

Universidad Nacional Autónoma de Honduras

Facultad de Ingeniería
Departamento de Ingeniería en Sistemas



Introducción a la industria del software

Aspectos Básicos de la Industria del SW

¿Qué es la industria del Software?

- La **industria del software** es la industria que involucra la investigación, desarrollo, comercialización y distribución de software.
- Es la piedra angular de toda innovación tecnológica ya que este impacta en forma directa o indirecta sobre todas las actividades económicas, debido a que estas posee una alta demanda en conjunto con los servicios de tecnología de información.

Pero, ¿Qué es el Software?

- Es conjunto de instrucciones de lenguajes de computadora, que cuando se ejecuta brinda una función y rendimiento deseado.
- Es el conjunto de estructuras de datos que permite la manipulación adecuada de los datos para transformarlos en información útil para los usuarios y que además sirve de agente multiplicador en la calidad y ganancias de las organizaciones o individuos que lo usan.

Desde la Perspectiva de la Industria del Software, este se clasifica en:

A) Por la forma de la entrega:

- ✓ Software “enlatado”: Un producto de Software enlatado responde a especificaciones de uso extendidos aplicables a una industria o actividad específica, a la vez que se le da un carácter universal.

Ejem: el MS-Office, Corel Draw o Adobe Acrobat

- ✓ **Software “Hecho a la medida”:** Se refiere a la creación o modificación de productos de software que responde a las especificaciones particulares de un cliente, por lo cual el producto final entregado a este, es de carácter único, en esta modalidad el código del software es de carácter cerrado, es decir que no puede ser objeto de modificación por parte del cliente.

- ✓ **Software de Código Abierto:** Este es el software al cual el usuario tiene acceso al código fuente de la aplicación, pudiendo así modificar el mismo, un ejemplo de esto es el sistema operativo Linux.

B) Por alcance Funcional:

- ✓ **Software Empresarial:** Por lo general este tipo de software es de carácter empaquetado o hecho a la medida, de uso personal tales como hojas de calculo, procesadores de texto, u orientados a resolver funciones de negocios de mayor complejidad tal es el caso de ERP o CRM.

- ✓ **Software del Sistema:** Este es el tipo de software que incluye a los sistemas operativos, herramientas de administración, backup, tuning, y recuperación de información, estos software sirven como puente entre el hardware y las aplicaciones de Software.
- ✓ **Software de Uso Personal:** Se define como el software de uso masivo en computadores personales, estos contiene programas de hojas de calculo, de planillas, procesadores de texto, navegadores de Internet, aplicaciones multimedia, administradores de agenda y correo electrónico.

- ✓ **Software de Diseño, animación y simulación de Sistemas en la Ingeniería y Arquitectura:** Este es por lo general en la mayoría de los casos utilizados por profesionales de diseño en las áreas como ser Sistemas (CAD y Herramientas CASE), Electrónica (EDA), Arquitectura y Construcción (AEC), Manufactura (CAM), e Ingeniería (CAE).

- ✓ **Software de Seguridad:** Estos brinda administración de seguridad al nivel del sistema, accesos de usuarios, seguridad en redes, todo lo referente a protección de virus y Spam, firewall, encriptación, detección y rechazo de intrusos y análisis de vulnerabilidades.

Aspectos o ámbitos de la Industria de Software

- **Normativas y certificaciones de Calidad:**
- ✓ Una de las dificultades al momento de evaluar la calidad de un software es que la mayoría de las características que definen al software no se pueden cuantificar fácilmente; lo que dificulta su medición.

Para solucionar el problema anterior se ha definido en la industria del software una serie de Normas entre las cuales encontramos a:

- o **Factores de calidad de McCall:** Estos establecen los factores que permite la calidad se divide en dos grupos que son los que se mide in directamente (cualitativos) y los que se miden directamente (cuantitativos).

- ✓ Mc Call establece que en ambos casos debe presentarse una medición y que estas se deben de comparar para obtener algún indicio sobre la calidad del producto de software.
- ✓ Este punto de vista se centra en tres aspectos importantes los cuales son sus características operativas, su capacidad para experimentar cambios y su capacidad para adaptarse a nuevos entornos

- Mc Call orienta su criterio de calidad basándose en métricas para aspectos tales como:
 - ✓ Corrección
 - ✓ Confiabilidad
 - ✓ Flexibilidad
 - ✓ Integridad
 - ✓ Facilidad de Reutilización
 - ✓ Interoperabilidad
 - ✓ Facilidad de Uso
 - ✓ Facilidad de Mantenimiento
 - ✓ Eficiencia
 - ✓ Portabilidad

- o **ISO 9126:** La ISO, bajo la norma ISO-9126, ha establecido un estándar internacional para la evaluación de la calidad de productos de software el cual fue publicado en 1992 .

El estándar ISO-9126 establece que cualquier componente de la calidad del software puede ser descrito en términos de una o más de seis características básicas, las cuales son:

- ✓ **Funcionalidad:** Se define como una serie de atributos que permiten calificar si un producto de software maneja en forma adecuada el conjunto de funciones que satisfagan las necesidades para las cuales fue diseñado, por lo general en este aspecto debe realizarse la siguiente pregunta, ¿Las funciones y propiedades satisfacen las necesidades explícitas e implícitas?

Para evaluar la funcionalidad se basa en los siguientes atributos:

- **Adecuación:** Se enfoca a evaluar si el software cuenta con un conjunto de funciones apropiadas para efectuar las tareas que fueron especificadas en su definición.
- **Exactitud:** Este atributo permite evaluar si el software presenta resultados o efectos acordes a las necesidades para las cuales fue creado.
- **Interoperabilidad:** Permite evaluar la habilidad del software de interactuar con otros sistemas previamente especificados.
- **Conformidad:** Evalúa si el software se adhiere a estándares, convenciones o regulaciones en leyes y prescripciones similares.

- ✓ **Confiabilidad:** Aquí se agrupan un conjunto de atributos que se refieren a la capacidad del software de mantener su nivel de ejecución bajo condiciones normales en un periodo de tiempo establecido, al momento de evaluar este parámetro debemos de preguntarnos, ¿Se puede mantener el nivel de rendimiento bajo ciertas condiciones y por cierto tiempo?

Las subcaracterísticas que el estándar sugiere son:

- **Tolerancia a fallas:** Se refiere a la habilidad de mantener un nivel específico de funcionamiento en caso de fallas del software o de cometer infracciones de su interfaz específica.
- **Recuperación.** Se refiere a la capacidad de restablecer el nivel de operación y recobrar los datos que hayan sido afectados directamente por una falla.

- ✓ **Usabilidad:** Consiste de un conjunto de atributos que permiten evaluar el esfuerzo necesario que deberá invertir el usuario para utilizar el sistema. Al momento de evaluar este parámetro debemos de preguntarnos, ¿El software es fácil de usar y de aprende a usarlo?

Los aspectos para evaluar la usabilidad son:

- **Comprensibilidad:** Se refiere al esfuerzo requerido por los usuarios para reconocer la estructura lógica del sistema y los conceptos relativos a la aplicación del software.
- **Facilidad de Aprender:** Establece atributos del software relativos al esfuerzo que los usuarios deben hacer para aprender a usar la aplicación.

- ✓ **Eficiencia:** Esta característica permite evaluar la relación entre el nivel de funcionamiento del software y la cantidad de recursos usados. Los aspectos a evaluar son:
- **Comportamiento con respecto al Tiempo:** Atributos del software relativos a los tiempos de respuesta y de procesamiento de los datos.
- **Comportamiento con respecto a Recursos:** Atributos del software relativos a la cantidad de recursos usados y la duración de su uso en la realización de sus funciones.

- ✓ **Mantenibilidad:** Se refiere a los atributos que permiten medir el esfuerzo necesario para realizar modificaciones al software, ya sea por la corrección de errores o por el incremento de funcionalidad. La pregunta ha hacerse es, ¿El software es fácil de modificar y verificar? En este caso, se tienen los siguientes factores :
- **Capacidad de análisis.** Relativo al esfuerzo necesario para diagnosticar las deficiencias o causas de fallas, o para identificar las partes que deberán ser modificadas.

- **Capacidad de modificación:** Mide el esfuerzo necesario para modificar aspectos del software, remover fallas o adaptar el software para que funcione en un ambiente diferente.
- **Facilidad de Prueba:** Se refiere al esfuerzo necesario para validar el software una vez que fue modificado.

- ✓ **Portabilidad:** En este caso, se refiere a la habilidad del software de ser transferido de un ambiente a otro, aquí la interrogante es, ¿Es fácil de transferir de un ambiente a otro? Para su evaluación se debe considerar los siguientes aspectos:
- **Adaptabilidad:** Evalúa la oportunidad para adaptar el software a diferentes ambientes sin necesidad de aplicarle modificaciones.
- **Facilidad de Instalación:** Es el esfuerzo necesario para instalar el software en un ambiente determinado.

- **Conformidad:** Permite evaluar si el software se adhiere a estándares o convenciones relativas a portabilidad.
- **Capacidad de reemplazo.** Se refiere a la oportunidad y el esfuerzo usado en sustituir el software por otro producto con funciones similares

- o **Modelo de Madurez de Capacidades (MMC):**
Desarrollado por el SEI (Software Engineer Software), este establece un marco evolutivo de maduración de proceso del software con 5 niveles de certificación partiendo de un proceso ad-hoc hasta un proceso maduro y disciplinado.

Nivel 0: Incompleto

- Este nivel quiere decir que los procesos aun no se realizan o que no alcanza todas las metas y objetivos definidos para pasar al siguiente nivel.

Nivel 1: Realizado

- Todas las metas específicas del área de proceso han sido satisfechas y las tareas necesarias para producir el producto han sido realizadas

Nivel 2: Administrado

- Todos los criterios del nivel1 han sido cumplidos además todo el trabajo asociado con el área de procesos se ajusta a una política organizacional definida y todas las tareas de trabajo y producto están siendo monitoreados, controlados, y revisados.

Nivel 3: Definido

- Todos los criterios del nivel 2 son cumplidos a cabalidad. Además el proceso esta adaptando al conjunto de procesos estándar de la organización de acuerdo a las políticas de organización de la misma

Nivel 4: Administrado de Forma cuantitativa

- Todos los criterios de nivel 3 han sido cumplidos. Además el área de procesos se controla y mejora mediante mediciones y evaluaciones cuantitativas para la calidad y desempeño.

Nivel 5: Mejorado

- Todos los criterios de nivel 4 han sido cumplidos. Además el área de procesos se adapta y mejora mediante el uso de medios cuantitativos es decir estadísticos, para conocer la necesidad cambiante del cliente y mejorar de manera continua la eficacia del área de proceso que se esta considerando.

- El ámbito de protección por derechos de propiedad intelectual del software varia, según las características del mismo y del tipo de legislación del país donde se pretende reinvidicar la protección legal. En principio las normativas legales que protegían al software eran los derechos de autor o patentes, pero en conjunto con estos otros aspectos entre los cuales se cuenta:

Derechos de Autor

El derecho de autor son todas las creaciones expresadas en forma literaria, musical, artística y/o científica, son objetos de protección por derecho de autor las siguientes obras: obras escritas ya sean literarias, científicas, didácticas, eslóganes publicitarios, almanaques, anuarios, folletos, etc. también obras musicales, teatrales, artísticas tales como dibujo, pintura, grabado, escultura, fotografía, arquitectura, obras audiovisuales tales como cinematografía, video, etc, obras folklóricas y programas informáticos.

Marca

Son signos con caracteres distintivos y su función principal es la de permitir de diferenciar los productos o servicios de una empresa de otra. Es así como actúa en simultáneo con otros derechos.

Patentes de Software

- Se define como el derecho brindado a una persona jurídica o natural sobre una invención de un dispositivo ,software método de desarrollo del mismo, en el ambiente de las TIC.
- Se dice que las patentes de software en cierta forma han logrado crear practicas negativas en la industria del software entre las cuales podemos mencionar:

- Permite e Incentiva practicas monopolicas en el mercado (ejem: Microsoft)
- La cantidad de patentes para producir un solo producto de software puede ser del orden de miles, mientras que otras áreas de desarrollo tecnológico suele ser bastante limitada la cantidad de patentes para un solo producto.
- La cantidad de inversión para patentar un software es mínima comparada con la de otras áreas tecnológicas.

- En el caso de las patentes de Software detectar si un producto tiene código que esta sujeto a patentes por lo general tiene un costo muchas veces superior a la creación del código afectado.
- El tiempo de aprobación de patentes anda entre 5 a 10 años, en una industria como la del software esto es inaceptable debido a que las tecnologías cambia según las leyes de Moore.

Licencias de Software

- La Licencia representa el derecho legal de instalación y uso del software (no la propiedad). La licencia es el documento donde se establecen las condiciones en que cada propietario del software permite utilizar su software.
- una licencia permite el uso de una versión, idioma y plataforma determinada del software únicamente, Aunque la tendencia es simplificar dichas restricciones y muchas compañías tienen licencias multiplataformas, multilingües.

Enfoques en el licenciamiento de software.

- **Tradicional:** Es que cada usuario que instale el programa en su PC debe pagar por una licencia o sea 1 licencia por computadora.
- **En red:** Es instalados en varias maquinas que se encuentra en una red, este se basa en la concurrencia, es decir que este se basa en el hecho de que un programa puede ser ocupado por más de una persona simultáneamente, pero nunca por todos al mismo tiempo.

- **licenciamientos especiales o corporativos:**
Este se basa por el volumen, se orienta mas que todo a grandes corporaciones, instituciones académicas o gobiernos.

Este es el factor crítico y estratégico por excelencia en la industria del software, además este es el factor de éxito o fracaso de muchas organizaciones.

Algo que hay que tener cuidado es en la gestión de los recursos humanos ya que estos debe gestionarse para obtener una estructura del personal que sea eficiente, flexible a los cambios, multidisciplinar en las capacidades y habilidades de las personas que lo integran y además que cada uno de los miembros de la misma debe saber que hacer y como integrar su trabajo sus estimaciones, métricas, gestiones y puntos de vista propios a los del grupo que desarrolla que desarrolla un proyecto de ingeniería de software.

El personal subdividido en equipos debe poseer un líder de grupo el cual integre en su personalidad una serie de habilidades y capacidades tales como dotes de gestión, resolución de problemas, motivación, organización, disciplina, innovación, e influencia sobre los cada uno de los miembros y del equipo en general

A las personas que forman parte de los recursos humanos se les conoce con el nombre de participante, para estos existen 5 categorías en un proyecto y gestión del desarrollo de software las cuales son:

-Gestor Ejecutivo:

Son los que define modelos de negocio, a los cuales el proyecto de software debe alinearse.

-Gestor de Proyecto: Son quienes planifican, organizan, coordinan, motivan y controlan a los profesionales de software.

-Profesionales:

Son quienes proporcionan las habilidades y conocimientos técnicos para la ejecución de un proyecto de software.

-Clientes:

-Son quienes especifican alcances y requisitos del software.

-Usuarios Finales:

-Son quienes interactúan con el producto de software terminado.



El ámbito de producción del software es muy variado y depende del entorno de desarrollo de un proyecto o negocio, esto es por ejemplo desde el programador que sólo desarrolla un producto de software específico, hasta las factorías o grupos de desarrollo de miles de personas trabajando en forma coordinada en la producción de software y / o sus componentes.

Para el proceso de construcción de software existen una serie de modelos entre los cuales figuran:

Modelo en Cascada:

El más conocido, esta basado en el ciclo convencional de una ingeniería

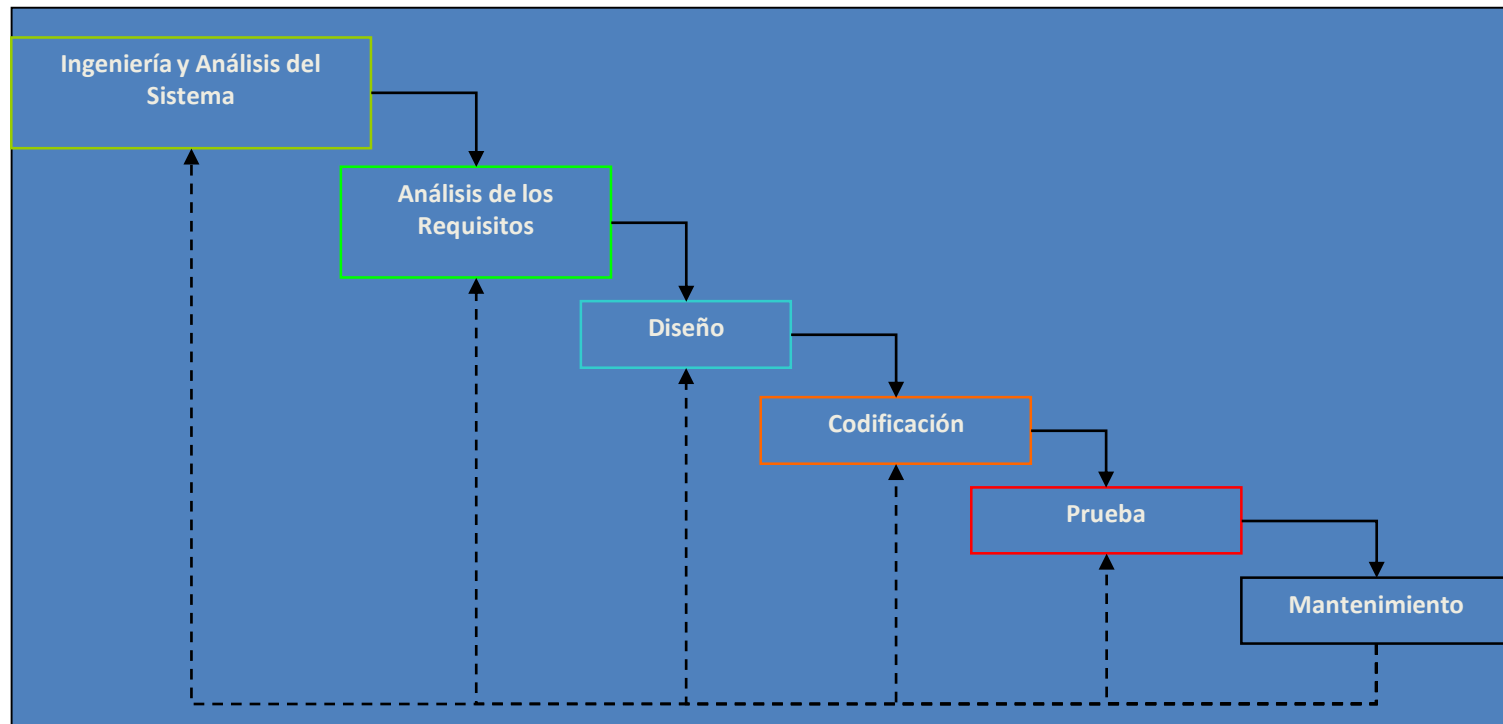


Diagrama de etapas en un modelo de cascada

Ingeniería y Análisis del Sistema: Debido a que el software es siempre parte de un sistema mayor el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al software.

Análisis de los requisitos del software: el proceso de recopilación de los requisitos se centra e intensifica especialmente en el software. El ingeniero de software (Analistas) debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas.

Diseño: el diseño del software se enfoca en cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación.

Codificación: el diseño debe traducirse en una forma legible para la maquina.
El paso de codificación realiza esta tarea. Si el diseño se realiza de una manera detallada la codificación puede realizarse mecánicamente.

Prueba: una vez que se ha generado el código comienza la prueba del programa.
La prueba se centra en la lógica interna del software, y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren.

Mantenimiento: el software sufrirá cambios después de que se entrega al cliente.
Los cambios ocurrirán debido a que hayan encontrado errores, a que el software deba adaptarse a cambios del entorno externo (sistema operativo o dispositivos periféricos), o debido a que el cliente requiera ampliaciones funcionales o del rendimiento.

Modelo Incremental:

El Modelo Incremental combina elementos del Modelo de Cascada con la filosofía interactiva de construcción de prototipos. En una visión genérica, el proceso se divide en 4 partes: Análisis, Diseño, Código y Prueba.

Se mantiene al cliente en constante contacto con los resultados obtenidos en cada incremento. Es el mismo cliente el que incluye o desecha elementos al final de cada incremento a fin de que el software se adapte mejor a sus necesidades reales. El proceso se repite hasta que se elabore el producto completo. De esta forma el tiempo de entrega se reduce considerablemente.

El Modelo Incremental es particularmente útil cuando no se cuenta con una dotación de personal suficiente. Los primeros pasos los pueden realizar un grupo reducido de personas y en cada incremento se añadirá personal, de ser necesario. Por otro lado los incrementos se pueden planear para gestionar riesgos técnicos.

Modelo Evolutivo:

Este modelo se utiliza cuando los requisitos del software son muy cambiantes. Estos son modelos iterativos, donde se desarrollan versiones del software, cada nueva versión es más completa y contempla nuevas funcionalidades.

Como el modelo de desarrollo incremental, el modelo de desarrollo evolutivo (algunas veces denominado como prototipado evolutivo) construye una serie de grandes versiones sucesivas de un producto.

En el modelo evolutivo, los requerimientos son cuidadosamente examinados, y sólo esos que son bien comprendidos son seleccionados para el primer incremento. Los desarrolladores construyen una implementación parcial del sistema que recibe sólo estos requerimientos. El sistema es entonces desarrollado, los usuarios lo usan, y proveen retroalimentación a los desarrolladores. Basada en esta retroalimentación, la especificación de requerimientos es actualizada, y una segunda versión del producto es desarrollada y desplegada.

Modelo Basados en componentes:

Es evolutivo por naturaleza y exige un enfoque interactivo para la creación del software. Sin embargo, el modelo ensamblador de componentes configura aplicaciones desde componentes separados del software (algunas veces llamados "clases"). Esto se debe gracias a que, si se diseñan y se implementan adecuadamente, las clases orientadas a objetos son reutilizables por las diferentes aplicaciones y arquitecturas de sistemas basados en computadoras.

En primer lugar se identifica las clases candidatas examinando los datos que se van a manejar por parte de la aplicación y el algoritmo que se va a crear para conseguir el tratamiento.

- aspectos de la industria de software

- En la economía moderna, la Industria de Software es la piedra angular de toda innovación tecnológica, ya que impacta en forma directa sobre todas las actividades económicas. La demanda por los productos de software y los servicios de información tecnológica tiene una de las tasas de crecimiento mundiales más altas en la actualidad.

- La globalización juega un papel importante en el desarrollo de la industria de software, ya que esta tiene una complejidad y competitividad fuertes y donde la conjunción de elementos y componentes innovadores pensados, desarrollados y producidos en distintas partes del mundo, se conjugan en una oferta única.

- El desarrollo de las actividades de investigación y desarrollo constituyen elementos críticos para el desarrollo de una industria como es la del software.

País	I&D / PIB	Año
Japón	2.98	2000
Estados Unidos	2.82	2001
Alemania	2.53	2001
Brasil	1.05	2000
España	0.97	2001
Chile	0.57	2001
México	0.40	2000
Argentina	0.39	2002
Bolivia	0.28	2001
Uruguay	0.24	2000
Colombia	0.16	2001
Ecuador	0.08	2000

- Otro indicador de la capacidad de generar innovaciones de los países esta dado por la cantidad de patentes registradas.

País	Cantidad de Patentes	Año
Estados Unidos	166,000	2000
Japón	121,000	2000
Alemania	48,000	2000
Brasil	3,589	2000
Argentina	1,291	2000
Chile	601	2000

- La investigación y desarrollo se encarga del análisis y selección de tecnologías, de la definición técnica del producto, de la elaboración de prototipos y de la definición del proceso de fabricación o producción masivo del producto. Estaríamos hablando de actividades como la evaluación de tecnologías y arquitectura, de la especificación de requisitos y, en general, del desarrollo y prueba de producto de software así como la definición de los procedimientos de instalación, configuración y explotación.

- Así pues la investigación está orientada, principalmente, a mejorar oportunidades de mercado, para anticipar la competencia y mejorar la capacidad de la empresa.
- Por su parte el sector de desarrollo debe enfocarse en proveer servicios de información más que productos de software, por cuanto la tendencia mundial es a requerir soluciones completas más que productos específicos.

- aspectos de la industria de software

- Muchas de las principales aplicaciones de las tecnologías de información y comunicaciones beneficiosas para el desarrollo social y económico no necesitan una infraestructura de telecomunicaciones altamente sofisticada.

- A medida que el desarrollo basado en el conocimiento vaya incidiendo más en el orden económico y social mundial, las redes de comunicación avanzadas jugarán un papel más importante en la adquisición e intercambio de información. Los países en desarrollo que no tengan una infraestructura de comunicación no podrán aprovechar las nuevas oportunidades que ofrecen estas redes.

- Sin una infraestructura de comunicaciones apropiada, las tecnologías de información y comunicaciones podrían acentuar, en vez de aminorar, las disparidades existentes en ingresos, riqueza y oportunidades.

Infraestructura

- Es la infraestructura la encargada de brindar a la industria la capacidad de proyectar su desarrollo para la exportación.

Tendencias futuras

- aspectos de la industria de software

Tendencias futuras

- Como en cualquier industria, las tendencias en el mercado afectan los métodos de comercialización, las ventas, la definición de productos y su logística.

Tendencias futuras

- Application Service Provider (ASP)
- Es proveer mediante la Internet los servicios de software de paquetes.
- Un usuario puede acceder al mismo desde cualquier punto geográfico o utilizando diversos dispositivos fijos.

Tendencias futuras

- Comercio Electrónico (e - Commerce)
- Ya hay muchos casos de éxito referenciales en este sentido, tanto de B2C como de B2B.

Tendencias futuras

- E - Learning, e - Government, e - Salud
- Internet está proyectando modelos de comercialización y logística masivos, de fácil acceso a la información y servicios basados en Tecnología de Información.

Tendencias futuras

- Complejidad
- Los computadores cada día crecen en potencia y capacidad y decrecen en precio y tamaño.

Tendencias futuras

- Consolidación
- Por sus características, las compañías chicas con productos tecnológicamente exitosos, necesitan estructuras comerciales y de servicios capaces de proyectar el producto a un mercado mayor.

Tendencias futuras

- Software Libre y/o de código abierto
- la proliferación de Linux y la base de sus características lo hacen hoy en esquema exitoso de creación, mejoramiento y difusión de un producto.

Tendencias futuras

- Tele-trabajo
- del concepto de tele - trabajo, trabajo a distancia o trabajo en el hogar difundido por el costo decreciente y la calidad de la telecomunicaciones y el hardware, así como también el desarrollo de la tecnología de información y el e- commerce cambió los hábitos de vida y laborales de la gente.