

APUNTES

Tipos de software:

De sistema - interactúa directamente con el hardware (de bajo nivel, muy cerca del hardware)

De aplicación - programas que están más lejos que el microprocesador y se valen del SO

De desarrollo - programas para hacer programas (VisualStudioCode, Eclipse, etc)

firmware: es software de sistema pero muy cercano al hardware (ej: cualquier tipo de código de programa que se ejecuta en un microcontrolador (perifericos raton teclado etc..))

drivers- programa entre el SO y el firmware del hardware. Son los controladores de dispositivos y mandan instrucciones básicas al hardware del firmware.

RELACION HARDWARE Y SOFTWARE

CPU (Central Processing Unit) - cerebro del ordenador lee y ejecuta instrucciones que están almacenadas en la ram.

Memoria RAM (Memoria principal)

Código fuente- es el código que ha desarrollado el programador.

código objeto- archivo binario no ejecutable, se basa en el código fuente y no es legible
código ejecutable

Byte code- código objeto que genera JAVA.

CICLO DE VIDA DE SOFTWARE

Ingeniería de software- disciplina que aplica principios y metodologías para el mantenimiento y desarrollo de sistemas software.

DESARROLLO DE SOFTWARE

análisis- Plantear requisitos

diseño - Se descompone y organiza el sistema en elementos componentes que pueden ser desarrollados por separado. (**mock up**- es un diseño estático de una página web o aplicación que presenta muchos de sus elementos de diseño finales pero no es funcional.)

codificación - Realización del código fuente (en lenguajes compilados, interpretados, de marca, etc)

pruebas - conseguir que el programa funcione mal, para buscar los fallos y arreglarlos.

mantenimiento -

Correctivo: Corregir los defectos una vez ocurrido el fallo.

Preventivo: Cuando se arregla el problema con antelación para prevenir fallos.

Perfectivo: Se mejoran las funcionalidades.

Evolutivo: Se añaden funcionalidades.

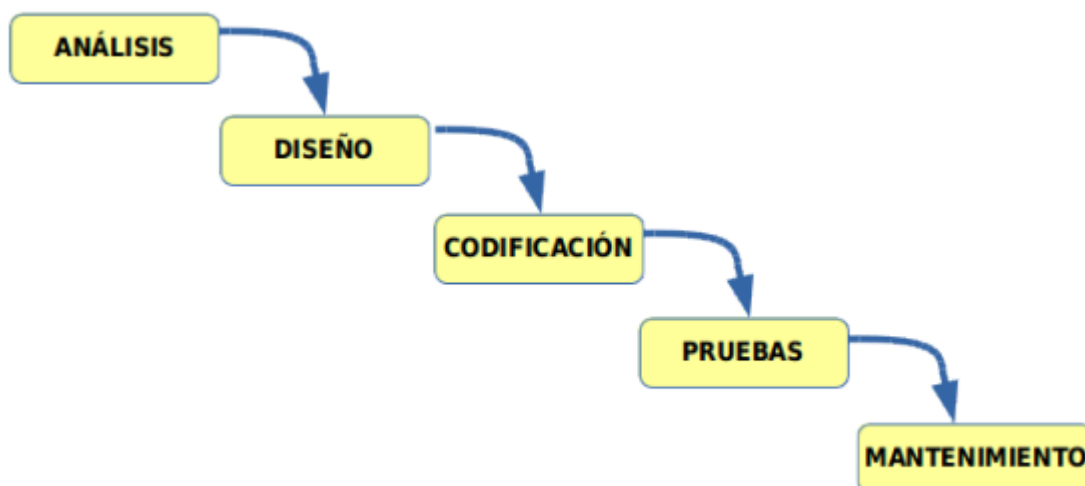
Adaptativo: Se adapta a nuevos entornos

Resultado tras cada fase

Ingeniería de sistemas: Especificación del sistema •
ANÁLISIS: Especificación de requisitos del software •
DISEÑO arquitectónico: Documento de arquitectura del software •
DISEÑO detallado: Especificación de módulos y funciones •
CODIFICACIÓN: Código fuente •
PRUEBAS de unidades: Módulos utilizables •
PRUEBAS de integración: Sistema utilizable •
PRUEBAS del sistema: Sistema aceptado •
Documentación: Documentación técnica y de usuario •
MANTENIMIENTO: Informes de errores y control de cambios

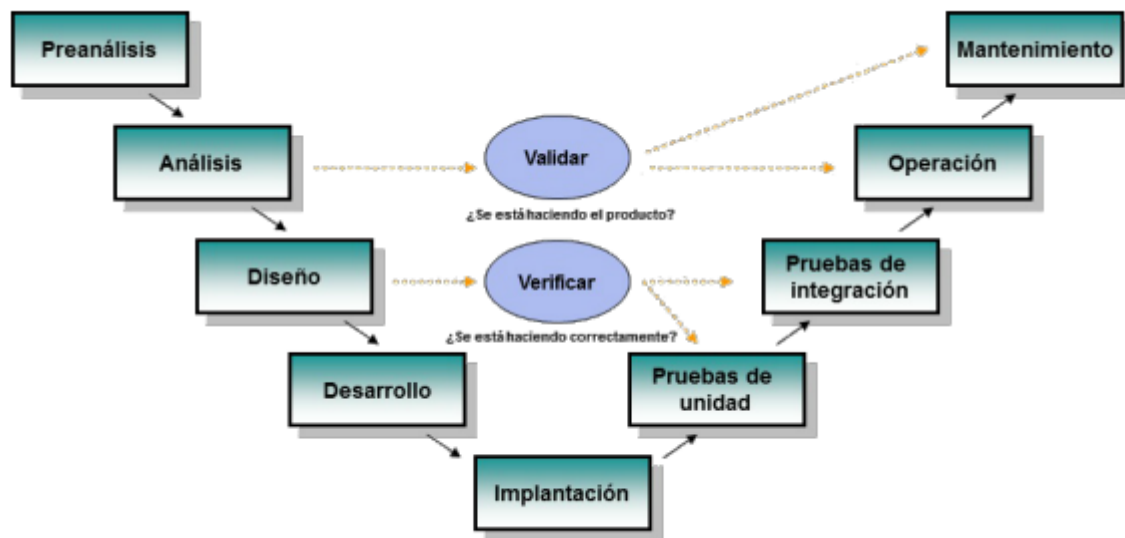
MODELOS DE DESARROLLO SOFTWARE

MODELO EN CASCADA



inconvenientes de este modelo: si el cliente no le gusta la versión final, se debe de empezar de nuevo el proceso, perdiendo tiempo y por tanto, dinero.

MODELO EN V



METODOLOGÍAS ÁGILES

Son métodos de ingeniería del software basados en el desarrollo iterativo, es decir, yendo poco a poco.

El trabajo se organiza en pequeños grupos multidisciplinarios (un equipo debe de ser autónomo, es decir, sus componentes deben de tener conocimientos en todos los aspectos).

Las metodologías más conocidas son:

- Kanban
- Scrum
- XP (eXtreme Programming)