**Tarea 2**

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

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

El ciclo de vida del desarrollo del software está formado por las fases necesarias para validar el desarrollo del software y así garantizar que este cumpla los requisitos para la aplicación y verificación de los procedimientos de desarrollo, asegurándose de que los métodos usados son apropiados.

Es muy costoso rectificar los posibles errores que se detectan tarde en la fase de implementación. Utilizando metodologías apropiadas, se podría detectar a tiempo para que los programadores puedan centrarse en la calidad del software, cumpliendo los plazos y los costes asociados.



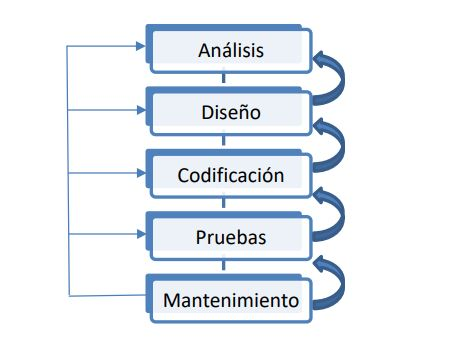
<https://intelequia.com/es/blog/post/ciclo-de-vida-del-software-todo-lo-que-necesitas-saber>

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

* Planificación: En esta parte del proceso se determina el ámbito del proyecto, el análisis de los riesgos, el estudio de viabilidad, la duración, la estimación del costo y la asignación de recursos a cada fase. Una planificación bien realizada, permite establecer las bases para un desarrollo orientado al éxito.
* Análisis: Aquí se descubre todo lo que se espera del software. Por ello, se realiza una exhaustiva investigación para llegar a una comprensión precisa de los requerimientos o características que debe poseer el programa. De esta forma, se elige o crea la arquitectura o estructura en donde operará.
* Diseño: Se estudian las posibles alternativas, algo que requiere de mucha atención. Se consideran todos los aspectos de la implementación tecnológica, como el hardware, el lenguaje y la red. Todo esto sirve para presentar algunos modelos de proceso, guiones gráficos, prototipos e, incluso, una simulación del diseño.
* Codificación: Se escribe el código fuente de cada componente.

Pueden utilizarse distintos lenguajes informáticos como los lenguajes de programación: C, C++, Java, Javascript, ... o lenguajes de otro tipo: HTML, XML, JSON, ...

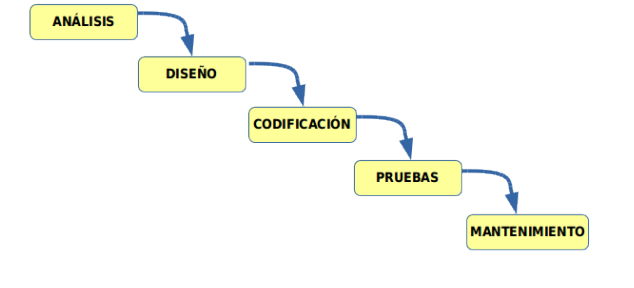
* Pruebas: Esta fase es muy impoirtante porque, antes de llegar al usuario, hay que comprobar que el programa ejecute las tareas especificadas. Además, sirve para detectar fallos y analizar el rendimiento del software. Aunque el desarrollador hace sus propias pruebas, se recomienda que también sean realizadas por alguien más.
* Mantenimiento: Se cree que al poner en práctica el software se acaba el trabajo, pero esto no es así. Es esencial mantener, optimizar y mejorar el programa para eliminar los errores detectados, adaptar nuevas necesidades o añadir nuevas funcionalidades. Esto significa que se requiere hacer actualizaciones frecuentes.



<https://rootstack.com/es/blog/cuales-son-las-fases-en-el-desarrollo-de-software>

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

El desarrollo en cascada es un procedimiento lineal que se caracteriza por dividir los procesos de desarrollo en sucesivas fases de proyecto. Al contrario que en los modelos iterativos, cada una de estas fases se ejecuta tan solo una vez.



<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/>

**4.- Ventajas e inconvenientes del modelo en cascada.**

|  |  |
| --- | --- |
| **VENTAJAS** | **INCONVENIENTES** |
| Usa una estructura clara | Dificulta los cambios |
| Determina el objetivo final rápidamente | Excluye al cliente o al usuario final |
| Transmite bien la información | Retrasa las pruebas hasta después de la  finalización |

<https://www.lucidchart.com/blog/es/pros-y-contras-de-la-metodologia-de-cascada>

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

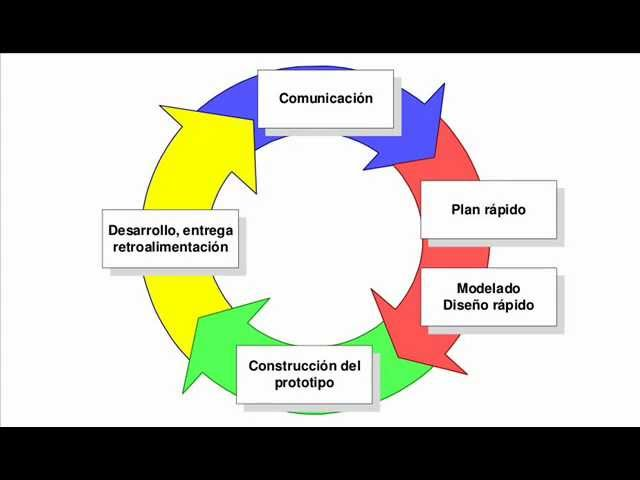
Verificar consiste en comprobar que todo el trabajo realizado está hecho correctamente o que algo funciona de forma correcta.

Una validación se trata de aprobar que un trabajo se ha realizado correctamente o que algo funciona como debe hacerlo.

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

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.

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.



[https://www.hostingplus.com.es/blog/modelo-de-prototipos-que-es-y-cuales-son-sus-etapas/#:~:text=Cómo%20funciona,o%20el%20rendimiento%2C%20entre%20otras](https://www.hostingplus.com.es/blog/modelo-de-prototipos-que-es-y-cuales-son-sus-etapas/" \l ":~:text=Cómo funciona,o el rendimiento%2C entre otras).

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

En la actividad de ingeniería, el modelo en espiral corresponde a las fases de los modelos clásicos: análisis., diseño, codificación…

Es un enfoque de desarrollo de software que puede ser considerado como una respuesta a los inconvenientes del desarrollo en cascada. 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.

En cuanto a su funcionamiento, vemos que se divide en varios ciclos:

Objetivo y determinación alternativa: Los objetivos se determinan conjuntamente con el cliente. Al mismo tiempo, se discuten posibles alternativas y se especifican las condiciones marco.

Análisis y evaluación de riesgos: Se identifican y evalúan los riesgos potenciales y las alternativas existentes. Los riesgos son registrados, evaluados y luego reducidos utilizando prototipos, simulaciones y software de análisis. En este ciclo, existen varios prototipos como plantillas de diseño o componentes funcionales.

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.

<https://es.ryte.com/wiki/Modelo_en_Espiral>