

## Tarea 2

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

- 1.- Define "Ciclo de vida del software".
- 2.- Nombra las fases principales del desarrollo de software y explica brevemente qué se hace en cada una de ellas.
- 3.- Explica brevemente en qué consiste el modelo en cascada cuando hablamos de desarrollo de software.
- 4.- Ventajas e inconvenientes del modelo en cascada.
- 5.- ¿Qué se entiende por verificación? ¿Y por validación?
- 6.- Explica cómo funciona el modelo de desarrollo mediante creación de prototipos.
- 7.- Explica cómo funciona el modelo espiral cuando se aplica al desarrollo orientado a objetos.

**1.** El ciclo de vida del desarrollo de software es la estructura que contiene los procesos, actividades y tareas relacionadas con el desarrollo y mantenimiento de un producto de software. Este proceso se divide en varias fases: análisis, diseño, codificación, pruebas y mantenimiento.



**2.** Las fases principales del desarrollo de software, como se dijo anteriormente, son análisis, diseño, codificación, pruebas y mantenimiento.

– **Análisis:** Se determina y define claramente las necesidades del cliente y se especifica los requisitos que debe cumplir el software a desarrollar. No obstante, se precisa que los requisitos sean completos y sin omisiones, concisos y entendibles por el cliente, entre otras especificaciones.

– **Diseño:** En esta fase ya se comienza a visualizar la solución con la ayuda de las anteriores fases. Se especifican las partes del proyecto para facilitar su desarrollo.

Se pueden encontrar:

- **Diseño arquitectónico:** Para definir la arquitectura.
- **Diseño detallado:** Para definir los problemas individuales de cada bloque.
- **Diseño de datos:** Para especificar el tipo de datos usados.
- **Diseño de interfaz:** Para plantear el diseño que verá el usuario.

En esta fase es común el uso de los mockup o maquetas, un diseño no funcional de la página/aplicación que ayudará al cliente a visualizar el producto final y facilitar la especificación de nuevos requisitos en una fase aún temprana del desarrollo.

– **Codificación:** Se elige el lenguaje de programación conveniente y se desarrolla el código fuente.

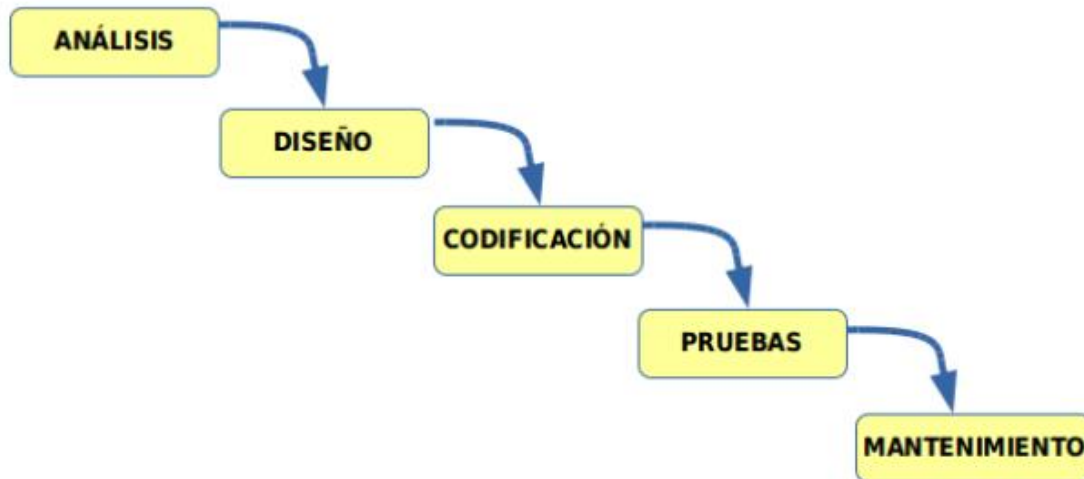
– **Pruebas:** El principal objetivo de las pruebas debe ser conseguir que el programa funcione incorrectamente y se detecten fallos (bugs). Para ello el equipo de pruebas puede utilizar métodos prácticos o pruebas automatizadas.

Se busca detectar el mayor número de fallos posible hasta asegurarnos de que la aplicación/página web sea funcional al 100%.

– **Mantenimiento:** Último paso, en el que se asegura la calidad del software una vez instalado por el usuario final. Para ello se especifican varios tipos de mantenimiento:

- **Preventivo:** Se corrigen defectos antes de que se produzcan.
- **Correctivo:** Corrige defectos
- **Perfectivo:** Mejora la funcionalidad.
- **Evolutivo:** Se añade funcionalidades nuevas.
- **Adaptativo:** Se adapta a nuevos entornos.

**3.** El modelo de cascada es un método de gestión de proyectos, en el que el proyecto se divide en distintas fases secuenciales, donde el equipo puede pasar a la siguiente fase sólo cuando se haya completado la anterior. Aplicado al desarrollo de software, y empleando los pasos previamente mencionados, un proyecto luciría así:



Este método se puede aplicar cuando hay una visión bastante clara de lo que debería ser el producto final, sin que los clientes tengan posibilidad de realizar cambios en este una vez haya comenzado y cuando la velocidad no es un factor determinante.

**4.** Por un lado tiene sus ventajas ya que es un modelo simple y fácil de usar, en un proceso bastante predecible y en el que las fases no se superponen. También debido a su rígida metodología es fácil de administrar.

Pero si el modelo en cascada se usa cada vez menos, es precisamente por su rigidez y su lentitud. Es un modelo poco dinámico, ya que cuando un equipo está trabajando en una fase del proyecto, los demás no pueden hacer prácticamente nada. Además de que no deja al cliente involucrarse en el proyecto por el trastorno que supondría un cambio a mitad de camino.

**5.** En el entorno de desarrollo de software, podemos decir que la verificación es el proceso de determinar la calidad del software. En una serie de procesos que implican la evaluación de la aplicación para determinar si cumple los requisitos y puede satisfacer a los clientes o usuarios finales. Estos procesos incluyen reuniones, inspecciones, revisiones y mucho más.

Por su parte, la validación consiste en comprobar los requisitos del cliente a través de la funcionalidad del software. Este proceso se lleva a cabo una vez completadas las verificaciones al final del desarrollo. Existen varios métodos de validación, como las

pruebas de caja negra (pruebas funcionales, enfocadas en la funcionalidad de la página/aplicación) o las pruebas de caja blanca (pruebas no funcionales, centradas en el diseño o la arquitectura).

**6.** El modelo de prototipos o modelo evolutivo se utiliza para dar al usuario una vista preliminar de parte del software. En este modelo el proceso llega hasta la fase de análisis, donde se crea en prototipo y es probado por el cliente para buscar disidencias e ir refinando lo que este busca. Esto se repite las veces que sea necesario hasta que el cliente esté satisfecho con el producto resultante.



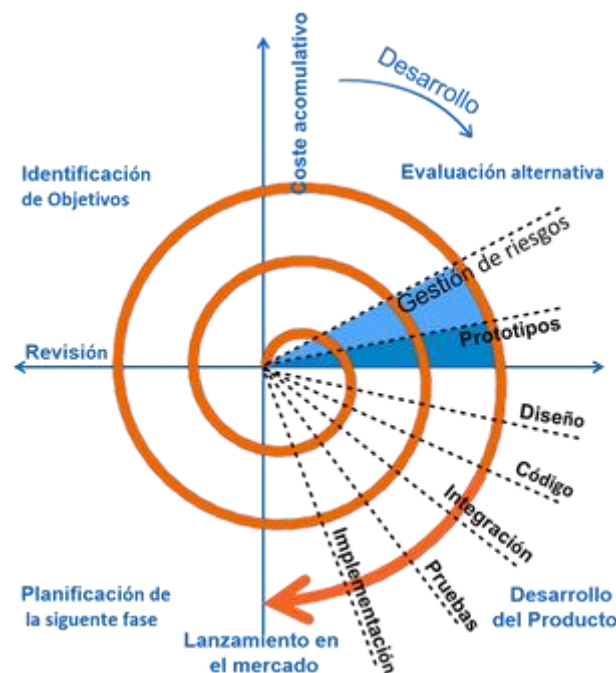
Podemos destacar dos tipos de prototipos:

- **Prototipo rápido:** En este modelo se da prioridad al desarrollo rápido pues el objetivo es conseguir una rápida evaluación del modelo. Finalmente se desecha.
- **Prototipo evolutivo:** En este modelo, además de ofrecer una visión previa de cómo será el producto, se pretende abarcar más aspectos y funciones del mismo (aunque sigue siendo un número limitado), ya que servirá como base del proyecto real.

**7.** El modelo en espiral describe el ciclo de vida de un software por medio de espirales, que se repiten hasta que se puede entregar el producto terminado. El producto se trabaja continuamente y las mejoras a menudo tienen lugar en pasos muy pequeños.

Se caracteriza por los siguientes ciclos:

- **Objetivo y determinación alternativa:** Los objetivos se determinan conjuntamente con el cliente.
- **Análisis y evaluación de riesgos:** Se identifican y evalúan los riesgos potenciales. También se evalúan las alternativas existentes.
- **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, y si una alternativa es una mejor solución, se prefiere en el siguiente ciclo.



Al esbozar el ciclo de vida de un software por medio de espirales, este modelo evita las desventajas de otros modelos (como el de cascada, por ejemplo) y enfatiza las ventajas. Al centrarse en la minimización del riesgo, este modelo tiene un componente financiero que puede ser relevante para los responsables de la toma de decisiones.