**Tarea 2**

**Ciclo de vida del software (Relación 1)**

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

El ciclo de vida del software es una secuencia estructurada del proceso de planificación y creación y desarrollo de un sistema, un software.

Este se compone de diversas fases, necesarias para validar el desarrollo de nuestro software y así, garantizar que se cumplan los requisitos para la aplicación y verificación de los procedimientos de desarrollo, asegurándose de que los métodos son apropiados.

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

* Análisis:

Esta fase corresponde al proceso a través del cual se intenta descubrir qué es lo que realmente necesitamos para la creación y el desarrollo de nuestro software y cómo llegar a las características que el sistema debe poseer.

* Diseño:

En esta fase se estudian las posibles opciones que se pueden implementar para la creación y construcción de nuestro software, se decide la estructura general

que se lleva a cabo y se van proporcionando ideas.

Esta fase es tardía y compleja, el diseño debe desarrollarse de manera

iterativa, después de todo, la solución que se da inicialmente no suele ser la

más adecuada, por lo que se vuelve atrás, repitiendo el proceso, añadiendo

factores e ideas para mejorarlo.

* Codificación:

Esta fase es la más complicada y trabajosa, se tienen que elegir las

herramientas adecuadas y un entorno de desarrollo que facilite el trabajo a la

hora de programar y meter códigos, además del lenguaje de programación

apropiado para nuestro software.

Dichas decisiones dependerán de las ideas tomadas de la fase anterior, el

diseño, y el entorno en el que se va a codificar para que funcione

correctamente como se desea.

Al programar hay que intentar que el código no sea indescifrable, por ejemplo,

usando una estructura ordenada y limpia, identificando las variables

correctamente y el espacio necesario, y facilitando la interpretación visual del

código.

* Pruebas:

La fase de pruebas del ciclo de vida del software busca detectar los fallos cometidos en las etapas anteriores para corregirlos.

Lo ideal es hacerlo antes de que el usuario final se los encuentre.

Se dice que una prueba es un éxito si se detecta algún error.

Esta fase del ciclo de vida del software hay que repetirla tantas veces como sea

posible y necesaria, ya que la calidad y estabilidad final del software dependerá

de esta fase.

* Mantenimiento:

Esta es una de las fases más importantes del ciclo de vida de desarrollo del software.

El software ni se rompe ni se desgasta con el uso, su mantenimiento incluye:

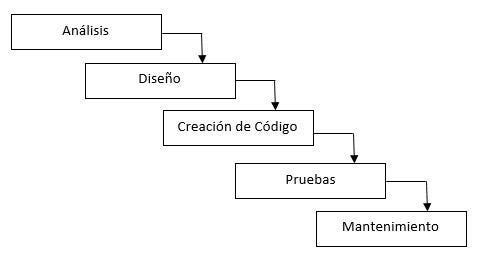
* + Eliminar los defectos detectados (mantenimiento correctivo).
  + Adaptarlo a nuevas necesidades (mantenimiento adaptativo).
  + Añadirle nuevas funcionalidades (mantenimiento perfectivo).

Cuanto mejor es el software, más tiempo hay que invertir en su mantenimiento. La principal razón es que se usará más, incluso de formas que no se habían previsto y, por ende, habrá más propuestas de mejoras.

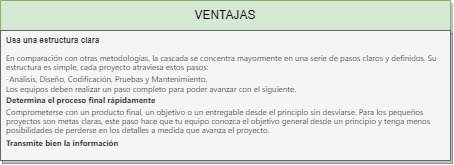
3.- Explica brevemente en qué consiste el modelo en cascada cuando hablamos de desarrollo de software.

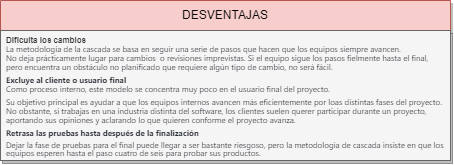
El método de cascada se basa en que los equipos sigan una secuencia de pasos y nunca avancen hasta que se haya completado la fase anterior. Esta metodología, en su forma tradicional, no deja prácticamente ningún lugar para cambios o revisiones imprevistos.

Esta estructura es apta para proyectos más pequeños que son fáciles de definir desde el inicio.



4.- Ventajas e inconvenientes del modelo en cascada.





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

La etapa de verificación se centra en tomar esa ejecución de la codificación y asegurarse de que cumple los requisitos que establecemos inicialmente en las fases anteriores, incluyendo la fase de análisis y la fase de diseño.

La validación se da si la etapa de verificación resulta exitosa, es decir, funciona como se pide y desea, sin errores, cumpliendo la función que pedimos en los requisitos.

Si funciona correctamente y estamos satisfechos con el proyecto, es válido, si no, no lo damos como válido y se repiten los procesos, verificando de nuevo.

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

El modelo de desarrollo mediante la creación 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.

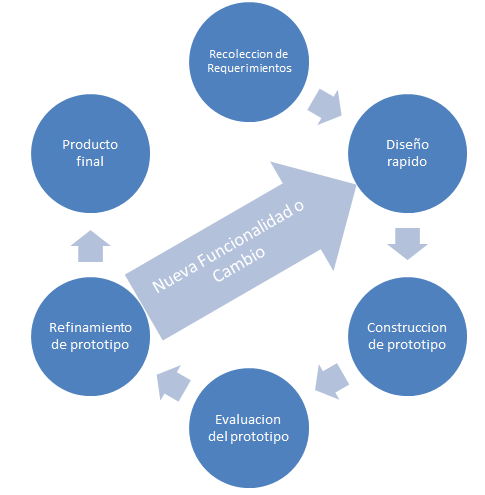
El prototipo se puede modificar cuando sea necesario, y todos los resultados obtenidos de las presentaciones y pruebas se deben anotar para utilizar posteriormente como ayuda en el desarrollo del producto final.

ETAPAS

* Requisitos de desarrollo: Se realiza un análisis para poder establecer cuáles son los requisitos del programa. Se trata de un diseño básico del prototipo donde se traza de forma inicial los requisitos necesarios para su desarrollo.
* Modelaje y desarrollo del código: En esta fase se construye el prototipo inicial según los requisitos establecidos, se debe priorizar el tiempo de desarrollo y hacer un uso óptimo de los recursos para reducir su coste.
* Evaluación: Una vez desarrollado el prototipo es necesario comprobar su funcionamiento, evaluando su funcionalidad y verificando que cumple realmente con los requisitos iniciales.
* Modificación: Tras evaluar el prototipo se corrigen los errores encontrados y se aplican las mejoras necesarias para que esté listo y ser probado por los usuarios.
* Documentación: Todo el diseño y desarrollo debe ser documentado para disponer de información precisa y clara del proceso.

Es muy importante el registro de cada paso o acción del desarrollo del prototipo, ya que es una guía útil a la hora de afrontar el diseño del producto final.

* Pruebas: El prototipo debe ser probado por los usuarios para poder recibir el *feedback* necesario y así evaluar su utilidad y rendimiento. Gracias a esto, se podrá desarrollar un *software* de mayor calidad que resuelva los problemas de los usuarios.



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

El desarrollo o modelo en espiral es un enfoque de desarrollo de software que puede ser considerado como una respuesta a los inconvenientes del desarrollo en cascada, describe el ciclo de vida de un software por medio de espirales, que se repiten hasta que se puede entregar el producto terminado.

Una característica clave del desarrollo en espiral es la minimización de los riesgos en el desarrollo de software, lo que podría resultar en un aumento de los costes totales, más esfuerzo y un lanzamiento retardado. Estos riesgos son contrarrestados por el enfoque incremental, haciendo primero prototipos, que luego pasan al menos una vez, por las fases de desarrollo de software.

Se caracteriza por las siguientes fases:

* Objetivo y determinación alternativa: Los objetivos se determinan conjuntamente con el cliente, se discuten posibles alternativas y se especifican las condiciones marco.
* Análisis y evaluación de riesgos: Se identifican y evalúan los riesgos potenciales. También se evalúan las alternativas existentes.

Los riesgos son registrados, evaluados y luego reducidos utilizando prototipos.

* Desarrollo y prueba: Los prototipos se amplían y se añaden funcionalidades.

El código real es escrito, probado y migrado a un entorno de prueba varias

veces hasta que el software pueda ser implementado en un entorno

productivo.

* Planificación del siguiente ciclo: El siguiente ciclo se planifica al final de cada etapa. Si se producen errores, se buscan soluciones.

