**U1: INTRODUCCION A LA TEORIA GENERAL DE SISTEMAS**

**Definición de sistemas**

* Conjunto de elementos que interactúan entre sí para lograr un objetivo común.
* *Teoría general de sistemas*:

A través del análisis de las totalidades (conjunto de cosas de una especie) y las interacciones internas de éstas y las externas con su medio, se permite la explicación de los fenómenos de la realidad. Además hace posible la predicción de la conducta futura de esa realidad.

Para abordar la realidad hay que tener una visión integral y total. El sistema total tiene una conducta que no puede ser explicada o prevista por cada una de las partes en forma independiente. Para explicar la realidad hay que pasar por todas las totalidades.

* Conjunto de elementos, con sus propiedades, la interacción entre los elementos y entre los elementos con el entorno, para lograr un objetivo común.

**Componentes básicos de los sistemas**

* *Ambiente/entorno*: Objetos que se relacionan con el sistema pero que no las puede controlar. No se relacionan entre sí gráficamente.
* *Entradas*: Elementos del entorno que ponen en funcionamiento al sistema.
* *Procesos*: Elementos internos.

El objetivo determina cuáles elementos del sistema se incluyen gráficamente.

**Clasificación de los sistemas**

* *Naturales*: Sistema nervioso, solar, de lenguaje.
* *Artificiales*: Construidos por el hombre. Sistema contable, de facturación, empresarial.
* *Abstractos*: Todos los elementos son conceptos, cosas no tangibles.
* *Concretos*: Por los menos dos de los elementos son objetos tangibles.
* *Abiertos*: Hay interacción con algún elemento de otro sistema (entorno).
* *Cerrados*: No tiene relación con ningún sistema externo.

**Propiedades de Ackoff**

1. Las propiedades de una parte hacen a las propiedades del sistema.
2. Las propiedades del sistema no dependen solo de una parte, sino también de las interacciones con el resto de los elementos.
3. Cada una de las partes del sistema son subsistemas y responden a las dos propiedades anteriores.

**U2: SISTEMAS DE INFORMACIÓN**

**Datos**:Son hechos recolectados a partir de observaciones o mediciones de la realidad.

**Información**: Es un conjunto de datos procesados y relacionados de manera funcional para un fin previsto.

**Sistemas de información:** Sistema con datos como entradas e información como salida.

Un sistema de información es un conjunto integrado de procesos orientados a la gestión y administración de un conjunto datos para cubrir una necesidad u objetivo. Su finalidad es administrar y procesar los datos para dar lugar a la información más elaborada, que se distribuye adecuadamente en una organización en función de sus objetivos.

**Categorías de SI**

Los SI se desarrollan para distintos fines, dependiendo de las necesidades de los usuarios y la empresa.

* Sistemas para el procesamiento de transacciones: Se desarrollaron para procesar grandes cantidades de información para las transacciones de negocios rutinarias, por lo que mejoran el procesamiento de transacciones de una organización. Una transacción es cualquier actividad que afecta a toda la organización. Estos sistemas brindan más velocidad, eficacia y exactitud.
* Sistemas de automatización de oficinas: Brindan apoyo a las personas que trabajan con datos para analizar la información y transformar o manipular los datos antes de compartirlos a través de la organización.
* Sistemas de trabajo de conocimiento: Brindan apoyo a profesionales como científicos, ingenieros y médicos, ayudándoles a crear conocimiento y a integrarlo a su organización o la sociedad.
* Sistemas de información administrativa: Estos sistemas incluyen el procesamiento de transacciones. Brindan soporte a los usuarios para realizar un espectro más amplio de tareas organizacionales que los sistemas de procesamiento de transacciones, incluyendo los procesos de análisis y toma de decisiones. Ayudan a los directivos a tomar decisiones estructuradas y resolver problemas. Se diseñan para que preparen en forma periódica reportes estructurados para el soporte de decisiones.
* Sistema para el soporte de decisiones: Son similares al sistema de información administrativa debido a que ambos dependen de una BD como fuente de datos. Este sistema está más enfocado a brindar respaldo a la toma de decisiones en todas sus fases, aunque la decisión aún corresponde exclusivamente del usuario. Los sistemas de soporte de decisiones se ajustan más a la persona o el grupo usuario. Ayudan a los directivos que deben tomar decisiones no muy estructuradas ni tan recurrentes. Deben ser más flexibles que los anteriores porque no siempre se sabe con anticipación la información necesaria.
* Inteligencia artificial y sistemas expertos: Capturan y utilizan el conocimiento de un experto para solucionar un problema específico en una organización. Se encargan de encontrar la mejor solución para un problema o una clase especifica de problemas.
* Sistemas de soporte de decisiones en grupo y sistemas de trabajo colaborativo asistido por computadora: Tienen el propósito de unir a un grupo en la búsqueda de la solución a un problema con la ayuda de diversas herramientas como los sondeos, los cuestionarios, las lluvias de ideas y la creación de escenarios.
* Sistemas de apoyo para ejecutivos: Apoyan a los ejecutivos en la toma de decisiones a nivel estratégico. Los ayudan a organizar sus actividades relacionadas con el entorno externo mediante herramientas gráficas y de comunicaciones. Amplían y apoyan las capacidades de los ejecutivos al darles la posibilidad de comprender sus entornos.

**Participantes de los sistemas**

1. Usuario: Es quien va a interactuar con el sistema final. Se le harán entrevistas y tendrá la decisión de aprobar el sistema. Es muy importante para su desarrollo exitoso.
2. Administradores: Puede haber diferentes administradores. Se preocupan por el desarrollo y la utilización de todos los recursos de la organización, el planeamiento estratégico y características financieras. La principal interacción que tienen con el analista es por los recursos de dinero, tiempo y personal.
3. Auditores de calidad: No están siempre, el objetivo es asegurar que el sistema se desarrolle de acuerdo con diversos estándares o normas externas.
4. Analista de sistemas: Persona clave porque interactúa con los demás participantes. Requiere habilidad para definir diagramas y facilidad para entrevistar y mediar con los usuarios. Identifica cómo la tecnología puede mejorar los procesos del negocio, diseña los sistemas de información y los procesos nuevos, y se asegura de cumplir todos los reglamentos de documentación.
5. Diseñadores de sistemas: Recibe los resultados del trabajo de análisis y lo transforma en un diseño arquitectónico que proveerá las especificaciones a los programadores, asociándolo con la tecnología disponible.
6. Programador: Codifican los requerimientos del diseñador en un lenguaje de programación.
7. Personal de operaciones: Puede estar en sistemas grandes donde participan distintos sectores de la empresa. Reciben el sistema ya probado y se encargan dar soporte en cuestiones de instalación.

**Tareas del analista de sistemas**

El analista de sistemas evalúa de manera sistemática el funcionamiento de un negocio mediante el examen de la entrada, el procesamiento de datos y su consiguiente [producción](http://www.monografias.com/trabajos54/produccion-sistema-economico/produccion-sistema-economico.shtml) de información, con el propósito de mejorar los [procesos](http://www.monografias.com/trabajos14/administ-procesos/administ-procesos.shtml#PROCE) de una organización. Interactúa con los demás participantes. Debe tener capacidad de abstracción y de análisis, ser disciplinado y poder coordinar a personas y a recursos variados para llevar a cabo los proyectos.

Requiere conocimientos relacionados con las técnicas de análisis de SI:

* Conocimiento del paradigma tradicional de la ingeniería del SW y del tradicional ciclo de vida del SW en [cascada](http://es.wikipedia.org/wiki/Desarrollo_en_cascada).
* Modelado funcional: [Diagrama de flujo de datos](http://es.wikipedia.org/wiki/Diagrama_de_flujo_de_datos), [diagrama de estado](http://es.wikipedia.org/w/index.php?title=Diagrama_de_estado&action=edit&redlink=1), etc.
* [Modelado de datos](http://es.wikipedia.org/wiki/Modelo_de_datos) y sus técnicas: [Diagrama entidad-relación](http://es.wikipedia.org/wiki/Diagrama_entidad-relaci%C3%B3n), [modelo relacional](http://es.wikipedia.org/wiki/Modelo_relacional), etc.
* Conocimiento de la tecnología: [arquitectura de software](http://es.wikipedia.org/wiki/Arquitectura_software), [bases de datos](http://es.wikipedia.org/wiki/Bases_de_datos), etc.

Los principales roles del analista de sistemas son:

* El analista de sistemas como consultor: Para las personas y sus empresas y pueden llegar a contratarlo específicamente para lidiar con las cuestiones relacionadas con los SI dentro de la empresa.
* El analista de sistemas como experto de soporte: Se basa en su experiencia profesional sobre HW y SW y su uso en los negocios. De esta manera, descubre detalles y documenta la política de negocio que pudiera existir en un sistema.
* El analista de sistemas como agente de cambio: Para la empresa, ya sea interno o externo. Está presente e interactúa con los usuarios y la empresa durante un periodo extendido. Es una persona que actúa como catalizador para el cambio, desarrolla un plan de cambio y trabaja con otros para facilitarlo.
* El analista como mediador: A menudo se encuentra en medio de usuarios, administradores, programadores, auditores y otros participantes, los cuales frecuentemente están en desacuerdo entre sí. Es tarea del analista imponer su propia opinión respecto a cómo debe ser el sistema o cuáles funciones debe cumplir. Además, debe ser capaz de obtener un consenso, lo que requiere del delicado arte de la diplomacia y la negociación.

**Ciclo de vida de un proyecto**

Se define como la metodología tradicional para desarrollar un SI, el cual divide el proceso de desarrollo del sistema en fases formales para el análisis y diseño del mismo. Estas fases constan en un conjunto de actividades que pueden llevarse a cabo por separado o no ya que varias pueden ocurrir al mismo tiempo y repetirse.

Una *metodología* consta de una serie de pasos bien definidos para desarrollar un sistema.

Tiene tres objetivos principales:

* Definir las actividades a llevarse a cabo.
* Lograr congruencia en la multitud de proyectos de desarrollo de una misma organización.
* Proporcionar puntos de control y revisión de las decisiones sobre continuar o no un proyecto.

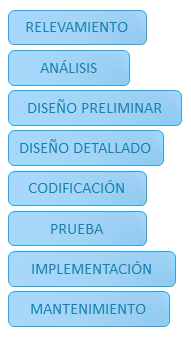
Organiza las actividades del administrador aumentando la probabilidad de que se aborden los problemas en el momento adecuado. Determinar qué ciclo de vida de proyecto utilizar es tarea del analista.

El ciclo de vida comprende seis fases:

* Investigación preliminar: La necesidad de recibir ayuda de un sistema de información puede surgir por diversas razones; sin importar cuales sean éstas, el proceso se inicia siempre con la petición de una persona.
* Determinación de los requerimientos: Comprensión de todas las fases importantes de la empresa que se encuentra bajo estudio. Los investigadores, al trabajar con los empleados y administradores, deben saber los procesos de una empresa para dar respuesta a las siguientes preguntas claves: ¿Qué es lo que hace? ¿Cómo se hace? ¿Con qué frecuencia se hace? ¿Qué tan grande es el volumen de transacciones o decisiones? ¿Cuál es el grado de eficiencia con el que se efectúa las tareas? ¿Existe algún problema?
* Diseño del sistema: Establece la forma en la que el sistema efectuará las obligaciones descriptas durante la fase de análisis. También se lo llama diseño lógico.
* Desarrollo del software: Los encargados de desarrollar programas pueden instalar SW comprado a terceros o escribir programas diseñados a la medida de la solicitud. La decisión depende del costo de cada alternativa, del tiempo disponible para escribir el programa y de la disponibilidad de los programadores.
* Prueba del sistema: Consiste en probar el sistema de manera experimental para comprobar si el SW no tiene fallas, es decir, se trata de que el sistema llegue a funcionar de acuerdo con las especificaciones y en la forma en que los usuarios esperen que lo haga.
* Implantación y evaluación: La implantación es el proceso de instalar nuevo equipo, preparar a los usuarios para usar el sistema, instalar la aplicación y construir todos los archivos de datos necesarios para utilizarla. Dado que las organizaciones y los usuarios cambian con el paso del tiempo, es necesario evaluar el sistema periódicamente.

**CICLO DE VIDA CLÁSICO**

Características

* + *Implementación ascendente*: Se espera que los programadores lleven a cabo las pruebas modulares, luego las de los subsistemas y luego las del sistema propiamente dicho.
* La eliminación de fallas suele ser extremadamente difícil en las últimas etapas de prueba.
  + *Progresión Secuencial*:Sus fases suceden secuencialmente, no se comenzaba con una etapa hasta que no se terminaba con la etapa anterior.
* No permitía el tratamiento de fenómenos naturales. Si había que cambiar algo en el sistema, también había que cambiar la documentación.

El cliente participa sólo en el relevamiento (objetivo). Lo siguiente que ve es la implementación.

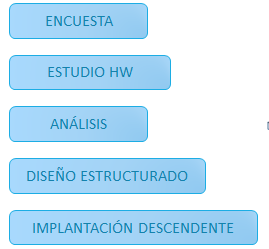
Ventajas:

* Apto para equipos débiles y productos maduros, se requiere de menos capital y herramientas para que funcione de manera óptima.
* Fácil de implementar y entender. Modelo conocido y utilizado con frecuencia.
* Creación de documentos.
* Metodología de trabajo efectiva: Definir antes que diseñar, diseñar antes que codificar.
* Su simple organización hace que resulte sencilla la explicación a usuarios.

Desventajas:

* Un proyecto rara vez sigue una secuencia lineal, esto crea una mala implementación del modelo. Es difícil para el cliente establecer explícitamente todos los requisitos al comienzo del proyecto.
* El proceso de creación del SW tarda mucho tiempo ya que debe pasar por el proceso de prueba y hasta que el software no esté completo no se opera.
* Cualquier error de diseño detectado en la etapa de prueba conduce al rediseño y nueva programación del código afectado, aumentando los costos del desarrollo.
* Una etapa determinada no se puede llevar a cabo a menos de que se haya culminado la etapa anterior.
* Se debe documentar cada etapa detalladamente.

**CICLO DE VIDA SEMIESTRUCTURADO**

Características

* + *Secuencia ascendente de codificación*: Implementación de arriba hacia abajo. Los módulos de alto nivel se codifican y prueban primero, seguidos por los de bajo nivel más detallados.
  + *Implementación descendente*: Ofrece retroalimentación entre el proceso de implementación y el de análisis.

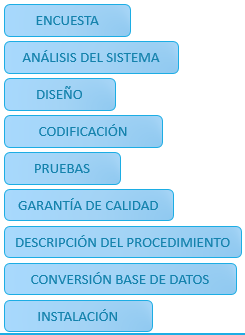
No duró mucho.

Ventajas:

* Permite poner en ejecución paralelamente parte de la codificación y de las pruebas.
* Mejoró la forma de documentar el proceso usando herramientas gráficas haciendo aún más rápido el proceso.
* Permite la retroalimentación entre la codificación, la prueba y la eliminación de fallas.

Desventajas:

* Los diseñadores tienen poco contacto con los analistas y no tienen contacto con el usuario.
* Se sigue perdiendo tiempo con la documentación.

**CICLO DE VIDA ESTRUCTURADO**

Características

* + Reunió lo mejor e indicó cómo trabajar. Es el que más se utiliza.
  + *Implementación radical*: Se realizan las actividades en paralelo, generalmente se usa cuando los usuarios no están seguros de lo que quieren.
  + *Implementación conservadora*: No se comienza con una actividad hasta que no se termina con la anterior, generalmente se utilizan para proyectos muy grandes que no son muy cambiantes.

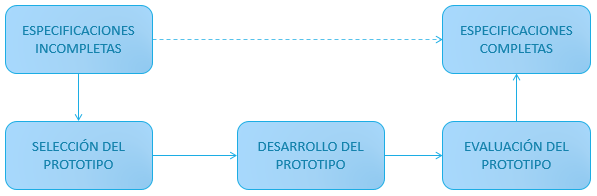
Ventajas:

* Las actividades están conectadas de manera tal que puedan llevarse a cabo en paralelo, lo que acelera el proceso. (análisis, diseño, codificación y pruebas)
* Permite realizar entregas parciales al usuario.
* Permite comprender de manera completa sistemas grandes y complejos.
* Detección de fallas más rápido.

Desventajas:

* El desarrollo de diagramas y esquemas es una tarea que consume mucho tiempo, sobre todo si el sistema es grande y complejo.
* Si el equipo de desarrollo no está bien organizado, puede que existan problemas de coordinación y se alargue el plazo de desarrollo por la gran cantidad de tapas que posee.

**CICLO DE VIDA PROTOTIPO**

Características

* Consiste en capturar un conjunto inicial de necesidades e implantarlas rápidamente con la intención de expandirlas y refinarlas iterativamente al ir aumentando la comprensión que tiene el usuario del sistema y quien lo desarrolla.
* Generalmente se utiliza con usuarios que no saben lo que quieren o que no trabajaron nunca con SI.
* Primero se hacen las pantallas y se las muestra al usuario, después se lleva a cabo la implementación (codificación).
* Consiste en:

1. Identificar los requerimientos de información que el usuario conoce junto con las características necesarias del sistema.
2. Desarrollar un prototipo que funcione.
3. Utilizar el prototipo anotando las necesidades de cambios y mejoras. Esto expande la lista de los requerimientos de sistemas conocidos.
4. Revisar el prototipo con base en la información obtenida a través de la experiencia del usuario.
5. Repetir los pasos anteriores las veces que sea necesario hasta obtener un sistema satisfactorio para el usuario.

Ventajas:

* El cliente es escuchado.
* Permite lograr un producto intermedio para conocer como responderán las funcionalidades previstas para el producto final.
* Se prueba antes de su entrega, lo que reduce costos y aumenta las probabilidades de éxito.
* Es el único apto para desarrollos en los que no se conoce a priori sus especificaciones o las tecnologías a utilizar.

Desventajas:

* Ansiedad del usuario al ver los prototipos ya que puede pensar que el sistema ya está terminado.
* Exige disponer de las herramientas adecuadas.
* No presenta calidad ni robustez.
* Es altamente costosa y difícil la administración temporal.

**Metodologías Ágiles**

Son una serie de técnicas para la gestión de proyectos que han surgido como contraposición a los métodos clásicos.

Todas las metodologías ágiles tienen 4 principios:

* *Individuos e interacciones* sobre procesos y herramientas.
* *SW funcionando* sobre documentación extensiva.
* *Colaboración con el cliente*sobre negociación contractual.
* *Respuesta ante el cambio* sobre seguir un plan.

Características

* Se apoyan en períodos de producción cortos y fijos (SPRINT) y mucha colaboración dentro del equipo del proyecto.
* El método tiene inherente en su ADN el cambio.
* El equipo cuenta con poder de decisión.
* Validación con el usuario. Tiene participación activa.
* Desarrollos en forma iterativa.
* Se aplica la regla 80/20 (hago el 80% de lo que me piden priorizando).
* Se completa cada función antes de continuar.
* Los requerimientos pueden evolucionar y eso no es un problema significativo.
* Son muy buenos para proyectos de mantenimiento y problemáticos para proyectos grandes.

Los 4 métodos ágiles más utilizados son:

**KANBAN**

Se basa en un sistema de producción que dispara trabajo sólo cuando existe capacidad para procesarlo. El disparador de trabajo es representado por tarjetas Kanban de las cuales se dispone de una cantidad limitada.

El elemento central es el tablero Kanban que registra el avance del proyecto y mide el flujo de tareas. Permite alcanzar un ritmo de producción sostenible e introducir cambios a los procesos con muy baja resistencia. Se emplea en organizaciones con muy baja madurez institucional.

El primer paso para implementar Kanban es identificar el flujo de trabajo. Es necesario limitar el número de tareas en cada columna. El equipo elige, adapta y adopta sus procesos según la necesidad.

Ventajas:

* Produce con calidad y a tiempo, lo que genera confianza en los clientes.
* Hace que el equipo mejore constantemente sus procesos y evita demoras.
* Hay satisfacción del personal cuando se completa una tarea y se mueve de una columna a la otra.

**SCRUM**

Está especialmente indicada para proyectos con un rápido cambio de requisitos.

El desarrollo de SW se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente.

La segunda característica importante son las reuniones a lo largo del proyecto. Éstas son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

**XP (Extreme Programing)**

Centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de SW, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

Se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.

Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

**FDD (Feature Driven Development)**

Define un proceso iterativo que consta de 5 pasos. Las iteraciones son cortas (hasta 2 semanas).

Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software.

**U3: ANÁLISIS Y DETERMINACIÓN DE REQUERIMIENTOS**

**Determinación de requerimientos**

Los requerimientos son las características que deben incluirse en un nuevo sistema. El análisis para determinarlos es llamado determinación de requerimientos, el cual consiste en el estudio del sistema para conocer cómo trabaja, evaluarlo y efectuar mejoras en caso de ser necesario. La determinación de requerimientos vincula el estudio de un sistema existente con la recopilación de detalles relacionados con él.

La persona encargada de llevar a cabo el análisis es el analista de sistemas, y su primer paso es comprender la situación actual y recopilar información. Ciertos tipos de requerimientos son tan fundamentales que son comunes en casi todas las situaciones.

Se divide en tres grandes actividades:

1. Anticipación de requerimientos: Consiste en prever las características del sistema con base en la experiencia previa en un área en particular y en el contacto con sistemas similares. Esto puede llevar al analista a investigar áreas y aspectos que de otra forma no serían tomados en cuenta y anticipar requerimientos.
2. Investigación de requerimientos: Es la actividad más importante del análisis. Los analistas estudian y documentan las características el sistema actual, utilizando para ello técnicas para encontrar datos, análisis de flujo de datos y análisis de decisión, e incluyen métodos para documentar y describir las características del sistema.
3. Especificación de requerimientos: Es el análisis de los datos recopilados que describen el sistema para determinar qué tan bueno es su desempeño, qué requerimientos se deben satisfacer y las estrategias para alcanzarlos.

Los analistas estructuran su investigación en base a 4 preguntas fundamentales:

• ¿Cuál es el proceso básico de la organización?

• ¿Qué datos utiliza o produce este proceso?

• ¿Cuáles son los límites impuestos por el tiempo y la carga de trabajo?

• ¿Qué controles de desempeño utiliza?

Para realizar estas preguntas se debe contemplar aspectos como: la finalidad del sistema, los pasos a seguir, lugar y quienes realizan las actividades, tiempos y frecuencias estimados de utilización del programa y quiénes utilizan la información resultante. Luego de reunir toda la información, se investiga en detalle los datos.

Es indispensable que todos estos procesos satisfagan todas las necesidades de quienes utilizarán el programa. Hay que tener en cuenta la información a utilizar, los datos requeridos, de qué manera se procesan, cómo se debe presentar la información obtenida y de qué manera afecta a las demás áreas de la organización.

**Técnicas de relevamiento**

Todas las técnicas tienen un mismo objetivo, relevar información relacionad con los requerimientos del sistema. Esa información es sobre el funcionamiento del sistema actual, problemas que surgen dentro del proyecto, nuevos requerimientos del usuario, expectativas o sentimientos del usuario sobre el proyecto. En general, los analistas emplean más de una de estas técnicas para estar seguros de llevar a cabo una investigación amplia y exacta.

1. **Entrevistas**: Son reuniones de dos o más personas en las cuales se mantiene una conversación para satisfacer las necesidades de información. Los entrevistados son usuarios de los sistemas existentes o que utilizarán el nuevo, y gerentes o empleados que proporcionan datos para el sistema o que serán afectados por él. El analista debe conocer el organigrama actualizado de la organización.

Este método es la mejor fuente de información cualitativa: opiniones, políticas, descripciones subjetivas de actividades y problemas. El analista podrá descubrir áreas mal comprendidas, expectativas poco realistas e incluso indicadores de resistencia hacia el sistema propuesto.

Clasificación:

* *No estructuradas*: Utilizan un formato pregunta-respuesta y son apropiadas cuando el analista desea adquirir información general acerca de un sistema. Este formato anima a los entrevistados a compartir sus sentimientos, ideas y creencias.
* *Estructuradas*: Utilizan preguntas estándar en un formato de respuesta abierta (permite que el entrevistado responda con sus propias palabras) o cerrada (utiliza un conjunto anticipado de respuestas).

Planificación y control de la entrevista: Para garantizar el éxito de la entrevista, la misma debe estar planificada y controlada. Se debe leer la documentación existente relacionada para saber el lenguaje que se emplea, coordinar entre los entrevistadores las preguntas para no realizar las mismas, determinar el tiempo máximo de duración de la entrevista, el momento y el lugar, y establecer el objetivo de la entrevista y comunicarlo.

1. **Cuestionario**: Permite a los analistas reunir información relacionada con varios aspectos de un sistema proveniente de un grupo grande de personas. Es un documento redactado en forma de interrogatorio, integrado con preguntas estrictamente relacionadas con el tema que se investiga. Debe ser acompañado de instrucciones claras y precisas.

Se usa cuando los interrogados se encuentran en distintas localidades, cuando los datos requeridos son cuantitativos, para sondear una muestra de usuario con el fin de detectar problemas o tener en cuenta aspectos importantes antes de la preparación de la entrevista, etc.

El empleo de formatos estandarizados puede proporcionar datos más confiables que otras técnicas y además, su amplia distribución asegura el anonimato de los encuestados, lo que puede llevar a respuestas más honestas, pero las mismas pueden ser limitadas.

La mayor diferencia con una entrevista es que no se puede cambiar el rumbo según el comportamiento del entrevistado. Las preguntas del cuestionario deben ser transparentes y se deben anticipar a las respuestas, deben ser concisas y cortas, deben agruparse las preguntas del mismo tema y mantener un estilo consistente.

1. **Revisión de registro:** Los registros y reportes pueden proporcionar al analista información valiosa con respecto a las organizaciones y a sus operaciones. Al revisar los registros, el analista examina la información relacionada con el sistema y los usuarios. La revisión de los registros puede efectuarse al comienzo del estudio o después, y sirve de base para comparar las operaciones actuales. Por lo tanto, los registros pueden indicar qué está sucediendo.

Los registros incluyen manuales de políticas, reglamentos y procedimientos estándares de operación utilizados por la mayor parte de las organizaciones como guías para los gerentes y empleados. Estos registros no indican la forma en la que se desarrollan las actividades en la realidad o dónde se encuentra todo el poder de la toma de decisiones. Sin embargo, pueden ser de gran ayuda para comprender el sistema al familiarizarlo con aquellas operaciones que necesitan apoyo y con las relaciones formales dentro de la organización.

1. **Observación:** Permite obtener información, que no se puede mediante otras técnicas, sobre la forma en que se efectúan las actividades. Es útil cuando el analista necesita observar la forma en que se manejan los documentos y se llevan a cabo los procesos, y si se siguen todos los pasos especificados.

El inconveniente que podría aparecer es la reacción de las personas frente a un observador y la imposibilidad de observar por periodos de tiempo muy largos para ver todos los procesos especiales.

1. **Muestreo:** Es la recopilación y análisis de datos seleccionados de una población de elementos de estudio. Se utiliza cuando la cantidad de datos a obtener es muy grande y cuando se requiere más tiempo del disponible para un estudio detallado.

La ventaja es la reducción de costos de recolección de datos y la agilidad. Se debe seleccionar adecuadamente el grupo muestral y la forma de obtener información deberá ser más precisa.

1. **Medición de trabajo administrativo:** Es un medio para determinar en qué medida una jornada normal de trabajo se aproxima a lo ideal. Permite acelerar los tiempos y establecer cuántos empleados son necesarios para realizar una determinada tarea. Las tareas de oficina que se pueden medir son sólo las de carácter repetitivo y cuantificable.

Esta técnica consta de tres pasos:

* *Indagar* sobre la rutina normal del trabajo y registrar los diferentes pasos en una decisión ordenada.
* *Determinar y registrar* el tiempo empleado en la ejecución de cada fase.
* *Calcular y registrar* el número de veces que se realiza una fase del trabajo.

**Documentación**

Documentar es una tarea necesaria e importante que debe realizarse en todas las etapas. Es un conjunto de información que indica qué hacen los sistemas, cómo lo hacen y para quién. Explica las características técnicas y la operación del sistema.

Existen varios tipos de documentación: La documentación de programas explica su lógica e incluye descripciones, diagramas de flujo, listados de programas y otros documentos. La documentación del usuario explica la naturaleza y capacidades del sistema y cómo usarlo.

Las herramientas para documentar son: árboles de decisión, tablas de decisión y español estructurado.

**Determinación de requerimientos funcionales y no funcionales**

Los requerimientos o requisitos de un sistema describen los servicios que ha de ofrecer el sistema y las restricciones asociadas a su funcionamiento que deben satisfacerse. La distinción entre requerimientos funcionales y no funcionales no siempre resulta evidente.

**Requerimientos funcionales**

Expresan la naturaleza del funcionamiento del sistema. Son declaraciones de los servicios que proveerá el sistema, cómo se debe comportar en situaciones particulares, cómo interactuar con su entorno, cuáles van a ser su estado y funcionamiento. En algunos casos, también declaran explícitamente lo que el sistema no debe hacer.

En principio, la especificación de requerimientos funcionales de un sistema debe estar completa y ser consistente, es decir, que todos los servicios solicitados por el usuario estén definidos y que los requerimientos no tengan definiciones contradictorias.

Los requerimientos funcionales deben ser redactados de forma comprensible para usuarios sin conocimientos técnicos avanzados, se debe especificar el comportamiento externo del sistema y evitar establecer características de su diseño y debe establecerse prioridad en los requisitos.

**Requerimientos no funcionales**

Definen cómo debe ser el sistema. Son aquellos requerimientos que no se refieren directamente a las funciones específicas, sino a las propiedades como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento. Definen las restricciones como la capacidad de los dispositivos de entrada/salida y la representación de datos que se utiliza en la interface del sistema.

Estos requerimientos surgen de la necesidad del usuario, debido a las restricciones en el presupuesto, a las políticas de la organización, a la necesidad de interoperabilidad con otros sistemas de SW o HW o a factores externos como los reglamentos de seguridad y las políticas de privacidad.

**Análisis de factibilidad**

Es una de las primeras etapas del desarrollo de un SI, el cual incluirá los objetivos, alcances y restricciones sobre un sistema, además de un modelo lógico de alto nivel del sistema actual. Se refiere a la disponibilidad de los recursos necesarios para llevar a cabo los objetivos o metas señalados sobre un proyecto. El estudio de la factibilidad lo lleva a cabo un pequeño equipo de personas capacitadas con técnicas de SI, que comprenden la parte de la organización que participará del análisis y diseño de sistemas.

Existen tres aspectos relacionados con el estudio de la factibilidad:

1. Factibilidad técnica: Evaluación para demostrar la facultad del sistema para ponerse en marcha y mantenerse, según las restricciones, objetivos y los recursos disponibles. Hay que considerar si el proyecto puede realizarse con el equipo actual, la tecnología existente de SW y el personal disponible, y si se necesita nueva tecnología, cuál es la posibilidad de desarrollarla.
2. Factibilidad legal: El sistema no debe infringir ningún tipo de norma o ley, ya sea a nivel local o nacional.
3. Factibilidad económica: Comprobar si el proyecto es sustentable económicamente, si los beneficios que se obtienen serán suficientes para aceptar los costos, y si los costos asociados con la decisión de no crear el sistema son tan grandes que se debe aceptar el proyecto.
4. Factibilidad operacional: Comprobar que la organización será capaz de darle uso al sistema, que cuenta con el personal capacitado para hacerlo, si tiene los recursos necesarios para mantener el sistema y si existirá cierta resistencia al cambio por parte de los usuarios.

**Elaboración de la propuesta de sistemas**

Además de identificar los objetivos, el alcance y las responsabilidades del proyecto, el analista también necesita preparar una propuesta de sistemas en la que se incluya la mayor cantidad de detalles sobre las necesidades, opciones y recomendaciones del sistema.

Hay diez secciones principales que conforman una propuesta de sistemas por escrito:

1. *Carta de presentación*: Debe listar a las personas que hicieron el estudio e incluir un resumen de los objetivos del mismo. Debe ser concisa y amigable.
2. *Portada del proyecto*: Debe incluir el nombre del proyecto, los nombres de los miembros del equipo de análisis de sistemas y la fecha en que se envía la propuesta.
3. *Índice de contenido*
4. *Resumen ejecutivo*: Se informa el quién, qué, cuándo, dónde, por qué y cómo de la propuesta. Debe incluir las recomendaciones del analista de sistemas y la acción deseada por parte de la administración.
5. *Esquema del estudio de sistemas con la documentación apropiada*: Información sobre los métodos o técnicas que se utilizaron en el estudio, y quién o qué fue lo que se estudió.
6. *Resultados detallados del estudio de sistemas*: Lo que descubrió el analista en relación con las necesidades humanas y de sistemas por medio de todos los métodos descritos. Se debe incluir las conclusiones sobre los problemas que experimentan los trabajadores al interactuar con tecnologías y sistemas.
7. *Alternativas de sistemas*: El analista debe presentar soluciones alternativas para los problemas mencionados. Una debe ser ‘dejar el sistema como está’. Hay que describir los costos, ventajas y desventajas de cada situación.
8. *Recomendaciones de los analistas de sistemas*: El analista presenta la solución recomendada. Hay que incluir las razones que respaldan la recomendación del equipo.
9. *Resumen de la propuesta*: Se informan los objetivos del estudio y la solución recomendada.
10. *Apéndices* (documentación variada, resumen de fases, correspondencia, etc): Puede incluir cualquier información que el analista considere de interés, pero que no sea esencial para comprender el sistema.

Una vez que termine de escribir la propuesta, debe seleccionarse cuidadosamente quiénes van a recibir el informe y entregarlo personalmente. La visibilidad es importante para la aceptación y el éxito en última instancia del sistema.

Por otro lado, es importante utilizar tablas y gráficos, además del texto, para presentar los aspectos básicos del sistema propuesto. Nunca hay que menospreciar un buen diseño. La integración de figuras en la propuesta ayuda a demostrar que se está consciente de las distintas formas en que las personas absorben la información. Las figuras en el informe complementan la información escrita y siempre se deben interpretar con palabras; nunca deben ir solas.

**ETAPAS QUE COMPRENDEN TODOS LOS CICLOS DE VIDA**

**1-Relevamiento**

**2-Análisis**

**3-DISEÑO**

En esta etapa se define cómo se implementa lo definido en el análisis.

Los objetivos para diseñar una buena interfaz son:

* Satisfacer las necesidades de los usuarios.
* Aumentar la velocidad de la captura de datos.
* Proporcionar la retroalimentación adecuada.
* Productividad.

Hay diferentes tipos de interfaces: de lenguaje natural, de preguntas y respuestas, menús, de llenado de formularios, interfaz gráfica de usuario.

Debe haber una comunicación significativa entre el diseño y la persona. Se debe estandarizar el color y el uso de íconos en todas las pantallas. Hay que validar que todos los datos sean correctos antes de guardar y tratar de que el trabajo del usuario sea mínimo.

**Diseño de salida**

La salida es la información que se entrega a los usuarios por medio del SI.

Objetivos:

* Diseñar la salida para satisfacer un propósito específico.
* Hacer significativa la salida para el usuario.
* Entregar la cantidad adecuada de salidas.
* Proporcionar una distribución adecuada de salida.
* Proporcionar la salida a tiempo.
* Elegir el método de salida más efectivo.

Los tipos de salidas son *externas* (la información sale de la organización para comunicarse con los clientes) o *internas* (la información permanece dentro de la organización).

Se debe seleccionar la *tecnología de salida* adecuada según los usuarios, el uso, el tiempo de almacenamiento y acceso, etc. Los tipos de tecnologías pueden ser: monitores, impresoras, animación, CD, DVD, salida electrónica.

**Diseño de entradas**

Las entradas deben ser efectivas, consistentes, simples y atractivas. Se debe tener en cuenta el diseño de pantallas o formularios para ingresar datos utilizados actualmente para intentar mantenerlo en el nuevo sistema ya que los usuarios están acostumbrados a tal ordenamiento.

**4-CODIFICACIÓN Y PUEBA**

En la mayoría de los casos codificacióny pruebavan de la mano.

**Fase de programación**

La codificación es el traspaso de las especificaciones de diseño en un lenguaje de programación determinado por el analista al inicio del proyecto, y debe ser acorde al objetivo deseado. Se debe tener en cuenta la cantidad de programadores requeridos para el proyecto. Esta etapa es la primera de evolución del programa.

Siempre desarrollar pensado en la reutilización.

**Fase de documentación**

Documentar es una tarea tan necesaria e importante como escribir el código. El código indica cómo funciona el sistema y la documentación indica porque lo hace.

Tipo de documentación:

* *Documentación interna*: Comentarios que se encuentran dentro del programa fuente y describen detalles importantes para el programador.
* *Documentación externa*: Es independiente al programa fuente. Suele hallarse en un manual impreso que se proporciona junto con el programa. Describe las características técnicas y de funcionamiento del código.
* *Manual del usuario*: Indica al usuario cómo utilizar y operar correctamente el sistema. Apunta netamente a lo funcional (no menciona a aspectos del código). Se hace después de terminado el programa.

**Fase de prueba**

La prueba del SW consiste en determinar cómo operan los componentes del sistema en situaciones representativas y ver si su comportamiento es el esperado, es decir, permite verificar la calidad del producto. Se prueba la presencia de errores pero no la ausencia de los mismos. Demuestra eficacia cuando detectan problemas en el momento oportuno. Puede darse antes o después de finalizado el sistema. Requiere un proceso ordenado para llevar adelante las pruebas.

Se basa en dos controles generales:

* *Verificación*:consiste en comprobar si el sistema y sus componentes cumplen con las especificaciones del problema. ¿Estamos resolviendo el problema correcto?
* *Validación*: consiste en evaluar el sistema y sus componentes, durante o al final del proceso de desarrollo y determinar si satisface los requisitos de las especificaciones. ¿Estamos resolviendo el problema de manera correcta?

**Consideraciones:**

* Es muy difícil probar todas las condiciones de un programa. No se puede demostrar que está libre de defectos.
* La prueba es buena o muy buena cuando hay una alta posibilidad de encontrar un error.
* Además de las pruebas del programador, otra persona debe realizarlas para encontrar errores que uno no ve.
* La prueba debe ser repetible. Probar con los mismos datos varias veces y con varios ejemplos diferentes.
* La prueba debe servir para localizar errores y poder corregirlos, no solamente detectar su presencia.

**Plan de pruebas:**

Está constituido por un conjunto de pruebas. Cada una debe dejar en claro qué tipo de propiedad se quiere probar, cómo se va a medir el resultado, especificar en qué consiste dicha prueba y los requerimientos de HW y SW que se requieren, y definir cuáles son los resultados que se esperan.

Cada prueba puede tener muchos de casos de prueba. Un caso de prueba consta de un propósito, los pasos de ejecución y el resultado que se espera.

Tipos de pruebas:

* **Módulos individuales:** Cada módulo se prueba en forma independiente.
  + *Caja Blanca o Estructural*: Usa la estructura interna del sistema para derivar los casos de prueba. Apunta a verificar el código.
  + *Caja Negra o Funcional*: Los casos de prueba se derivan de las especificaciones. Relacionado al usuario final. Prueba que los resultados sean correctos, sin importar como están codificados.
* **Integración:** Son sobre el sistema, a mayor escala. Se busca chequear la coherencia entre los distintos módulos en conjunto. Se hace en forma progresiva.
  + *Forma ascendente*: Se empieza con los módulos individuales y luego se integran los módulos hasta probar todo.
  + *Forma descendente*: Empieza con el sistema total integrado y luego se evalúan los módulos individuales.
* **Aceptación:** La realiza el usuario final y es funcional. Se hace sobre el sistema completo. Es quien le da el OK al programa.

**5-CONVERSIÓN**

Es la etapa de conversión de datos del sistema anterior al formato del nuevo sistema. Puede estar o no en el desarrollo de un sistema. Habrá que desarrollar programas denominados programas de interface, los cuales se van a utilizar solamente en esta etapa.

**6-CAPACITACIÓN**

Es la tarea final del equipo de desarrollo. Para llevarse a cabo debe haber un plan determinado con días, horarios y grupos de usuarios. Conviene empezar la capacitación antes de que el sistema empiece a operar. En esta etapa es cuando se entrega el manual de usuario, que es donde se detalla la parte funcional del sistema. Es conveniente incluir una fase de práctica. No conviene que el usuario pierda continuidad, no debe ser mucho antes de la implementación.

Es una tarea del analista, pero ciertos usuarios que están preparados y que tuvieron contacto con la etapa de desarrollo pueden hacerse cargo de esta tarea. Hay que tener en cuenta a los usuarios para la elección del tipo de método de capacitación.

**Métodos de capacitación**

* **Seminarios o grupos:** La ventaja es que permite llegar a mucha gente a la vez.
* **Capacitación de procedimientos:** Le proporciona al individuo los procedimientos escritos que detallan las actividades.
* **Capacitación tutorial:** Capacitación personalizada que cada usuario puede realizar en forma independiente.
* **Simulación:** La capacitación se hace primero simulando el ambiente de un caso real. La ventaja es que permite que el usuario logre un nivel de manejo aceptable del sistema.

**7-IMPLEMENTACIÓN**

**Fase de instalación del sistema**

Consiste en instalar y poner en funcionamiento el sistema desarrollado.

* Preparación del equipamiento necesario donde se instalará el sistema, puede requerir de proveedores.
* Preparar el área de trabajo del usuario.
* Instalación del SW. Los datos deben estar ingresados. Agregar otros datos necesarios.

**Fase de implementación del sistema**

Existen 4 enfoques para poner en funcionamiento el sistema:

1. **Conversión en forma directa:** se reemplaza el sistema existente por uno nuevo.

* Es barato.
* Existe un riesgo importante de fracaso, si no se hicieron buenas pruebas del sistema.

1. **Implementación en paralelo:** el sistema viejo y el sistema nuevo funcionan en paralelo por un tiempo. Sirve para comparar. A este enfoque también se lo cataloga como una fase de prueba. Es importante pactar una fecha para comenzar a utilizar sólo el sistema nuevo.

* Existe menos riesgo al fracaso.
* Mantenimiento de datos, costos por duplicidad de tareas.

1. **Enfoque modular:** Se implementa todo el sistema completo en un lugar (una sucursal). Se usa en sistemas grandes donde se particiona la empresa.

* El riesgo de fracaso está localizado. Los problemas pueden corregirse antes de implementar en otros lugares. La capacitación se puede facilitar al usar gente de un lugar donde el sistema ya está en uso.
* El periodo de implementación es más largo. No es aplicable a cualquier empresa.

1. **Enfoque en fases:** Se usa en situaciones de sistemas grandes. En este caso se particiona el sistema. Se implementa un fragmento del sistema en todos los lugares.

* Debería minimizarse el tiempo de implementación. Asegurarse que una porción del sistema anda en todas las sucursales. En caso de surgir errores en una sucursal, tomar como ejemplo las otras sucursales.
* Alto costo en programas de interfaces para el diálogo de datos entre el sistema viejo y el nuevo.

**8-REVISIÓN DEL SISTEMA**

Está conducida por los analistas y los usuarios. La idea es determinar que tan bien está funcionando el sistema y ver cómo ha sido aceptado por parte de los usuarios y qué ajustes hay que hacer. Es una tarea importante para recabar información en un futuro mantenimiento del sistema.

Para llevar a cabo esta tarea de revisión, hay dos tipos de métodos:

1. **Métodos Comunes:** Similares a los métodos de relevamientos de datos. Cuestionarios, entrevistas, observaciones directas, etc.
2. **Métodos Complementarios:**

* *Registro de eventos:* donde el usuario registra eventos críticos del sistema.
* *Evaluación del impacto:* para determinar cómo afecta el nuevo sistema al resto de la organización.
* *Encuestas de actitud:* cómo se siente el usuario con el nuevo sistema (revelar el lado malo del sistema).

**9-MANTENIMIENTO**

Consiste en hacerle mejoras y/o correcciones al sistema. Puede llegar a extenderse durante toda su vida útil. Hay que actualizar la documentación.

**Tipos de mantenimiento**:

* Correctivo: adaptaciones al sistema de emergencia ante casos de errores que surjan.
* Adaptativo: son modificaciones que hay que hacerle al sistema por alteraciones que se dan en el entorno.
* Perfectivo: son mejoras solicitadas por el usuario.

**Técnicas facilitar el mantenimiento:**

Para reducir la necesidad de mantenimiento o bien para que el mantenimiento se realice más eficientemente se debe:

* Definir con mayor precisión los requerimientos del usuario durante el desarrollo del sistema.
* Preparar lo mejor posible la documentación.
* En el momento de la programación, usar métodos más efectivos para el diseño de la lógica del procedimiento.
* Hacer un mejor uso de las herramientas existentes en el momento de hacer el sistema.
* Dirigir el proceso de desarrollo del sistema en forma efectiva siguiendo un plan.

**Mantenimiento estructurado**

Se da cuando el sistema es estructurado, es decir, fue diseñado en forma modular. Cada módulo tiene un mínimo efecto sobre los demás módulos, mediante pasaje de parámetros. Cuando hay que hacer mantenimiento de estos sistemas no es necesario modificar todo el programa, sino que se trabaja directamente con el modulo que corresponde, y el impacto con los otros módulos es mínimo.

**U8: MODELO DE DATOS**

**Modelo:** Representación abstracta de los datos recolectados para llevar adelante el sistema.

1. **Modelo esencial**: Indica lo que el sistema debe hacer para satisfacer los requerimientos del usuario.
   * Modelo ambiental: Es el que determina los límites del sistema. Se presenta el sistema y los elementos del entorno con los que interactúa. No se muestra nada del proceso. La herramienta con la que se trabaja es el diagrama de procesos. Hay que determinar bien el objetivo del sistema.
   * Modelo de datos: Es una representación gráfica de los datos que el sistema debe recordar para dar respuesta a los pedidos del usuario.
     + Modelo conceptual de datos: Análisis el sistema. Herramientas: diagrama de entidad relación y diccionario de datos.
     + Modelo lógico de datos: Es la etapa de diseño del modelo de datos de forma consistente, ayuda a reducir los datos. La herramienta de trabajo es la normalización.
     + Modelo físico de datos: Se implementa el modelo. Intervendría la programación.
   * Modelo de comportamiento: Representa gráficamente las transformaciones del sistema sobre las entradas para arrojar una salida. Herramientas: diagrama de flujo de datos, diagrama de transición de estado y especificaciones de procesos.
2. **Modelo de implementación**: Aparece la tecnología para mostrar las funcionalidades del sistema.
   * Modelo de usuario: Muestra aspectos relevantes del sistema desde el punto de vista del usuario (interfaces).
   * Modelo de sistema: Destaca como se incorpora la tecnología desde el punto de vista del procesamiento de datos.
     + Modelo de procesadores: Muestra los procesadores que van a intervenir en el sistema.
     + Modelo de tareas: Muestra las tareas que se van a realizar y los procesadores que las llevarán a cabo.
   * Modelo de programas: Muestra la organización del código organizado en módulos. Herramienta: Carta de estructura.

**Modelo de datos**

Se representan las entidades y relaciones que existen entre ellas y que permiten cumplir con el objetivo del sistema.

Es un sistema formal y abstracto que permite describir los datos de acuerdo con reglas y convenios predefinidos. Es formal, puesto que los objetos del sistema se manipulan siguiendo reglas perfectamente definidas y utilizando exclusivamente los operadores definidos en el sistema, independientemente de lo que estos objetos y operadores puedan significar.

Es una combinación de tres componentes:

1. Colección de estructura de datos.
2. Colección de operadores o reglas de inferencia.
3. Colección de reglas generales de integridad, los cuales definen un conjunto de datos consistente.

Validación del modelo de datos: Surgirán en el relevamiento.

1. *Visiones de contexto:* son conocimientos que describen políticas propias de una organización, sus recursos y su modo de operación.
2. *Visiones de usuario:* son los distintos requerimientos de información planteados por el usuario.

**Elementos para armar el modelo de datos:**

* Entidades**:** representan conceptos de interés que una organización considera relevantes. Es algo de lo cual interesa almacenar información. Pueden ser tangibles o intangibles. Tienen que tener un nombre representativo. Dentro de la entidad se almacenan sus *atributos*.
* Atributos**:** Son los datos de intereses de la entidad y son atómicos. Pueden clasificarse:
  + *Atributos claves:* es el atributo que identifica de manera única a un registro. Cuando una entidad tiene muchas claves, se pueden crear claves ficticias. No pueden asumir valores nulos.
  + *Atributos complementarios:* describen la entidad.

**Claves candidatas**: un atributo de una entidad es clave candidata si cumple con dos características:

1. Unicidad**:** en cualquier momento no existen dos tuplas o registros con el mismo valor.
2. Minimalidad: si una clave es compuesta, al eliminar cualquiera de sus componentes perdemos la condición de unicidad.

Clave externa o foránea: es una combinación de atributos de una entidad que son a su vez una clave primaria para otra entidad.

**Vinculaciones** (relaciones entre las entidades):

* Dependencia funcional: dada una entidad R, el atributo B de R depende funcionalmente del atributo A de R sí y sólo sí cada valor de B está asociado a cada valor de A. Todos los atributos no claves dependen del atributo clave.

Las dependencias funcionales pueden ser de dos tipos:

* + *Completa***:** Se dice que el atributo Y de la relación R es por completo dependiente funcionalmente del atributo X de la relación R si depende funcionalmente de X y no depende funcionalmente de ningún subconjunto propio de X (es decir, no existe un subconjunto propio Z de los atributos componentes de X tales que Y sea funcionalmente dependiente de Z). El atributo nombre de libro depende de forma funcionalmente completa de código de libro.
  + *Incompleta***:** fecha de edición y editorial dependen de forma funcionalmente no completa de código de libro y número de edición.
* Inter-entidades: son las relaciones que existen entre varias entidades del sistema, hay de tres tipos:
  + *1 a 1***:** No se representa en el modelo de datos. Cuando a cada elemento de la clave de A se le asigna un único elemento de la entidad B y para cada elemento de la clave de la entidad B se le asigna un único elemento de la entidad A. No pueden aparecer dos entidades con las mismas claves. La solución es unificar las entidades.
  + *1 a muchos***:** cuando la clave de la entidad A posee varios elementos relacionados a la entidad B y cuando una clave de la entidad B posee un único elemento relacionado a la entidad A. Son las únicas que representamos en el modelo de datos.
  + *Muchos a muchos***:** Cuando una clave de la entidad A posee varios elementos relacionados de la entidad B y cuando una clave de la entidad B posee varios elementos relacionados de la entidad A. La solución es la creación de una entidad con la combinación de las claves de ambas entidades.

**Normalización**

Es una técnica que permite encontrar una estructura de datos que resulte adecuada. Mediante la aplicación sistemática de una sucesión de pasos formales va transformando las entidades originales en otras más adecuadas en función del objetivo fijado.

* Tiene una estructura mínima que evita la redundancia de datos.
* Asegura la no pérdida de información y que se cubran todas las visiones de contexto y de usuario.
* Facilita los procesos de alta, baja y modificación.
* Máxima pretensión de supervivencia (hace el modelo adaptable a cambios).

Entidades normalizadas:

1. No hay tuplas o registros duplicados, ósea que hay unicidad de contenido.
2. No hay orden entre las tuplas o registros.
3. No hay orden en los atributos.
4. Todos los valores de los atributos son atómicos.
5. Homogeneidad estructural (los registros de una tabla deben representar lo mismo).

**Proceso de normalización**: Consta de varias fases que se denominan formas normales.

Primera forma normal: Una entidad R está en primera forma normal si y solo si todos los dominios sólo contienen valores atómicos. Consta de armar el grafo de dependencia funcional.

*Problemas:*

* *Alta*: no podría guardar un registro hasta que poder completar todos los campos de la tabla.
* *Baja*: al borrar un registro podría eliminar los datos de algún atributo del registro que podría necesitar en el futuro.
* *Modificación*: sería tedioso al aumentar la cantidad de registros.

Segunda forma normal: Una entidad R está en segunda forma normal si y solo si está en primera forma normal y cada atributo no clave depende de la clave primaria en forma funcionalmente completa. Los problemas de la primera forma normal están solucionados. Pero surgen otros problemas, similares a los de la primera forma normal pero trasladados a una entidad:

* *Alta*: sigo sin poder guardar registros hasta completar la tabla (Ej: cargar ciudades sin tener proveedores en ella).
* *Baja*: al borrar un registro podría eliminar los datos de algún atributo del registro que podría necesitar en el futuro (Ej: borrar un proveedor y eliminar una ciudad sin desearlo).
* *Modificación*: sería tedioso al aumentar la cantidad de registros.

Tercera forma normal: Una entidad está en tercera forma normal si y solo si está en segunda forma normal y no existen dependencias funcionales entre atributos no claves. Las entidades quedan normalizadas sin problemas.

**Especificaciones de proceso (Miniespecificaciones)**

Debe describir cómo se logra transformar el flujo de datos que llega en los flujos que salen, así como las normas que gobiernan la transformación.

La especificación debe expresarse de una manera que pueda verificarse tanto de parte del usuario como del analista.

El proceso debe especificarse en una forma que pueda ser comunicada efectivamente al público amplio que está involucrado.

Son un lenguaje estructurado:

* Debe ser comprensible tanto para el usuario como para el analista.
* Utiliza una sintaxis limitada, sin llegar a ser tan rígida como la codificación.
* Las descripciones deben ser precisas y concisas.
* Utiliza verbos precisos, sin ambigüedad.
* Los nombres utilizados deben estar definidos en el diccionario de datos o ser términos locales.
* Aparecen los “Términos locales”: variables que se definen en la especificación y son solo relevantes en ella.
* La sintaxis son sentencias declarativas simples (asignaciones), construcciones de decisión (if), construcciones de repetición (while) y combinaciones de las últimas dos.

**Tablas de decisión**

Herramienta de documentación de procesos complejos. Importante en el análisis de decisiones estructuradas. Ventaja: ayudan al analista a asegurar la completitud. Es fácil verificar errores como situaciones imposibles, contradicciones y redundancias.

Se compone de cuatro cuadrantes:

|  |  |
| --- | --- |
| Condiciones que  se pueden dar | Alternativas de condición |
| Cada una de las acciones que se pueden realizar | Entradas de acción: acción que se va a realizar según las condiciones dadas |

Para construir la tabla se necesita determinar el tamaño máximo de la tabla, eliminar situaciones imposibles, eliminar inconsistencias o redundancias y simplificar las tablas tanto como sea posible.

Pasos:

1. Determinar el número de condiciones que podrían afectar a la decisión. El número de condiciones se vuelve el número de filas que tenemos en el cuadrante superior izquierdo.
2. Determinar el número de posibles acciones que se pueden realizar.
3. Determinar el número de alternativas de condición para cada condición.
4. Completar las alternativas de condición.
5. Completar la tabla insertando la “X” indicando la acción a realizar teniendo en cuenta alternativa de condición.
6. Reducir el número de situaciones que den como resultado la misma acción.
7. Verificar si la tabla presenta situaciones imposibles, contradicciones o redundancias.
8. Reorganizar las condiciones y acciones para hacer comprensible la tabla (redibujar).

**Árbol de decisión**

Se usan cuando ocurre una bifurcación compleja en un proceso de decisión estructurada.

Con un nodo ( ) se indica una acción. Con un círculo ( ) se indica una condición a evaluar.

No se busca una resolución óptima. Se construye de izquierda a derecha. Se debe seguir una numeración correlativa.

*Ventajas:*

* No necesariamente tiene que ser simétrico.
* Las acciones se pueden repetir.
* Las condiciones no importantes están ausentes.
* Son entendidos con más rapidez que las tablas de decisión por un usuario final.

Conviene usar **tablas de decisión** cuando hay combinaciones complejas de acciones, condiciones y reglas, y cuando se requiere un método que evite eficazmente situaciones imposibles, contradicciones y redundancias.

Conviene usar un **árbol de decisión** cuando la secuencia de condiciones y acciones sea crítica, y cuando no todas las condiciones sean relevantes para cada acción.

**Conversión a un nuevo Sistema**

Para la implementación hay que convertir físicamente el sistema de información viejo a uno nuevo o modificado.

**Estrategias de conversión**

* Conversión directa: en una fecha específica, el sistema viejo se abandona y el nuevo se pone en uso. Puede tener éxito sólo si antes se lleva a cabo una comprobación extensa, y funciona mejor cuando se pueden tolerar algunos retrasos en el procesamiento. Se considera un enfoque arriesgado para la conversión. Podrían ocurrir trastornos en el entorno laboral si los usuarios se ofenden porque se les obliga a usar un sistema desconocido sin recursos. No hay ninguna forma adecuada para comparar los nuevos resultados con los viejos.
* Conversión paralela: se ejecuta al mismo tiempo el sistema viejo y el nuevo, en paralelo. Cuando se obtienen los mismos resultados todo el tiempo, el nuevo sistema se pone en uso y el viejo se detiene. Una ventaja es la posibilidad de verificar los nuevos datos contra los viejos. Las principales desventajas incluyen el costo de ejecutar dos sistemas al mismo tiempo y el agobio en los empleados de virtualmente doblar su carga de trabajo durante la conversión.
* Conversión gradual o por fases: combina los anteriores, sin incurrir en todos los riesgos. El volumen de las transacciones manejado por el nuevo sistema aumenta gradualmente conforme el sistema se introduce por fases. Las ventajas incluyen permitir a usuarios que se involucren gradualmente con el sistema, la posibilidad de descubrir y recuperar errores sin desperdiciar mucho tiempo y la capacidad de agregar características una por una. Las metodologías ágiles tienden a usar esta metodología de conversión.
* Conversión modular: Usa la construcción de subsistemas independientes y operacionales para cambiar de los sistemas viejos a los nuevos de forma gradual. Conforme se modifica y acepta cada módulo, se pone en uso. Las ventajas son que cada módulo se prueba completamente antes de ser usado y que los usuarios se familiarizan con cada módulo conforme se vuelve operacional. Esta retroalimentación ha ayudado a determinar los atributos finales del sistema. Las metodologías orientadas a objetos utilizan con frecuencia esta metodología.
* Conversión distribuida: Se refiere a una situación en que se contemplan muchas instalaciones del mismo sistema. Una conversión completa se hace (con cualquiera de los cuatro enfoques) en un sitio. Cuando esta conversión se completa exitosamente, se prosigue con las demás. Una ventaja es que se pueden detectar y contener un problema antes de que repercuta simultáneamente en todos los sitios. Una desventaja es que incluso cuando una conversión es exitosa, cada sitio tendrá su propia gente y cultura, además de sus propias peculiaridades regionales y locales para trabajar, y se deben manejar en consecuencia.