

Tarea 2 Matthew Farrell

Ciclo de vida del software (Relación 1)

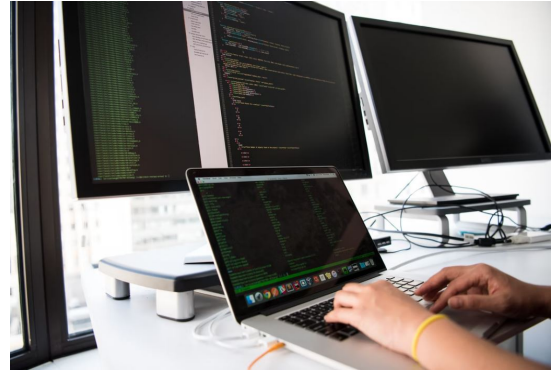
1.- Define "Ciclo de vida del software".

- Son todas las fases necesarias para empezar el desarrollo del software y garantizar que la aplicación cumpla los requisitos del cliente
- Las aplicaciones se han empezado a desarrollar de esta manera ya que anteriormente era muy caro arreglar errores que se encontraban más tarde en el proceso de desarrollo de la aplicación.
- Con las fases es más fácil detectar y planear qué fallos pueden haber en el desarrollo

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

- Las fases principales son
- - Análisis
 - En esta fase definimos los requisitos del cliente para cumplir el software a desarrollar
 - Diseño
 - En esta fase se descompone y organiza el sistema en elementos que se pueden ser desarrollados por separado
 - Se especifica la interrelación entre la funcionalidad de los elementos componentes

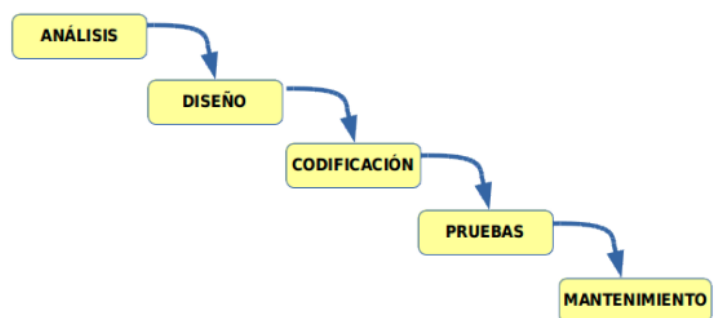
- Codificación
 - En esta fase se escribe el código fuente de la aplicación
 - Se utilizan lenguajes como C, C++, Java y Javascript
- Pruebas
 - En esta fase se consigue y se revisa que el programa funcione sin problemas
 - Somete el programa al máximo número de situaciones posibles
- Mantenimiento
 - Hay que realizar cambios ocasionales y revisiones, en esta fase se rehace parte del trabajo en las fases previas para hacer cambios correctivos, perfectivos, evolutivos o adaptativos



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

Es un modelo de desarrollo que identifica las fases principales del desarrollo de software

- Las fases se realizan en el orden indicado
- El resultado de una fase es la entrada de la siguiente fase
- Es un modelo rígido y no flexible que se adapta mal al cambio continuo de especificaciones que nos encontramos en el proceso del desarrollo de una aplicación



4.- Ventajas e inconvenientes del modelo en cascada.

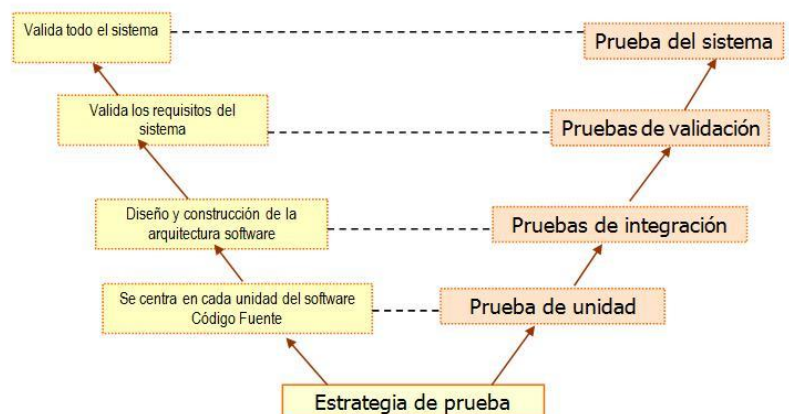
- ventajas:
 - Es un modelo muy fácil de entender
 - está orientado a documentos

- promueve una metodología de trabajo efectiva que consiste en definir antes que diseñar y diseñar antes que codificar
- Desventajas:
 - En la vida real es raro que siga una secuencia lineal, creando una mala implementación del modelo y que el proyecto tenga problemas de desarrollo
 - El proceso de creación tarda mucho tiempo ya que tiene que pasar por el proceso de prueba y hasta que el software no esté 100% completado no se sigue
 - Cualquier error en el diseño de prueba significa que se tiene que rediseñar y tendría que reprogramar código aumentando los costos
 - Una etapa del proyecto no se puede llevar a cabo si la etapa anterior no se haya hecho

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

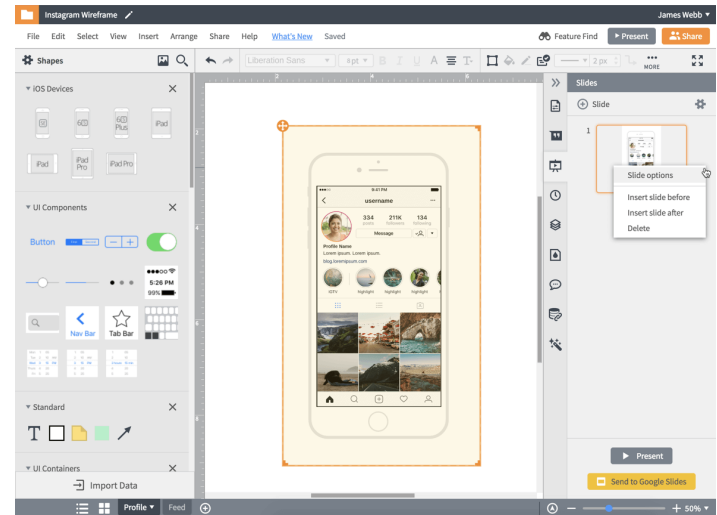
- Verificación: comprueba el diseño con las pruebas de integración y de unidad para comprobar si se está desarrollando correctamente la aplicación con las especificaciones previamente establecida

- Validación: comprueba y analiza el mantenimiento y la operación para comprobar si el producto se está haciendo bien y que satisfaga la intención funcional del programa



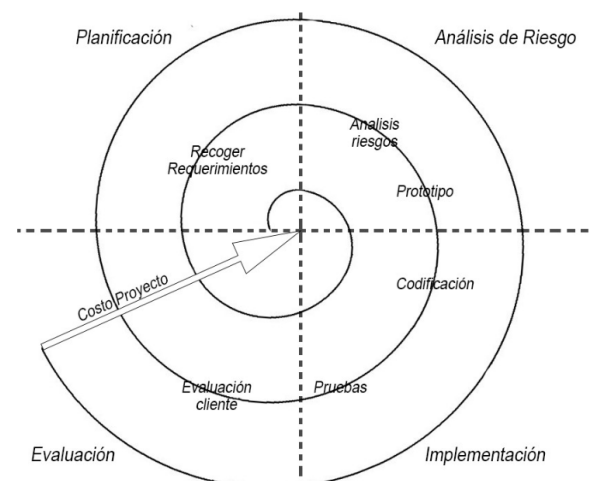
6.- Explica cómo funciona el modelo de desarrollo mediante creación de prototipos.

- En la primera fase se crea un prototipo durante la fase de análisis y es probado por el usuario o el cliente para refinar los requisitos del software a desarrollar
- En la segunda fase tenemos dos tipos de prototipos
 - Prototipos rápidos:
 - el prototipo puede estar desarrollado con otro lenguaje y otras herramientas
 - el prototipo se desecha cuando termina su trabajo
 - Prototipos evolutivos:
 - el prototipo está diseñado en el mismo lenguaje y herramientas del proyecto
 - el prototipo se usa como base de desarrollo del proyecto



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

- El modelo espiral tiene cuatro fases
 - Planificación: los requisitos se recogen durante la fase de planificación
- Análisis de riesgos: se emprende un proceso para identificar el riesgo y soluciones alternativas
- Implementación: en esta fase se desarrolla el software y al final sus respectivas pruebas



- Evaluación: esta fase permite el cliente evaluar el proyecto hasta la fecha, antes de que el proyecto pase a la siguiente espiral
- Este modelo es bueno ya que se aprende del trabajo previo y cuando vuelves a una fase que ya está previamente hecha se pasa con más fluidez y más rapidez