

Guía para el examen de Desarrollo de software I

¿Qué es el software?

- El software son los programas junto con todos los documentos asociados y la configuración de datos que se necesitan para hacer que estos operen de manera correcta.

¿Qué es la ingeniería de software?

- Es una disciplina que comprende todos los aspectos de la producción del software desde las etapas iniciales de la especificación del sistema.

Importancia de la ingeniería de software

- Las instituciones comenzaron a darse cuenta de que no eran capaces de producir el software requerido
- **Causas de la crisis del software**
 - o Inicio en los 60, en donde el escenario dominado principalmente por el hardware a otro en donde el software se volvió más importante y complejo. Hardware -> Software
 - o Incremento del esfuerzo requerido en el desarrollo de software debido a la complejidad de los nuevos productos de hardware.

¿Cuáles son las principales causas de la crisis del software?

1. El incremento de la complejidad de los productos
2. Lento crecimiento de la productividad de los programadores
3. Creciente demanda de software
 - o **Al aumentar el tamaño y complejidad del software se requería que la productividad de los desarrolladores fuera mayor, pero no era así.**

¿Qué busca el desarrollo de software?

- Busca pasar una idea abstracta, generada por ejemplo por un cliente, a un sistema concreto que implemente la idea de forma correcta.

Proceso de software

- Indica la forma en la que se puede dividir una tarea compleja (no manejable) en pequeños procedimientos que pueden ser manejados y resueltos de forma individual
- **Características generales**
 - o Sistemático: Indica un conjunto de procesos a seguir
 - o Formal o semi-formal: Describe como debe ser desarrollado el software
- **Enfoques del proceso de software**
 - o Cascada
 - o Prototipo
 - o Programación extrema y procesos ágiles

Fases del desarrollo de software

- **Planeación:** El ciclo de vida de desarrollo de sistemas es el proceso de determinar como un sistema de información puede ayudar a solventar las necesidades de negocios, el diseño de sistemas, la construcción de estos y liberarlos a los usuarios. Este ciclo de vida es común en la mayoría de los proyectos y sirve como un marco de trabajo para entender como son realizados los proyectos de sistema de información.
- **Analista de sistemas:** Pueden servir como agentes de cambio quienes identifican necesidades de mejoras organizacionales, diseñan sistemas para implementar dichos cambios, y entrenan y motivan a otros para usar el sistema. Los analistas de sistemas deben de entender como aplicar la tecnología para resolver problemas de negocios. Estos trabajan de forma cercana con los miembros del equipo de un proyecto de forma que el equipo desarrolle el sistema correcto en una forma efectiva.
 - o **Habilidades de una analista:** Entender que cambiar, saber cómo cambiarlo y convencer a otros de la necesidad de cambiarlo requiere de un amplio rango de habilidades. Estas pueden ser organizadas en 6 principales categorías:
 - **Técnicas:** Entender los ambientes técnicos existentes en la organización, los fundamentos de las nuevas tecnologías y las formas en las cuales ambas pueden integrarse en una solución técnica.
 - **De negocios:** Son requeridas para entender como la tecnología de la información puede ser aplicada a situaciones de negocios y asegurar obtener un valor de negocio real.
 - **Analíticas:** Resuelven problemas de forma continua tanto a nivel de proyectos como a nivel organizacional y coloca sus habilidades analíticas bajo prueba de forma regular.
 - **Interpersonales:** Requieren comunicarse de forma efectiva uno a uno con usuarios y administradores de negocios y los programadores.
 - **Administrativas:** Deben de ser capaces de realizar presentaciones a grupos pequeños, grandes y escribir reportes.
 - **Éticas:** Debe ser justo, honesto y ético con los miembros del equipo.

Ciclo de vida del desarrollo de sistemas

- **Planeación:** Esta fase es importante para entender por qué un sistema de información debe ser construido y determinar cómo el equipo del proyecto va a construirlo. Se compone de 2 pasos:
 - o **Iniciación del proyecto:** El valor de negocio para la organización es identificado - ¿cómo reducir costos o incrementar ingresos?
 - **Requerimiento de sistema:** Presenta un breve resumen de una necesidad de negocios, y explica como un sistema puede apoyar a atender la necesidad creando un valor de negocio.
 - El departamento de SI trabaja junto con el departamento que realizó el requerimiento de sistema para realizar un análisis de factibilidad. El análisis de factibilidades examina 3 aspectos clave:
 - **Factibilidad técnica:** ¿Podemos construirlo?
 - **Factibilidad económica** ¿Proveerá un valor de negocio?
 - **Factibilidad organizacional** Si lo construimos, ¿Será usado?

- **Administración del proyecto:** Durante la administración del proyecto, el administrador del proyecto crea un plan de trabajo, asigna al personal para el proyecto y apoya a mantener la administración del equipo del proyecto y dirige el proyecto hacia el SDLC.
- **Análisis:** Responde las preguntas de:
 - ¿Quién va a usar el sistema?
 - ¿Qué hará el sistema?
 - ¿Dónde?
 - ¿Cuándo será usado?

Durante esta fase, el equipo del proyecto investiga cualquier sistema actual, identifica oportunidades de mejora y desarrolla un concepto para el nuevo sistema. Esta fase se divide en 3 partes:

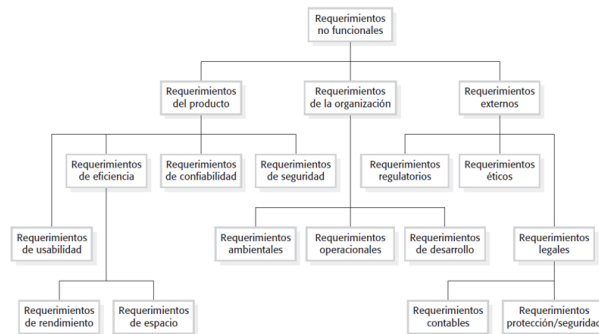
- Una estrategia de análisis es desarrollada para guiar los esfuerzos del equipo del proyecto. Tal estudio puede incluir el estudio del sistema actual y sus problemas y formas imaginativas para diseñar un nuevo sistema
- Reunir requerimientos **(a través de entrevistas, cuestionarios, etc.)**. El análisis de la información recabada lleva al desarrollo de un nuevo concepto. Los conceptos son usados para crear modelos de análisis que describen el funcionamiento del sistema.
- Los análisis, conceptos del sistema y los modelos son combinados dentro de un documento llamado **propuesta del sistema**, la cual es presentada al patrocinador del proyecto para que decidan si el proyecto puede continuar.
- **Diseño:** Decide **Cómo** el sistema va a operar en términos de hardware, software e infraestructura de red; la interfaz de usuario, formularios y reportes a ser usados; los programas específicos, bases de datos y archivos que serán requeridos. Esta fase se divide en 4 partes:
 - **La estrategia de diseño debe ser determinada.** Esto clarifica si el sistema va a ser desarrollado por programadores de la propia compañía, si el desarrollo será asignado a otra compañía o si se comprará un paquete de software existente.
 - Se debe realizar el **diseño de la arquitectura base** para el sistema, que describa el hardware, software y red que será utilizada. Se definirá el **diseño de la interfaz** que indicará cómo el usuario puede moverse a través del sistema y los **formularios y reportes** a ser utilizados.
 - **La especificación de la base de datos y archivos será desarrollada.** Esto define con exactitud “qué” datos serán almacenados y “dónde” serán almacenados.
 - El equipo de analistas desarrolla el diseño del sistema el cual define los programas que deben ser escritos y **qué** hará cada programa.
- **Implementación:** Es la fase final del SDLC durante la cual el sistema es construido (o adquirido). Esta fase requiere de más tiempo y atención. Se divide en 3 pasos:
 - **Construcción del sistema:** El sistema es construido y probado para asegurar que el desempeño sea el diseñado. Debido a que el costo de arreglar bugs puede ser inmenso, las pruebas son una de las fases más críticas en la implementación.

- **La instalación del sistema:** Es el proceso donde un sistema antiguo es retirado y un sistema nuevo es puesto en funcionamiento.
- **Plan de apoyo para el sistema:** Este plan usualmente incluye una revisión post-implementación (formal o informal) así como una forma sistemática para identificar necesidades de cambios mayores o menores del sistema.

Requerimientos

- Ingeniería de requerimientos (IR): Es el proceso de establecer los servicios que el cliente requiere del sistema de software. Los resultados de la IR son la **Especificación de requerimientos de un software (ERS)**.
- Requerimientos: Pueden ser de alto nivel, descripciones abstractas y especificaciones que van desde sketches hasta representaciones formales (representaciones matemáticas)
 - Requerimientos vs Metas
 - **Metas:** Son objetivos de un negocio, organización o sistema.
 - **Requerimientos:** Especifica como una meta puede ser alcanzada mediante el sistema propuesto.
 - Evolución de los requerimientos
 - Requerimientos de usuario: Son oraciones abstractas escritas en lenguaje natural y acompañado con diagramas informales.
 - Requerimientos del sistema: Son descripciones detalladas de los servicios y de las restricciones. Usualmente hacen referencia a especificaciones de funcionalidad o características tecnológicas, todos estos **obtenidos a partir del análisis de los requerimientos de usuario**. Estos son utilizados en las especificaciones del contrato entre el cliente y el contratista por lo que deben ser **estructurados y precisos**. En muchas ocasiones los **casos de uso** juegan un papel importante.
 - Especificaciones del diseño del software: Surgen de la documentación, del análisis de los requerimientos de usuario y requerimientos del sistema.
 - Tipos de requerimientos
 - **Requerimientos funcionales:** Describen los servicios que el sistema debe de proveer y como debe reaccionar el sistema a cierto tipo de entradas. Pueden ser de alto nivel y generales o pueden ser detalladas expresando entradas, salidas, excepciones, etc. Pueden ser representados en:
 - Lenguaje natural
 - Modelos visuales
 - Métodos formales
 - **Requerimientos no funcionales:** Son impuestos por el entorno en el cual, el sistema opera. Ejemplos:
 - Límites de tiempo
 - Propiedad de calidad
 - Adherencia a estándares
 - El uso de lenguajes de programación
 - **Requerimientos obtenidos del dominio o tipo de aplicación**

Requerimientos



- Software: Es una descripción abstracta de un conjunto de elementos computacionales que resulta en un sistema concreto.

Requerimientos, objetivos y conceptos básicos

- Stakeholders: Representa una amplia clase de individuos quienes tienen algún interés (stake) en el **éxito (o falla)** del sistema en cuestión.
- Fuente de los requerimientos:
 - o Stakeholder
 - o Dominio de la aplicación
 - o Documentación
- Identificar el propósito del sistema: Requiere de la identificación y definición de requerimientos del sistema, esto resulta ser una **Actividad compleja**:
 - o Complejidad propia del propósito/requerimientos.
 - o Frecuentemente la gente no sabe que es lo que quiere hasta que se le muestra un prototipo.
 - o Requerimientos que cambian con el tiempo.
 - o Stakeholders con conflictos con los requerimientos o metas.
- Falta de integridad: Se refiere al hecho de que frecuentemente es difícil identificar todos los requerimientos:
 - o Recolección de requerimientos incompletos
 - o Software con carencia de funcionalidad
- Falta de pertenencia: Hace referencia a la relevancia de los requerimientos.
- Estrategia: Definir el sistema como un conjunto de misiones u objetivos.

¿Por qué los requerimientos irrelevantes pueden ser problemáticos?

- Pueden introducir inconsistencias
- Pueden gastar recursos del proyecto

Ingeniería de requerimientos

- IR es un conjunto de actividades relacionadas con identificar y comunicar el propósito de un sistema intensivo de software, y el contexto en el cual será utilizado.

Especificación de requerimientos

- Descripción formal de las tareas que el sistema a construir debe realizar para satisfacer las necesidades del cliente

Estándar IEEE 830 para la especificación de requisitos

- Estructura:
 - Introducción: En esta sección se proporcionará una introducción a todo el documento de ERS.
 - Propósito
 - Propósito del documento
 - A quien va dirigido
 - Ámbito del sistema
 - Nombre del sistema
 - Lo que hará el sistema y lo que no
 - Beneficios, objetivos y metas
 - Se referencian todos aquellos documentos de nivel superior
 - Definiciones, acrónimos y abreviaturas
 - Se definen todos los términos, acrónimos y abreviaturas
 - Referencias
 - Lista de todos los documentos referenciados
 - Visión General del Documento
 - Describe los contenidos y la organización del documento
 - Descripción general: Se describen todos aquellos factores que afectan al producto y a sus requisitos (su contexto).
 - Perspectiva del producto
 - Relación con otros productos
 - Dependencia de otros servicios
 - Funciones del producto
 - Resumen a grandes rasgos de las funciones del futuro sistema
 - Características de los usuarios
 - Describe las características generales de los usuarios del producto.
 - Restricciones
 - Describe las limitaciones que se imponen sobre los desarrolladores del producto
 - Suposiciones y dependencias
 - Describe factores que, si cambian, pueden afectar a los requisitos.
 - Requisitos futuros
 - Mejoras del sistema futuras
 - Requisitos específicos: Todo requisito aquí especificado describirá comportamientos externos del sistema, perceptibles por parte de los usuarios, operadores y otros sistemas
 - Interfaces externas
 - Describe los requisitos que afectan a la interfaz de usuario, interfaz con otros sistemas e interfaz de comunicaciones.
 - Funciones
 - Se especifican todas aquellas acciones (funciones) que deberá llevar a cabo el software
 - Por ejemplo: El sistema deberá.

- Requisitos de rendimiento
 - Se detallan los requisitos relacionados con la carga que se espera tener que soportar el sistema.
- Restricciones de diseño
 - Todo aquello que restrinja las decisiones relativas al diseño de la aplicación
- Atributos del sistema
 - Se detallan los atributos de calidad del sistema, fiabilidad, mantenibilidad, portabilidad y, muy importante, la seguridad.
- Otros requisitos
 - Cualquier otro requisito que no encaje en otra sección
- Apéndices
 - Puede contener todo tipo de información relevante para la ERS pero que, propiamente, no forme parte de la ERS, por ejemplo:
 - Formatos de entrada/salida de datos, por pantalla o en listados.
 - Resultados de análisis de costes
 - Restricciones acerca del lenguaje de programación