El conjunto de soluciones del siguiente sistema de ecuaciones

$$\begin{cases} x + y + z = 1 \\ 2x - y + 2z = 0 \\ 3x + 3z = 2 \end{cases}$$

- [A]** Es un plano **[B]** Es vacío **[C]** Es una recta **[D]** Es un punto **[E]** Es finito

10. Sean $A=\{2, 3, 4\}$ y $B=\{-1, 0, 1, 2\}$ dos conjuntos. ¿Cuántas funciones $f : A \rightarrow B$ hay?

- [A]** 3^4 **[B]** 4^3 **[C]** 3 **[D]** 4 **[E]** $3 \cdot 4$

Diseño Lógico y Arquitectura de Computadoras

1. Para la siguiente tabla de transición encuentre las ecuaciones booleanas de f_1 y f_2 usando flip-flops tipo D.

Estado Presente		Entradas		Estado Siguiente		Salidas	
f1	f2	x	y	f1	f2	S1	S2
0	0	0	0	0	1	0	1
0	0	0	1	1	0	0	1
0	0	1	0	0	0	0	1
0	0	1	1	1	1	0	1
0	1	*	1	0	0	0	0
0	1	*	0	0	1	0	0
1	0	*	*	0	0	1	0
1	1	*	*	1	0	1	1

A
$$\begin{aligned}f_1 &= f_1 f_2 y + f_1 f'_2 x + f'_1 f'_2 y \\f_2 &= f_1 x y + f'_1 y + f'_1 f_2\end{aligned}$$

D
$$\begin{aligned}f_1 &= f_1 f_2 + f'_1 f_2 S_1 + f_1 f_2 x \\f_2 &= f_2 x y + f'_1 f'_2 x y' + f'_1 f_2 x' y\end{aligned}$$

B
$$\begin{aligned}f_1 &= f'_1 f'_2 y + f_2 x' y' + f_2 x y' + f_1 f_2 \\f_2 &= f'_1 x' y' + f'_1 f'_2 x y + f'_1 f_2 x y'\end{aligned}$$

E
$$\begin{aligned}f_1 &= f_2 y + f_1 x \\f_2 &= f_1 f_2 + f_2 f'_1\end{aligned}$$

C
$$\begin{aligned}f_1 &= f'_1 f_2 y + f_2 x y + f_1 f_2 \\f_2 &= f_1 x' y + f_2 x y' + x y\end{aligned}$$

2. Del problema anterior encuentre las ecuaciones booleanas de las salidas $S_1 S_2$.

A $S_1 = f_1, \quad S_2 = f'_1 f'_2 + f_1 f_2$ **D** $S_1 = f_1 f_2 + y, \quad S_2 = f_1 f_2$

B $S_1 = f'_1, \quad S_2 = f_1 f_2 + f_1 f'_2$ **E** $S_1 = f'_1 f'_2 + y, \quad S_2 = f'_1 f_2$

C $S_1 = f_1 x' + f_2 y, \quad S_2 = f_1 x + f_2$

3. Efectúe la siguiente suma de números binarios de 8 bits usando complemento a dos: 0001 0001 + 1110 0111 dejando el resultado en decimal.

A 240 **B** 112 **C** -9 **D** 8 **E** -8

4. Cuántas líneas de un bus de direcciones de 16 bits se necesitan para direccionar una memoria de 4k?

A 11 **B** 13 **C** 15 **D** 14 **E** 12

5. Un microprocesador responde a una atención de interrupciones insertando en la pila no sólamente la dirección de regreso sino también el contenido del registro de estados del microprocesador. Si el apuntador de pila tiene el valor F8F0 antes de que el microprocesador sea interrumpido, cuál será su valor después de la interrupción?
- A** F8F2 **B** F8EE **C** F8F1 **D** F8EF **E** F8ED
6. Un ALU tiene dos registros A y B de 8 bits. El registro A se modifica de la siguiente forma: $A = A \text{ OPER } B$ en donde OPER puede ser una de las siguientes operaciones: AND, OR, XOR, ADD, SUB. A tiene el valor 1101 1001. Cuál debe ser la operación que debe ejecutar el ALU para que A tenga el valor 1110 0100 después de la operación?
- A** AND **B** SUB **C** XOR **D** ADD **E** OR
7. Para el problema anterior, cuál debe ser el valor de B?:
- A** 1110 0100 **B** 0001 0111 **C** 0000 1011 **D** 1101 1001 **E** 1000 1000
8. Cuál es la diferencia principal entre un microprocesador RISC y un CISC?
- A** el número de ALU's internos
B su capacidad de direccionamiento de memoria
C su velocidad
D su tamaño
E el número de instrucciones que pueden ejecutar
9. Acceso directo a memoria se refiere a:
- A** La habilidad de un microprocesador para leer o escribir datos en cualquier localidad de memoria.
B La habilidad de un microprocesador para leer o escribir datos en cualquier localidad de memoria a la misma velocidad.
C Dispositivos de entrada/salida transfieran información directamente a la memoria sin intervención del microprocesador.
D Una memoria especialmente diseñada para hacer una transferencia rápida de datos.
E El microprocesador es el único que directamente accesa la memoria.
10. Encuentre la representación en complemento a dos del siguiente número en hexadecimal: F3A1.
- A** 0A5E **B** FA5E **C** 0A57 **D** 0A5E **E** F321

Algoritmos, Autómatas y Lenguajes Formales

1. La definición de algoritmo **NO** requiere que:

- A** Su ejecución termine después de ejecutar un número finito de pasos
- B** Cada instrucción esté claramente especificada y ejecute en tiempo finito
- C** Siempre encuentre la respuesta correcta
- D** El tiempo de ejecución de cada instrucción esté acotado por una función lineal dada
- E** Todas las anteriores

2. La siguiente máquina de Turing tiene como propósito sacar una copia de una cadena binaria, i.e., de una configuración inicial:

...###10001###...

se desea pasar a la configuración final:

...###10001#10001##...

La idea es: para cada dígito a copiar, se marca éste con una “x” o “y”, se recorre la cinta hasta encontrar espacio en el extremo derecho, se copia el dígito, se regresa a la izquierda y se restituye el dígito original. Por ejemplo:

...###10001###...
...###x0001###...
...###x0001#1#...
...###10001#1#...
...###1y001#1#...
...

El programa es el siguiente (la cabeza se encuentra leyendo el dígito más significativo al inicio de la operación), con q1 como estado inicial, y q9 final:

Estado	Leyendo Símbolo				
	0	1	x	y	#
q1	(q2,x,R)	(q4,y,R)			(q8,#,R)
q2	(q2,0,R)	(q2,1,R)			(q3,#,R)
q3	(q3,0,R)	(q3,1,R)			(q6,0,L)
q4	(q4,0,R)	(q4,1,R)			(q5,#,R)
q5	(q5,0,R)	(q5,1,R)			(q6,1,L)
q6	(q6,0,L)	(q6,1,L)			(q7,#,L)
q7	(q7,0,L)	(q7,1,L)	?	(q1,1,R)	
q8	(q8,0,L)	(q8,1,L)			(q9,#,L)
q9	-	-	-	-	-

En la tabla anterior, la notación (q, s, D) indica que “estando en un estado q , graba el símbolo s en la posición en la que está la cabeza, y después mueve la cabeza en la dirección indicada por D , ya sea a la izquierda ($D = L$) o a la derecha ($D = R$)”.

Identificar la transición faltante de q7 leyendo una “x”:

- A** (q1,0,R)
- B** (q2,0,R)
- C** (q1,1,R)
- D** (q1,1,L)
- E** Ninguna de las anteriores

3. Para n suficientemente grande, la notación $g(n) = O(f(n))$ para indicar la complejidad de $g(n)$ establece que:

- A** Las medidas de complejidad de g y f son equivalentes
- B** Existe una constante k tal que $|f(n)| \leq k \cdot |g(n)|$
- C** Existe una constante k tal que $|g(n)| \leq k \cdot |f(n)|$
- D** Existe una constante k tal que $g(n) \cdot f(n) \leq k$
- E** Ninguna de las anteriores

4. Para cualquier expresión regular R :

- A** Es posible a veces construir un aceptador finito M tal que $L(M) = L(R)$
- B** Es siempre posible construir un aceptador finito M tal que $L(M) = L(R)$
- C** Es imposible construir un aceptador finito M tal que $L(M) = L(R)$
- D** Es siempre posible construir un aceptador finito M tal que $L(M) = L(R)$, aunque el aceptador tenga que ser no determinístico
- E** Ninguna de las anteriores

5. Dado el siguiente algoritmo para ordenar un vector a de N elementos de tipo T :

```
procedure ordena( T a[N] )
begin
    int i, j, min;
    for i:= 1 to N-1
    begin
        min:= i;
        for j:= i+1 to N
        begin
            if (a[j] < a[min]) then
                min:= j
            end;
            intercambia(a,min,j)
        end
    end
end
```

y suponiendo que el procedimiento `intercambia` ejecuta en tiempo constante, su complejidad puede expresarse como:

- A $O(N)$
- B $O(i + j)$
- C $O(N^2)$
- D $O(N^3)$
- E Ninguna de las anteriores

6. El algoritmo de ordenamiento de un vector a de longitud n conocido como bucket sort tiene la forma siguiente (donde B es un vector auxiliar de listas ligadas):

1. int i, n
2. $n := \text{longitud}(a)$
3. for $i := 1$ to n
4. inserta $A[i]$ en la lista $B[\text{floor}(n.A[i])]$
5. for $i := 0$ to $n-1$
6. ordena lista $B[i]$ con metodo de insercion
7. concatena listas $B[0], B[1], \dots, B[n-1]$

Si sabemos que el ciclo de los pasos 3,4 ejecuta en promedio en tiempo n , que el paso 7 puede hacerse también en tiempo lineal, y que el algoritmo completo ejecuta en promedio en tiempo lineal, ¿qué podemos concluir respecto al paso 6?

- A Que debe ejecutarse en promedio en tiempo constante
- B Que debe ejecutarse en promedio en tiempo cuadrático
- C Que debe ejecutarse en promedio en tiempo lineal
- D Que debe ejecutarse en promedio en tiempo $\lg n$
- E No se puede concluir nada

7. Dadas las operaciones de: (i) Unión, (ii) Concatenación, y (iii) Intersección, la clase de los lenguajes libres de contexto es cerrada bajo:

- A** (i), (ii), y (iii)
- B** (i) y (ii)
- C** (iii)
- D** (ii) y (iii)
- E** Ninguna de las anteriores

8. La gramática libre de contexto (el signo ‘ ϵ ’ representa a la cadena vacía):

$$\begin{aligned}S &\longrightarrow AB|a \\A &\longrightarrow a \\B &\longrightarrow \epsilon\end{aligned}$$

es equivalente a la gramática con producciones:

- A** $S \longrightarrow AB, A \longrightarrow a, B \longrightarrow b$
- B** $S \longrightarrow a$
- C** $S \longrightarrow A$
- D** $S \longrightarrow A, A \longrightarrow a|B, B \longrightarrow aA$
- E** Ninguna de las anteriores

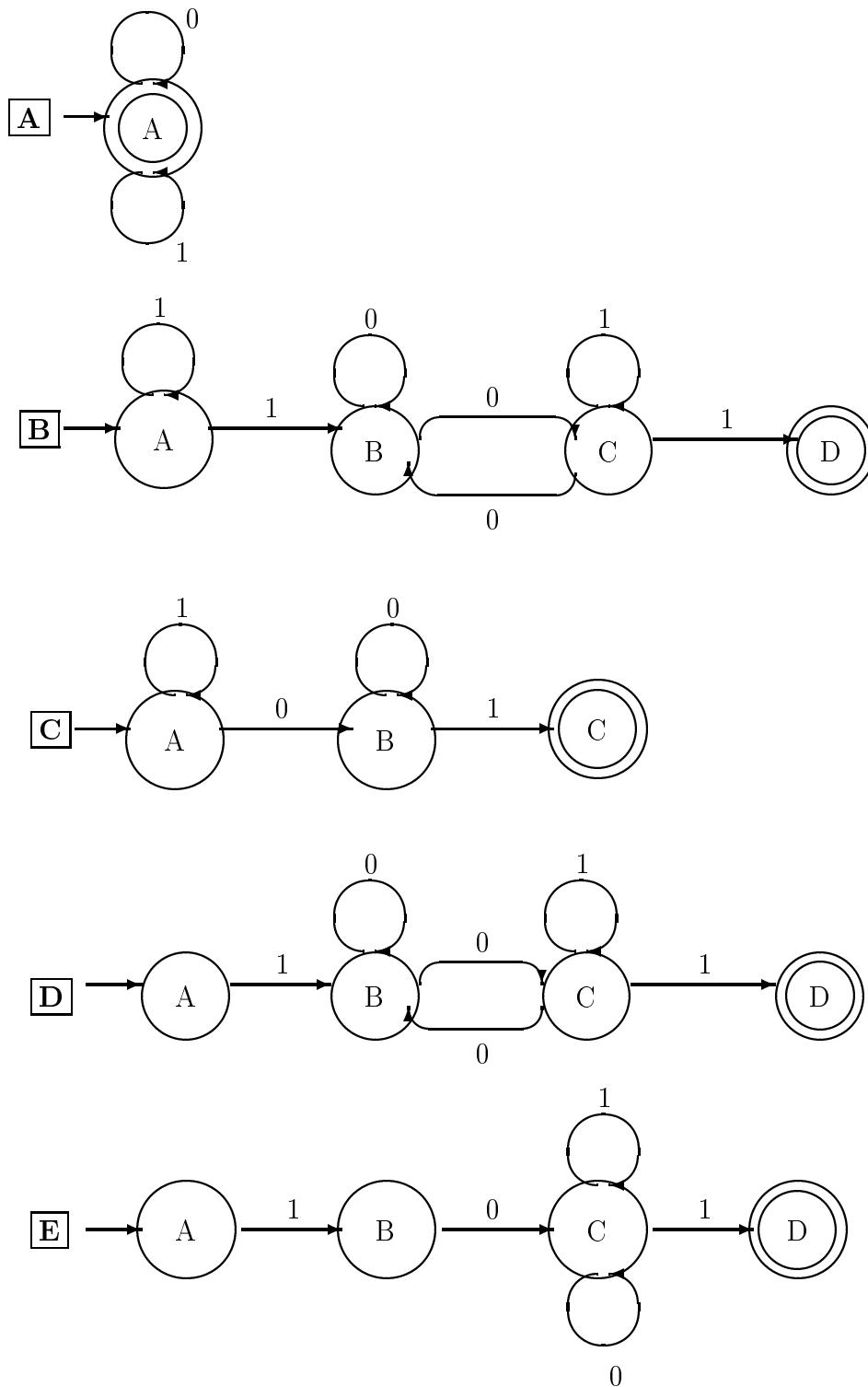
9. Los lenguajes sensibles al contexto pueden ser caracterizados como aquellos que:

- A** Son reconocibles por un autómata de pila (*pushdown automaton*)
- B** Requieren de una máquina de Turing no determinística con cinta de longitud limitada (*linear bounded automaton*) para su reconocimiento
- C** Están formados por gramáticas que tienen reglas de la forma $X \longrightarrow \alpha$, para X no terminal y α un cadena de terminales y no terminales
- D** Requieren de una máquina de Turing no determinística sin limitaciones en la cinta para su reconocimiento
- E** Ninguna de las anteriores

10. La gramática lineal:

$$\begin{aligned}\Sigma &\longrightarrow A \\A &\longrightarrow 1B \\B &\longrightarrow 0B|0C \\C &\longrightarrow 0B|1C|1\end{aligned}$$

corresponde al autómata (ver siguiente figura):



Se lista una serie de libros que los estudiantes pueden usar como referencia. Desde luego, no se presupone que los estudiantes hayan leido *todos* los libros, ni que conocen *todo* el material a detalle. Se pretende más bien que los alumnos tengan una gama amplia de referencias que consultar.

Lenguajes, Compiladores y Sistemas Operativos.

- **Temario.**

1. Lenguaje de máquina.
2. Ensambladores, Macros, Ligadores, Compiladores, Intérpretes.
3. Sistemas operativos.
4. Definición formal de la gramática de un lenguaje de programación.

- **Bibliografía.**

- John J. Donovan, *Systems Programming*, Mc Graw Hill, 1981.
- Wirth Nicklaus, *Algoritmos y Estructuras de Datos*, Prentice Hall, 1986.

- **Respuestas:** 1 - D, 2 - D, 3 - D, 4 - D, 5 - A, 6 - E, 7 - B, 8 - B, 9 - D, 10 - E.

Estructuras de Datos y Programación.

- **Temario.**

1. Conceptos de tipos de datos abstractos.
2. Estructuras de datos lineales (listas, pilas, colas,etc.)
3. Estructuras arborescentes.
4. Tablas de dispersión.
5. Algoritmos de ordenamiento y búsqueda.
6. Recursión.
7. Programación.

- **Bibliografía.**

- Alfred V. Aho, John E. Hopcroft, y Jeffrey D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley. 1974.
- Nicklaus Wirth, *Algoritmos y Estructuras de Datos*, Prentice Hall, 1986.

- **Respuestas:** 1 - D, 2 - C, 3 - B, 4 - D, 5 - A, 6 - D, 7 - D, 8 - D, 9 - D, 10 - B.

Matemáticas para la Computación.

- **Temario.**

1. Algebra.
2. Lógica Matemática.
3. Geometría Analítica.
4. Cálculo Diferencia e Integral.
5. Ecuaciones Diferenciales.
6. Probabilidad y Estadística.

- **Bibliografía.**

- Copi I. M. *Lógica Simbólica*, pp.1-50.
- Swokowski E. W. *Algebra Universitaria*, Capítulo 7.
- Granero Rodríguez F. *Algebra y Geometría Analítica*, Capítulos 1,2,4.
- Stein S. K., Alonso Linares A., *Cálculo con Geometría Analítica*, Capítulo 2.
- Rainville E. D., Bedient P.E. *Ecuaciones Diferenciales*, Capítulos 5,6.
- Lippman S. A. *Elementos de Probabilidades y Estadística*, Capítulo 1.

- **Respuestas:** 1 - C, 2 - B, 3 - C, 4 - D, 5 - E, 6 - A, 7 - C, 8 - C, 9 - B, 10 - B.

Diseño Lógico y Arquitectura de Computadoras

- **Temario.**

1. Fundamentos de Sistemas Digitales.
 - (a) Operaciones Booleanas.
 - (b) Simplificación de Ecuaciones Booleanas.
 - (c) Diseño de Sistemas Secuenciales.
 - (d) Diseño Básico de Componentes Principales (ALU's, CPU's, etc).
2. Códigos de Representación (binario, octal, hexadecimal,etc).
 - (a) Operaciones Básicas (ADD, OR, XOR, Complementos, etc).

- **Bibliografía.**

- M. Morris Mano, *Lógica Digital y Diseño de Computadoras*, Prentice Hall Hispanoamericana, 1993.
- Ronald J. Tocci, Frank J. Ambrosio, Lester P. Laskowski, *Microprocessors and microcomputers : hardware and software*, Prentice Hall, 1997.
- Hayes, John Patrick, *Digital system design and microprocessors*, McGraw-Hill, 1984.

- **Respuestas:** 1 - C, 2 - A, 3 - E, 4 - E, 5 - B, 6 - D, 7 - C, 8 - E, 9 - C, 10 - C.

Algoritmos, Autómatas y Lenguajes Formales.

- **Temario.**

1. Algoritmos
 - (a) Problemas computables y no computables (*intractable problems*)
 - (b) Máquinas de Turing
 - (c) Clases de Complejidad (de lineal a NP)
 - (d) Notación de “O mayúscula” (*big O*)
 - (e) Análisis práctico de complejidad de algoritmos
 - (f) Complejidad de problemas clásicos de ordenamiento y búsqueda
2. Autómatas y Lenguajes Formales
 - (a) Autómatas finitos y sus lenguajes
 - (b) Autómatas de pila (*pushdown*) y sus lenguajes
 - (c) Aplicaciones
 - (d) Jeraquía de Chomsky
 - (e) Cerradura y complejidad de operaciones sobre gramáticas (unión, intersección, reconocimiento, etc.)

- **Bibliografía.**

- Alfred V. Aho, John E. Hopcroft, y Jeffrey D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley. 1974.
Para secciones 1.a, 1.b, 1.c, 1.e, 1.f
- Thomas H. Cormen, Charles E. Leiserson, y Ronald L. Rivest. *Introduction to Algorithms*. MIT Press. 1990.
Para secciones 1.c, 1.d, 1.e, 1.f
- John E. Hopcroft y Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley. 1979.
Para sección 2
- Robert Sedgewick. *Algorithms in C++*. Addison-Wesley. 1992.
Para secciones 1.d, 1.e

- **Respuestas:** 1 - D, 2 - A, 3 - C, 4 - B, 5 - C, 6 - A, 7 - B, 8 - B, 9 - B, 10 - D.