

ACTIVIDADES DESARROLLO SOFTWARE

1. ¿Qué se hace en la fase de análisis del desarrollo de una aplicación? ¿Qué tipos de requisitos se especifican en esta fase?

Lo que se hace en la fase de análisis del desarrollo de una aplicación es:

- **Entrevistas:** es hablar con el cliente.
- **Desarrollo conjunto de aplicaciones:** entrevista muy estructurada aplicable a grupos de personas.
- **Planificación conjunta de requisitos:** es un subconjunto de JAD, se caracteriza por estar dirigidas a la alta dirección y en consecuencia los productos resultantes son los requisitos de alto nivel estratégico.
- **Brainstorming:** reuniones en grupo con el objetivo de generar ideas de distintos puntos de vista para dar solución a un problema.
- **Prototipos:** versión inicial del sistema, se usa para clasificar algunos puntos, demostrar los conceptos, para adentrarse en los problemas y buscar soluciones.
- **Casos de uso:** Es la técnica definida en UML, se basa en escenarios que describen como se usa el software en una determinada situación.

Por otra parte los requisitos necesarios pueden ser; funcionales (que describen con detalle la función que realiza el sistema, cómo reacciona ante determinadas entradas, como se comporta en situaciones particulares, etc.) o no funcionales (que tratan sobre las características del sistema, como puede ser la fiabilidad, mantenibilidad, sistema operativo, plataforma hardware, restricciones, limitaciones, etc.)

2. ¿Cuáles son las notaciones gráficas que se usan en la fase de diseño? Explica en una frase para qué sirve cada una.

Las notaciones gráficas que se usan en la fase de diseño son:

- Diagrama de flujo: es una herramienta para el diseño procedimental, se usan símbolos para indicar cada parte del proceso (cuadrado, triangulo, flecha).
- Diagrama de cajas: es para desarrollar una representación de diseño procedimental.
- Tabla de decisión: representa en modo de tabal las condiciones y las acciones que se llevan a cabo combinando las condiciones del modulo de programa.
- Pseudocódigo: es un texto descriptivo para realizar el diseño de un algoritmo.

3. Explica de forma breve los modelos de ciclo de vida que hemos visto

Ciclo de vida en cascada: las etapas para el desarrollo del software tienen un orden, hay para empezar una etapa hay que finalizar la anterior, en este ciclo si se cambia algo de alguna etapa, a partir de esa etapa se tiene que modificar todo lo siguiente.

Ciclo de vida en cascada con retroalimentación: es como el anterior salvo que se introduce una retroalimentación entre etapas, de forma que se puede volver atrás en cualquier momento, y modificar.

Ciclo de vida evolutivo: los hay de dos tipos, modelo Iterativo Incremental y Modelo en Espiral.

Modelo Iterativo Incremental: está basado en el modelo en cascada con retroalimentación, donde las fases se repiten y refinan, mejorando las fases siguientes.

Modelo en Espiral: es una combinación entre el modelo Iterativo incremental y el de cascada. El software se va construyendo repetidamente en forma de versiones que son cada vez mejores, incrementando su funcionalidad.

4. ¿Qué lenguaje se suele utilizar para representar el diseño de un software orientado a objetos? Indica qué significan sus siglas y busca un ejemplo de éste.

El lenguaje que se suele utilizar para ello es el UML. Es un lenguaje modelado basado en diagramas que sirve para expresar modelos. UML provienen de las siglas Unified Modeling Language, Lenguaje de Modelado Unificado. Por ejemplo, diagramas estructurales de clase.

5. ¿En qué consisten las tareas de verificación y validación del software?

Verificación: conjunto de actividades que tratan de comprobar si se está construyendo el producto correctamente, (si el software implementa correctamente una función específica).

Validación: Conjunto de actividades que tratan de comprobar si es el producto correcto (si el software construido se ajusta a los requisitos del cliente).

6. ¿Qué técnicas se utilizan para llevar a cabo el diseño de casos de prueba? ¿En qué consisten?

Se usan dos técnicas:

- **Prueba de caja blanca:** se centran en validar la estructura interna del programa (los detalles procedimentales del código).
- **Prueba de caja negra:** se centran en validar los requisitos funcionales sin fijarse en el funcionamiento interno del programa (necesitan saber la funcionalidad que el código ha de proporcionar).

7 . La documentación de un proyecto software se divide en dos clases: la documentación del proceso y del producto. ¿Qué contiene cada una?

La documentación del proceso: son documentos donde se registran el proceso de desarrollo y mantenimiento. Se indican planes, estimaciones y horarios que se utilizan para predecir u controlar el proceso de software, informando de como se ha

de implementar el proceso.

La documentación del producto: esta documentación describe el producto que está siendo desarrollado. Define dos tipos, la del sistema (orientada al desarrollo y mantenimiento del mismo) y la del usuario (orientada a los usuarios que utilizan el sistema).

8. ¿Qué información debe contener la documentación del usuario y del sistema?

La documentación del usuario:

- Descripción funcional del sistema
- Documento de instalación del sistema.
- Manual introductorio.
- Manual de referencia del sistema.
- Guía del administrador del sistema.
- Tarjeta de referencia rápida.

La documentación del sistema:

- Fundamentos del sistema.
- El análisis y especificación de requisitos.
- Diseño.
- Implementación.
- Plan de pruebas del sistema.
- Plan de pruebas de aceptación.
- Los diccionarios de datos.

9. Explica con tus palabras y de forma breve las tareas que se llevan a cabo durante la fase de mantenimiento del software

Implementación del proceso: el responsable de mantenimiento se hace cargo de documentar y planear las actividades de mantenimiento según los problemas que surjan o las peticiones de los usuarios.

Análisis de problemas y modificaciones: aquí el responsable de mantenimiento, analizará el informe del problema, lo verificará y documentará las modificaciones y resultados de los análisis.

Implementación de las modificaciones: aquí se determina la documentación, viendo lo necesario para llevar a cabo el proceso de implementación de las modificaciones.

Revisión/aceptación del mantenimiento: se llevará a cabo las revisiones, para obtener la aprobación de la finalización satisfactoria de la modificación.

Migración: se preparará y documentará un plan de migración, indicando los usuarios a los que se notificará.

Retirada del software: se retirará el software por petición del propietario, dejando toda la documentación del plan de retirada que se comunicará al usuario. Se llevará a cabo en paralelo la retirada del antiguo software y la introducción del nuevo.

10. ¿Qué diferencia hay entre un lenguaje de alto nivel y de bajo nivel?

El lenguaje de bajo nivel es entendible por las máquinas, mientras que el de alto nivel es el lenguaje natural en inglés; y por otra parte el lenguaje de bajo nivel son específicos para cada ordenador, es decir son dependientes, mientras que el de alto nivel son independientes a la máquina, no dependen del hardware del ordenador y no requieren ningún conocimiento de código máquina por parte del usuario que lo utiliza, el de bajo nivel sí.

Alicia Acosta Ternero