

Tarea 2

Ciclo de vida del software (Relación 1)

1.- Define "Ciclo de vida del software".

El ciclo de vida del software es un concepto que se refiere a las fases por las que pasa un software desde un concepto inicial hasta su mantenimiento y posterior discontinuación. Las fases del ciclo de vida del software están diseñadas para servir de guía en el proceso del desarrollo de software y pueden variar dependiendo de la metodología de desarrollo (como el modelo en cascada o el modelo en v)

Las fases principales son:

- Análisis
- Diseño
- Codificación
- Pruebas
- Mantenimiento

2.- Nombra las fases principales del desarrollo de software y explica brevemente que se hace en cada una de ellas.

Las fases principales del ciclo del software son:

1. Análisis
2. Diseño
3. Codificación
4. Pruebas
5. Mantenimiento

Análisis

Se determina y define claramente las necesidades del cliente y se especifica los requisitos que debe cumplir el software a desarrollar

- La especificación de requisitos debe:
 - Ser completa y sin omisiones
 - Ser concisa y sin trivialidades
 - Evitar ambigüedades. Utilizar lenguaje formal.
 - Evitar detalles de diseño o implementación
 - Ser entendible por el cliente
 - Separar requisitos funcionales y no funcionales
 - Dividir y jerarquizar el modelo
 - Fijar criterios de validación

Diseño

Durante esta etapa, se crea un diseño detallado del software, especificando cómo se verá y cómo funcionará. Se descompone y organiza el sistema en elementos componentes que pueden ser desarrollados por separado.

Las actividades habituales son las siguientes:

- Diseño arquitectónico
- Diseño detallado
- Diseño de datos

- Diseño de interfaz de usuario

Codificación

En esta fase, los programadores escriben el código fuente del software de cada componente. Es la etapa en la que se traducen los requisitos y el diseño en código ejecutable.

Pruebas

El principal objetivo de las pruebas debe ser conseguir que el programa funcione incorrectamente y que se descubran los fallos que tiene para poder arreglarlos. Debemos someter al programa al máximo número de situaciones diferentes

Mantenimiento

Durante la explotación del sistema software es necesario realizar cambios ocasionales. Para ello hay que rehacer parte del trabajo realizado en las fases previas.

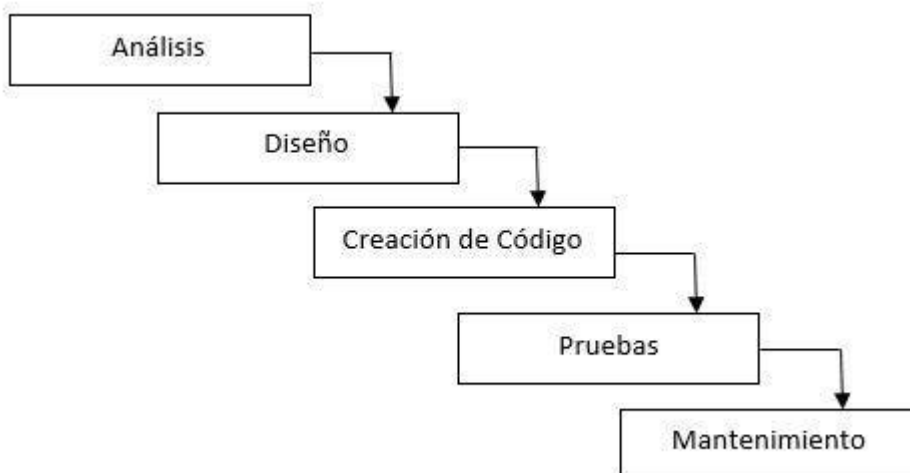
los tipos de mantenimiento son:

- Correctivo: se corrigen defectos.
- Perfectivo: se mejora la funcionalidad.
- Evolutivo: se añade funcionalidades nuevas.
- Adaptativo: se adapta a nuevos entornos.

3.- Explica brevemente en qué consiste el modelo en cascada cuando hablamos de desarrollo de software.

El modelo en cascada es un modelo de desarrollo de software con enfoque lineal que identifica las fases principales del desarrollo software y éstas se realizan siguiendo un orden rígido y establecido, de manera que el resultado de una fase es el comienzo de la siguiente. Las fases del modelo en cascada son las siguientes:

1. Análisis
2. Diseño
3. Codificación
4. Pruebas
5. Mantenimiento



4.- Ventajas e inconvenientes del modelo en cascada.

Ventajas del modelo en cascada:

- Claridad en su estructura
- Se documentan todas las etapas del desarrollo facilitando la documentación completa del proyecto.
- Fácil de gestionar al ser un modelo lineal en el que el resultado de una fase es el comienzo de la siguiente
- Funciona muy bien en proyectos pequeños cuyos requisitos estén bien definidos

Inconvenientes del modelo en cascada:

- No permite el desarrollo de más de una fase simultáneamente.
- Carece de flexibilidad para adaptarse a cambios en el desarrollo.
- El cliente no ve el producto hasta etapas finales y puede que no cumpla con las expectativas del cliente..
- Si los requisitos no están bien definidos el producto final no será el esperado.

5.- ¿Qué se entiende por verificación? ¿Y por validación?

Verificación se refiere al proceso de evaluación de un software para determinar si cumple con los requisitos establecidos. Durante la verificación, se revisan documentos, códigos, planes y especificaciones para garantizar que se sigan los estándares y procesos adecuados. Con ello podemos encontrar posibles errores en el desarrollo del software.

Validación evalúa un software durante o al final del proceso de desarrollo para determinar si satisface las necesidades del usuario y se ajusta a su propósito previsto. La validación implica pruebas dinámicas del sistema en un entorno simulado o real para verificar que el software se comporte según lo esperado y cumpla con los requisitos del cliente.

6.- Explica cómo funciona el modelo de desarrollo mediante creación de prototipos.

Un modelo prototipo o modelo de desarrollo evolutivo es utilizado principalmente en el desarrollo de software para ofrecer al usuario una visión previa de cómo será el programa o sistema. Se le dice de desarrollo evolutivo al modelo de prototipo porque evoluciona hasta convertirse en el producto final.

En un modelo de prototipos las características fundamentales son:

- Tiempo. El prototipo se desarrolla en menos tiempo para poder ser probado y testeado.
- Coste. La inversión en un modelo de prototipo es ajustada, lo que requiere un uso óptimo de los recursos.
- Conciso. El prototipo debe incluir los requisitos y características básicas de la aplicación para poder evaluar su funcionamiento y utilidad.
- Evolutivo. El prototipo evoluciona gracias a la interacción con los usuarios.
- Funcional. El prototipo es una aplicación que funciona.

El modelo de prototipos se centra en un diseño rápido que representa las características principales del programa que el usuario podrá ver o utilizar. De esta manera pueden

probarlo y dar su opinión sobre distintos aspectos como la usabilidad, la utilidad o el rendimiento, entre otras.

El prototipo se puede modificar cuando sea necesario y todos los resultados obtenidos de las presentaciones y pruebas se deben anotar para utilizar posteriormente como ayuda en el desarrollo del producto final.

Se destacan dos tipos de prototipos:

- Prototipos rápidos
 - El prototipo puede ser desarrollado usando otro lenguaje o herramientas diferentes al proyecto y finalmente el prototipo se desecha.
- Prototipos evolutivos
 - El prototipo está diseñado en el mismo lenguaje y herramientas del proyecto y se utiliza de base para desarrollar el proyecto..

7.- Explica cómo funciona el modelo espiral cuando se aplica al desarrollo orientado a objetos.

El modelo en espiral aplicado al desarrollo orientado a objetos implica establecer objetivos, analizar y diseñar clases y objetos clave, implementarlos en código, realizar pruebas y evaluar resultados. Basándose en esa evaluación, se planifica la siguiente iteración, permitiendo ajustar los objetos según los requisitos cambiantes del proyecto. Este enfoque iterativo y flexible se alinea con los principios del desarrollo orientado a objetos.

Jesús Reyes Espejo