

Entornos de Desarrollo

01 Desarrollo de Software

José Luis González Sánchez



Contenidos

¿Qué voy a aprender?

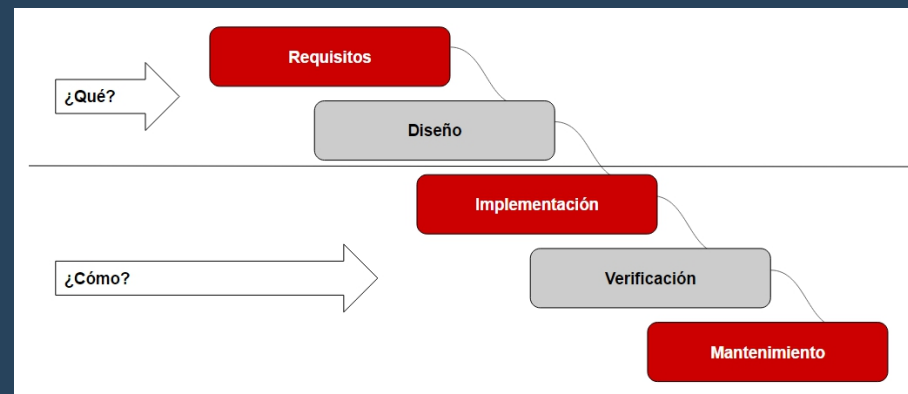


Contenidos

1. Desarrollo de Software
2. Ciclos de Vida en el desarrollo Software
3. Metodologías Ágiles

Desarrollo de Software

- Entendemos por Desarrollo de Software todo el proceso que ocurre desde que se concibe una idea hasta que un programa está implementado en el ordenador y funcionando.
- **Ingeniera del software:** se define como la ciencia y el arte de especificar, diseñar y desarrollar programas, documentación y procedimientos operativos.
- El proceso de desarrollo, que en un principio puede parecer una tarea simple, consta de una serie de pasos de obligado cumplimiento, pues sólo así podremos garantizar que los programas creados son eficientes, fiables, seguros y responden a las necesidades de los usuarios finales (aquellos que van a utilizar el programa).
- Como veremos con más detenimiento a lo largo de la unidad, el desarrollo de software es un proceso que conlleva una serie de pasos. Genéricamente, estos pasos son los siguientes:

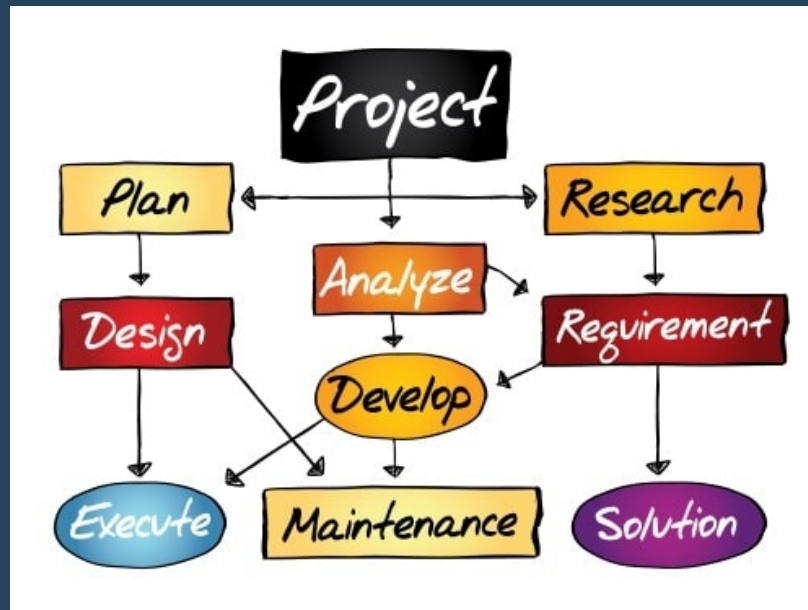


Desarrollo de Software - Fases

- **Análisis del problema:** Consiste en ver y comprender que tarea se quiere resolver. Es imprescindible partir de una especificación de requisitos lo más exacta y detallada posible. El proceso de comprensión y simplificación del mundo real se llama análisis del problema y lo que se obtiene tras el análisis es el modelo.
- **Diseño de una solución:** Consiste en cómo se va a resolver el problema, suele consistir en dividir el problema principal en problemas más sencillos cuya combinación resuelve la tarea final.
- **Especificación de módulos:** Consiste en, para cada subproblema ideado, diseñar una solución que lo resuelva lo más eficiente posible, esto se realiza con un algoritmo.
- **Codificación:** Una vez definidos los algoritmos, los traducimos al lenguaje de programación que tengamos que usar. **Un lenguaje de programación** es un conjunto de símbolos y reglas sintácticas especialmente diseñado para transmitir ordenes al ordenador (C, Java, Pascal...).
- **Pruebas:** Servirán para corregir posibles errores, tendremos dos tipos de errores:
 - **Sintácticos:** Producidos por un mal uso del lenguaje.
 - **Semánticos:** Lo que está equivocado es la solución que yo he ideado.
- **Mantenimiento:** Hay tres tipos:
 - **Correctivo:** Sirve para corregir posibles errores o fallos del programa.
 - **Perfectivo:** Se usa para perfeccionarlo.
 - **Adaptativo:** Sirve para adaptarlo a nuevas situaciones.

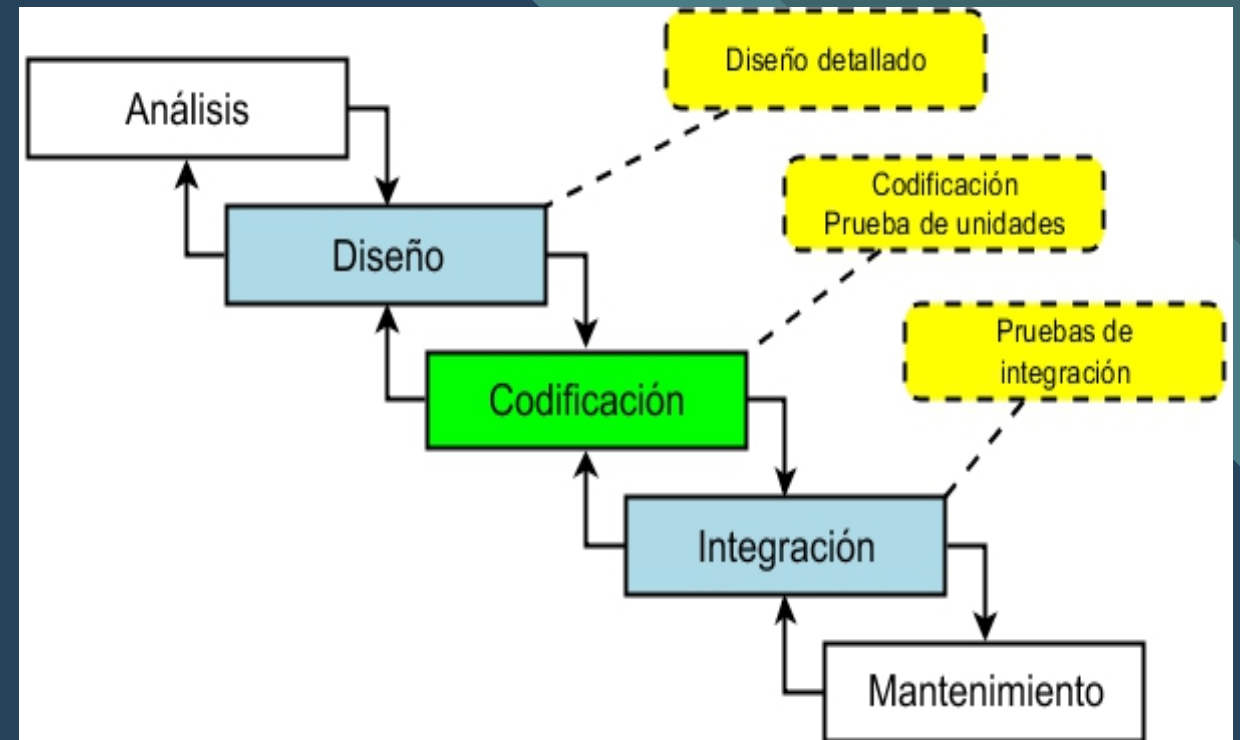
Ciclos de Vida en el Desarrollo Software

- La serie de pasos a seguir para desarrollar un programa es lo que se conoce como Ciclo de Vida del Software.
- Cada etapa vendrá explicada con más detalle en el punto de la presente unidad dedicado a las fases del desarrollo y ejecución del software.
- Diversos autores han planteado distintos modelos de ciclos de vida, pero los más conocidos y utilizados son los que veremos.



Modelo en Cascada (con realimentación)

- Sin realimentación: es el modelo de vida clásico del software. Es prácticamente imposible que se pueda utilizar, ya que requiere conocer de antemano todos los requisitos del sistema. Sólo es aplicable a pequeños desarrollos, ya que las etapas pasan de una a otra sin retorno posible. (se presupone que no habrá errores ni variaciones del software).
- Con realimentación: es uno de los modelos más utilizados. Proviene del modelo anterior, pero se introduce una realimentación entre etapas, de forma que podamos volver atrás en cualquier momento para corregir, modificar o depurar algún aspecto. No obstante, si se prevén muchos cambios durante el desarrollo no es el modelo más idóneo. Es el modelo perfecto si el proyecto es rígido (pocos cambios, poco evolutivo) y los requisitos están claros.

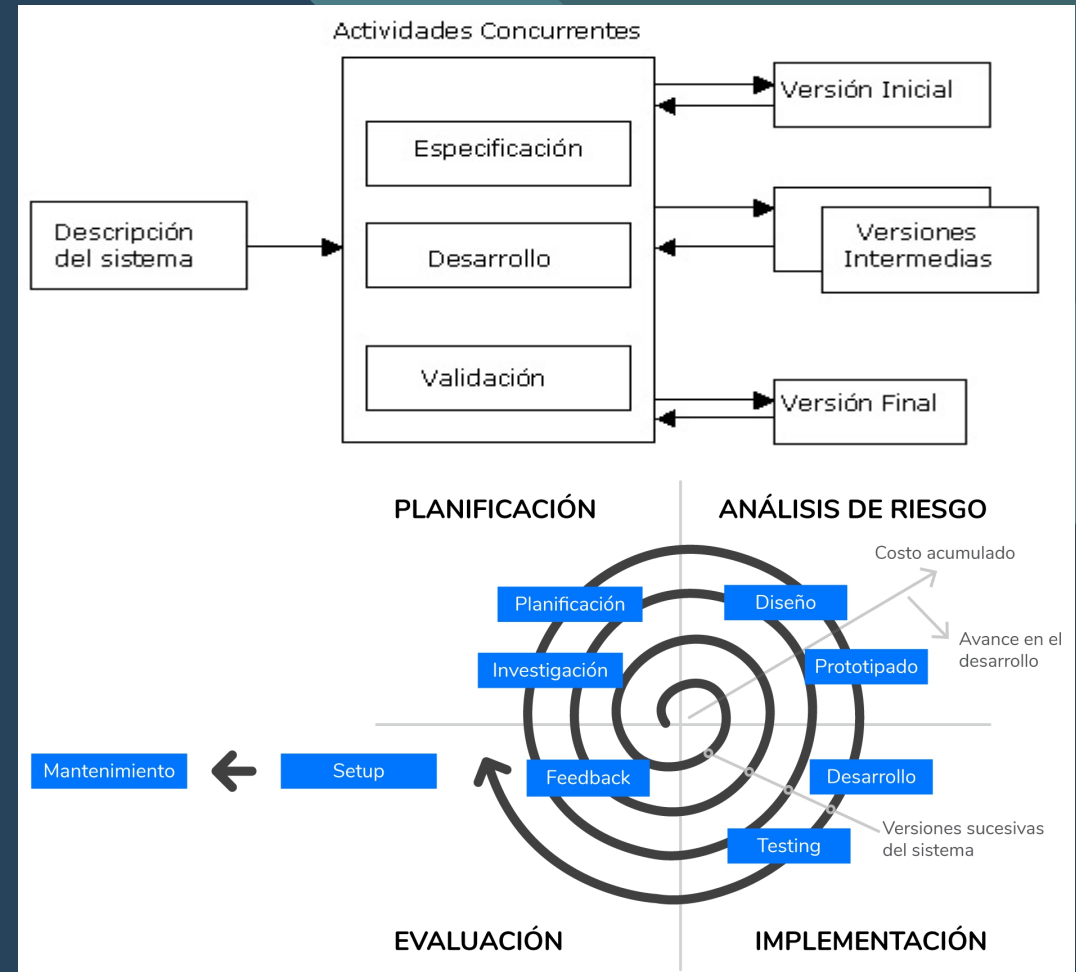


Modelo Evolutivo

- **Modelos Evolutivos**

Son más modernos que los anteriores. Tienen en cuenta la naturaleza cambiante y evolutiva del software. Distinguimos dos variantes:

- **Modelo Iterativo Incremental.** Está basado en el modelo en cascada con realimentación, donde las fases se repiten y refinan, y van propagando su mejora a las fases siguientes.
- **Modelo en Espiral.** Es una combinación del modelo anterior con el modelo en cascada. En él, el software se va construyendo repetidamente en forma de versiones que son cada vez mejores, debido a que incrementan la funcionalidad en cada versión. Es un modelo bastante complejo.
- La idea detrás de este modelo es el desarrollo de una implantación del sistema inicial, exponerla a los comentarios del usuario, refinarla en N versiones hasta que se desarrolle el sistema adecuado. Una ventaja de este modelo es que se obtiene una rápida **realimentación del usuario**, ya que las actividades de especificación, desarrollo y pruebas se ejecutan en cada iteración.



Ventajas e Inconvenientes

- **VENTAJAS DEL EVOLUTIVO RESPECTO AL SECUENCIAL.**

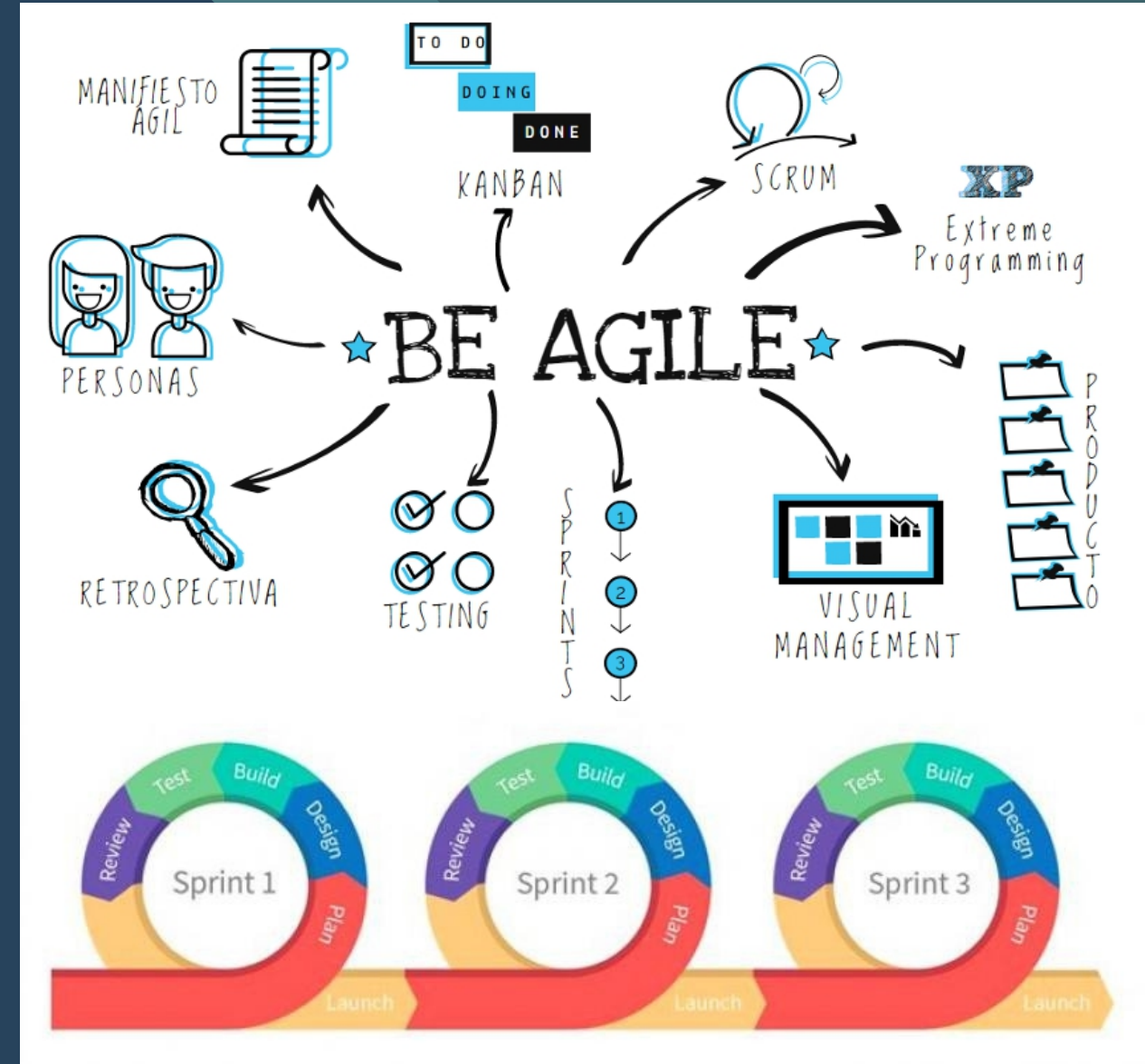
- La especificación puede desarrollarse de **forma creciente**.
- Los usuarios y desarrolladores logran un **mejor entendimiento** del sistema. Esto se refleja en una mejora de la calidad del software.
- Es **más efectivo** que el modelo de cascada, ya que cumple con las necesidades inmediatas del cliente.

- **VENTAJAS DEL SECUENCIAL RESPECTO AL EVOLUTIVO.**

- Los administradores **necesitan entregas para medir el progreso**. Si el sistema se necesita desarrollar rápido, no es efectivo producir documentos que reflejen cada versión del sistema por lo que un **diseño más clásico es más eficiente**.
- Sistemas mejor estructurados: Los **cambios continuos pueden ser perjudiciales para la estructura del software** haciendo costoso el mantenimiento por lo que el secuencial en cascada, más estructurado, es más eficiente para este caso.
- **Para el evolutivo se requieren técnicas y herramientas**. Para el rápido desarrollo se necesitan herramientas que pueden ser incompatibles con otras o que poca gente sabe utilizar. Por lo tanto, cuando disponemos de un equipo suficientemente experto y numeroso para encargarse de las tareas secuenciales es preferible el secuencial

Metodologías Ágiles

- En el cambiante mundo del **desarrollo de software**, los requisitos aplicados a un proyecto parecen incrementarse en el tiempo, mientras se reduce el plazo que se asigna para cada trabajo.
- El software de **desarrollo Agile se refiere a un grupo de metodologías aplicadas en la creación de software que basa su desarrollo en un ciclo iterativo**, en el que las necesidades y soluciones evolucionan a través de la colaboración entre los diferentes equipos involucrados en el proyecto.
- Los métodos Agile por norma general promueven una gestión de proyectos disciplinada que fomenta la constante inspección del código y la adaptación de éste, un sistema organizado que permite y facilita el trabajo en equipo, la auto-organización y favorece el rendimiento del tiempo de desarrollo.



A light blue circular icon containing a white opening quotation mark.

"El problema de los programadores es que nunca se sabe lo que están haciendo hasta que es demasiado tarde".

- Seymour Cray

A light blue circular icon containing a white closing quotation mark.

Recursos

- Twitter: <https://twitter.com/joseluisgonsan>
- GitHub: <https://github.com/joseluisgs>
- Web: <https://joseluisgs.github.io>
- Discord: <https://discord.gg/WKKvSJCS>
- Aula Virtual: <https://aulavirtual33.educa.madrid.org/ies.luisvives.leganes/course/view.php?id=246>



Gracias

José Luis González Sánchez

