

## TAREA 1: PERTINENCIA DEL TEMARIO

**Josué David Hernández Ramírez.**

Escuela Superior de Cómputo  
Instituto Politécnico Nacional, México  
*jhernandezr1605@alumno.ipn.mx*

### 1 Temas

#### 1.1 Ingeniería de Software

##### 1.1.1 Conceptos básicos de ingeniería de software.

###### **¿Qué es software?**

Programas de cómputo y su documentación asociada: requerimientos, modelos de diseño y manuales de usuario, puede ser creado desarrollando nuevos programas, configurando sistemas de software genérico o reutilizando software existente.

###### **¿Qué es la ingeniería de software?**

Una disciplina de la Ingeniería que concierne a todos los aspectos de la producción de software, utilizando las herramientas y técnicas apropiadas para resolver el problema planteado, de acuerdo a las restricciones de desarrollo y a los recursos disponibles.

###### **¿Qué es un proceso de software?**

Un conjunto estructurado de actividades cuya meta es el desarrollo o evolución de un software.

Algunas actividades genéricas en todos los procesos de software son:

1. Especificación: qué debe hacer el software y cuáles son sus especificaciones de desarrollo.
2. Desarrollo: producción del sistema de software Validación, verificar que el software cumple con lo solicitado por el cliente.
3. Evolución: cambiar/adaptar el software a las nuevas demandas.

**¿Qué es un modelo de proceso de software?**

Representación formal y simplificada de un proceso de software, presentada desde una perspectiva específica.

Modelos Genéricos:

1. Cascada, separar en distintas fases de especificación y desarrollo.
2. Desarrollo Iterativo, la especificación, desarrollo y validación están interrelacionados.
3. Prototipo, un modelo sirve de prototipo para la construcción del sistema final.
4. Basado en componentes, asume que partes del sistema ya existen y se enfoca a su integración.

**¿Cuáles son los costos de la ingeniería de software?** El costo total de un software esta dividido aproximadamente de la siguiente forma:

- 60% costos de desarrollo
- 40% costos de pruebas
- Los costos dependen del tipo de sistema que se desarrolla y de los requerimientos del mismo tales como desempeño y confiabilidad, la distribución de los costos depende del modelo de desarrollo empleado.

**¿Qué es CASE?** CASE es Computer-Aided Software Engineering son programas que son usados para dar soporte automatizado a las actividades del proceso de software como:

- Las herramientas CASE son comúnmente usadas para dar soporte a los métodos de software.
- Módulos de análisis que verifican que las reglas del método se cumplan.
- Generadores de reportes que facilitan la creación de la documentación del sistema
- Generadores de código a partir del modelo del sistema.

**1.1.2 Atributos y características del software**

**Características** Para poder comprender lo que es el software (y consecuentemente la ingeniería del software), es importante examinar las características del software que lo diferencian de otras cosas que los humanos pueden construir.

1. El software se desarrolla, no se fabrica en un sentido clásico.

2. **El software no se estropea.**
3. **Aunque la industria tiende a ensamblar componentes, la mayoría del software se construye a medida.**

**Atributos** El software debe proveer la funcionalidad y desempeño requeridos por el usuario y debe ser mantenible, confiable, eficiente y aceptable.

- **Mantenible:** El software debe poder evolucionar.
- **Confiable:** No debe causar daños económicos o físicos.
- **Eficiente:** No desperdiciar recursos del sistema.
- **Aceptable:** Los usuarios deben de aceptarlo.
- Debe ser entendible, utilizable y compatible con otros sistemas.

### 1.1.3 Importancia y aplicación del software

#### Importancia del Software

Cada software desarrolla funciones específicas dentro de una diversa gama de aplicaciones, y sin duda alguna uno de los programas que mayor utilidad representa dentro de una empresa, son los denominados Sistemas de Soporte a la Decisión (DSS).

De esta manera, la toma de decisiones se convierte en una variable crítica de éxito dentro de las empresas, y es aquí donde radica la importancia de un DSS.

#### Aplicaciones del Software

El software puede aplicarse en cualquier situación en la que se haya definido previamente un conjunto específico de pasos procedimentales.

El contenido y el determinismo de la información son factores importantes a considerar para determinar la naturaleza de una aplicación de software. El contenido se refiere al significado y a la forma de la información de entrada y salida. El determinismo de la información se refiere a la predictibilidad del orden y del tiempo de llegada de los datos.

Las siguientes áreas del software indican la amplitud de las aplicaciones potenciales:

- **Software de sistemas:** es un conjunto de programas que han sido escritos para servir a otros programas.  
Algunos programas de sistemas (por ejemplo: compiladores, editores y utilidades de gestión de archivos) procesan estructuras de información complejas pero determinadas.

- **Software de tiempo real:** coordina, analiza, controla sucesos del mundo real conforme ocurren, se denomina de tiempo real.
- **Software de gestión:** Las aplicaciones en esta área re estructuran los datos existentes para facilitar las operaciones comerciales o gestionar la toma de decisiones.  
Además de las tareas convencionales de procesamiento de datos, las aplicaciones de software de gestión también realizan cálculo interactivo (por ejemplo: el procesamiento de transacciones en puntos de ventas).
- **Software de ingeniería y científico:** está caracterizado por los algoritmos de manejo de números. Las aplicaciones van desde la astronomía a la vulcanología, desde el análisis de la presión de los automotores a la dinámica orbital de las lanzaderas espaciales y desde la biología molecular a la fabricación automática.
- **Software empotrado:** reside en memoria de sólo lectura y se utiliza para controlar productos y sistemas de los mercados industriales y de consumo. El software empotrado puede ejecutar funciones muy limitadas y curiosas (por ejemplo: el control de las teclas de un horno de microondas) o suministrar una función significativa y con capacidad de control (por ejemplo: funciones digitales en un automóvil, tales como control de la gasolina, indicadores en el salpicadero, sistemas de frenado, etc.).
- **Software de computadoras personales:** . El procesamiento de textos, las hojas de cálculo, los gráficos por computadora, multimedia, entretenimientos, gestión de bases de datos, etc.
- **Software basado en Web:** Las páginas Web buscadas por un explorador son software que incorpora instrucciones ejecutables y datos.
- **Software de inteligencia artificial:** hace uso de algoritmos no numéricos para resolver problemas complejos para los que no son adecuados el cálculo o el análisis directo.

#### 1.1.4 Ciclo de vida del software

Un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso.

### 1.1.5 Modelos de procesos

#### ¿Qué es un modelo de proceso de software?

Representación formal y simplificada de un proceso de software, presentada desde una perspectiva específica.

Ejemplos de perspectiva del proceso del hardware.

- Flujo de trabajo, secuencia de actividades.
- Flujo de datos, flujo de la información.
- Rol/acción, quien realiza qué.

#### Modelos genéricos

1. **Lineal secuencial:** sugiere un enfoque sistemático, secuencial, para el desarrollo del software. Comprende el análisis, diseño, codificación, pruebas y mantenimiento.
2. **Cascada:** Tiene las mismas características que el modelo lineal como se había mencionado anteriormente pero en este se tiene la capacidad de regresar si se detecta un error en cualquiera de las etapas de planeación, desarrollo, diseño, pruebas y mantenimiento del software.
3. **Incremental:** El modelo incremental combina elementos del modelo lineal secuencial con la filosofía interactiva de construcción de prototipos. El software se ve como una integración de resultados sucesivos obtenidos después de cada interacción.
4. **Desarrollo rápido de aplicaciones:** es un modelo de proceso del desarrollo del software lineal secuencial que enfatiza un ciclo de desarrollo extremadamente corto.  
El modelo DRA es una adaptación a alta velocidad del modelo lineal secuencial en el que se logra el desarrollo rápido utilizando una construcción basada en componentes.
5. **Prototipos:** Comienza con la recolección de requisitos. El desarrollador y el cliente encuentran y definen los objetivos globales para el software, identifican los requisitos conocidos y las áreas del esquema en donde es obligatoria más definición.  
El prototipo lo evalúa el cliente/usuario y se utiliza para refinar los requisitos del software a desarrollar. La iteración ocurre cuando el prototipo se pone a punto para satisfacer las necesidades del cliente, permitiendo al mismo tiempo que el desarrollador comprenda mejor lo que se necesita hacer.

6. **Esprial:** La meta del modelo espiral del proceso de producción del software es proporcionar un marco para diseñar tales procesos.
7. **Basados en componentes:** asume que partes del sistema ya existen y se enfoca a su integración.

## 1.2 Proceso de gestión de proyecto

### 1.2.1 Ámbito del software

El ámbito del software describe el control y los datos a procesar, la función, el rendimiento, las restricciones, las interfaces y la fiabilidad; Se evalúan las funciones descritas en la declaración del ámbito, y en algunos casos se refinan para dar más detalles antes del comienzo de la estimación.

Dado que las estimaciones del coste y de la planificación temporal están orientadas a la función, muchas veces es útil llegar a un cierto grado de descomposición. Las consideraciones de rendimiento abarcan los requisitos de tiempo de respuesta y de procesamiento.

### 1.2.2 Estudio de factibilidad

Los autores Putnam y Myers tratan este aspecto cuando escriben que: no todo lo imaginable es factible ni siquiera en el software.

La factibilidad del software tiene cuatro dimensiones sólidas:

- Tecnología ¿Es factible un proyecto técnicamente? ¿Está dentro del estado actual de la técnica?.
- Financiamiento ¿Es factible financieramente? ¿Puede realizarse a un coste asumible por la empresa de software y por el cliente?
- Tiempo ¿Pueden los proyectos adelantarse a los de la competencia?
- Recursos ¿La organización cuenta con los recursos suficientes para tener éxito?

La respuesta es sencilla depende de la experiencia, ya que puede que se haya hecho antes algún proyecto de este tipo o puede que no se tenga experiencia en el proyecto y por lo tanto no son fáciles.

### **1.2.3 Análisis de riesgo**

El riesgo se mide por el grado de incertidumbre en las estimaciones cuantitativas establecidas por recursos, coste y planificación temporal. Si no se entiende bien el ámbito del proyecto o los requisitos del proyecto están sujetos a cambios, la incertidumbre y el riesgo son peligrosamente altos.

Lo que es más importante, el cliente y el planificador, deben tener presente que cualquier cambio en los requisitos del software significa inestabilidad en el coste y en la planificación temporal.

### **1.2.4 Recursos**

### **1.2.5 Estimación**

### **1.2.6 Planificación del proyecto**

### **1.2.7 Supervisión y control del plan del proyecto**