

Tipos de software:

Software: conjunto de componentes lógicos los cuales pueden ser modificados y son necesarios para realizar tareas en específico.

·**De sistema:** (sistema operativo, drivers, controladores) interactúa directamente con el hardware es el software que interactúa con el hardware, es de bajo nivel

·**De aplicación:** (suite informática, navegador, edición de imagen) conjunto de programas que el usuario usa más habitualmente es el más usado

·**De desarrollo:** (editores, compiladores, intérpretes) son programas los cuales permiten crear y desarrollar varias aplicaciones y webs

·**Firmware:** es un software el cual es el encargado de controlar todo el hardware de un dispositivo, es el software más cercano al hardware.

Ejemplo: cualquier tipo de aparato con minicontrolador.

·**Driver:** entre software y hardware, permite comunicar con firmware/hardware y recibe órdenes del sistema operativo.

Jerarquía de memorias.

Relación Hardware-Software

·**Disco Duro:** es la memoria secundaria y almacena de forma permanente los archivos tanto ejecutables como datos.

·**Memoria ram:** es la memoria principal y es una memoria volátil la cual almacena de forma temporal en binario archivos ejecutables y los archivos de datos necesarios.

·**CPU:** lee y ejecuta instrucciones de la RAM así como todos los datos necesarios, además de tener memoria caché.

·**E/S:** la entrada recoge información del exterior y la lleva al interior, la salida lleva información del exterior y la lleva al exterior.

·**Código fuente:** es el código del programa entendible por un ser humano para hacer un programa.

·**Código objeto:** es un archivo binario, se genera a partir del código fuente y no se puede ejecutar.

·**Código ejecutable:** es un archivo el cual si se ejecuta en el sistema operativo se completa su tarea

Solo para Lenguaje compilado

- **Lenguaje compilado:** simplifica el lenguaje para que varios dispositivos diferentes

- **Lenguajes interpretados:** no usa compilador, usan scripts.

- **Script:** son líneas de comando que se ejecutan de manera ordenada.

Ingeniería del software: Disciplina que estudia los principios y metodologías para el desarrollo y mantenimiento de sistemas software.

Desarrollo de software:

- **Análisis:** analizar requisitos

- **Diseño:** implementa idea, donde se define la arquitectura

- **Codificación:** programar código

- **Pruebas:** se comprueba el interacciones con el programa

- **Mantenimiento:** se implementan parches y actualizaciones

Análisis: analiza y define las necesidades del cliente y y especifica los requisitos que debe cumplir el software.

Especificaciones de requisitos:

- Ser completa y sin omisiones

- Ser concisa y sin trivialidades

- Evitar ambigüedades. Usar lenguaje formal

- Evitar detalles de diseño e implementación

- Ser entendible por el cliente

- Separar requisitos funcionales y no funcionales

- Dividir y jerarquizar el modelo: dividir en partes el diseño planeado

- Fijar criterios de validación: poner límites donde no se puede sobrepasar el diseño específico.

Diseño:

- Descompone y organiza el sistema en elementos componentes que pueden llegar a ser desarrollado por separado.

- Especifica la interrelación y funcionalidad de los elementos componentes.

Actividades habituales:

- **Diseño arquitectónico:** Definir como solucionar el programa

- **Diseño detallado:** definir los problemas por bloques

- **Diseño de datos:** especifica los datos usado en la aplicacion

- **Diseño de interfaz de usuario:** es lo que el usuario verá y interactuara

Codificación:

- Se escribe el código fuente de cada componente.
- Pueden utilizarse distintos lenguajes informáticos:
- **Lenguajes de programación:** C, C++, Java, Javascript, ...
- **Lenguajes de otro tipo:** HTML, XML, ...

Pruebas:

- Su objetivo es conseguir información en el funcionamiento del programa.
- Se debe de poner al programa en varias situaciones.

Mantenimiento:

- Se realizan cambios a lo largo de cambios, para ello hay que rehacer parte del trabajo realizado en las fases previas.

Tipos de mantenimiento:

- **Correctivo:** corrige defectos
- **Perfectivo:** mejorar funcionalidad
- **Evolutivo:** añade funcionalidades nuevas
- **Adaptativo:** adaptarse a nuevos entornos.
- **Preventivo:** se corrigen defectos antes de que se produzcan altercados.

Mock Up

Resultado tras cada fase:

- **Ingeniería de sistemas:** especificación de sistema
- **Análisis:** especificación de requisitos del software
- **Diseño arquitectónico:** documento de arquitectura del software
- **Diseño detallado:** especificación de módulos y funciones

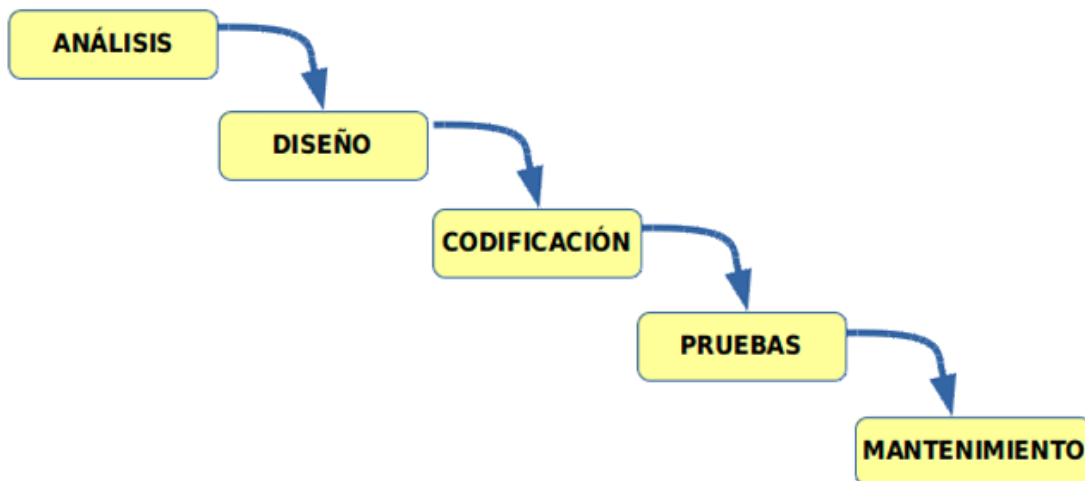
CODIFICACIÓN: código fuente

- **Pruebas de unidades:** módulos utilizables
- **Pruebas de integración:** sistemas utilizable
- **Pruebas de sistema:** sistema aceptado
- **Documentación:** documentación técnica de y de usuario
- **Mantenimiento:** informes de errores y control de cambios

Modelos del desarrollo del software

Modelos clásicos:

- Cascada
- Modelo en V



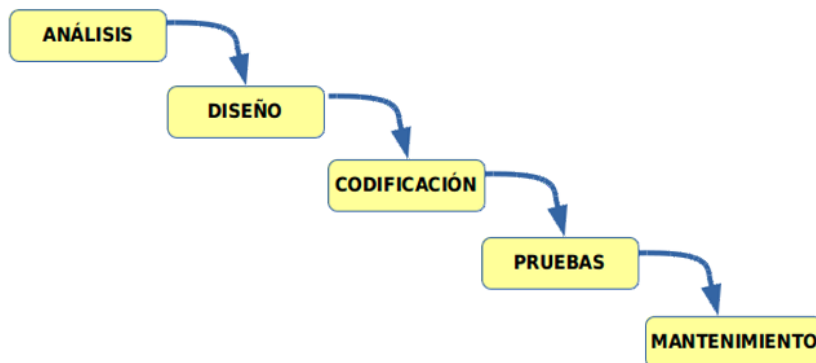
Modelo de construcción de prototipos

Modelo evolutivo o incrementales

- **Modelo en espiral** (iterativos: repetición de operación hasta que se cumpla la condición)
- **Metodologías ágiles** (adaptativos: diseñado para adaptarse y cambiar de forma sin intervención del programador)

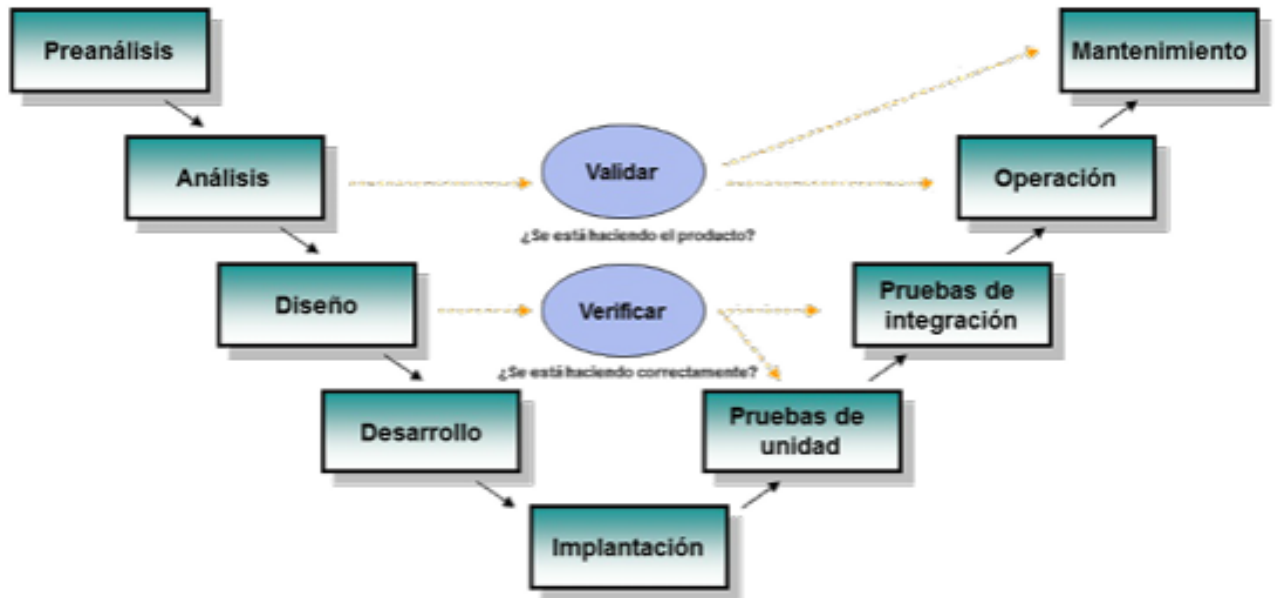
Modelo de cascada:

- El modelo más antiguo
- Identifica las fases principales del desarrollo software
- Cuando se termina una fase se comienza la siguiente
- Muy rígido y de mala adaptación
- Tiene variantes con mayor y menor cantidad de pasos



Modelo en V

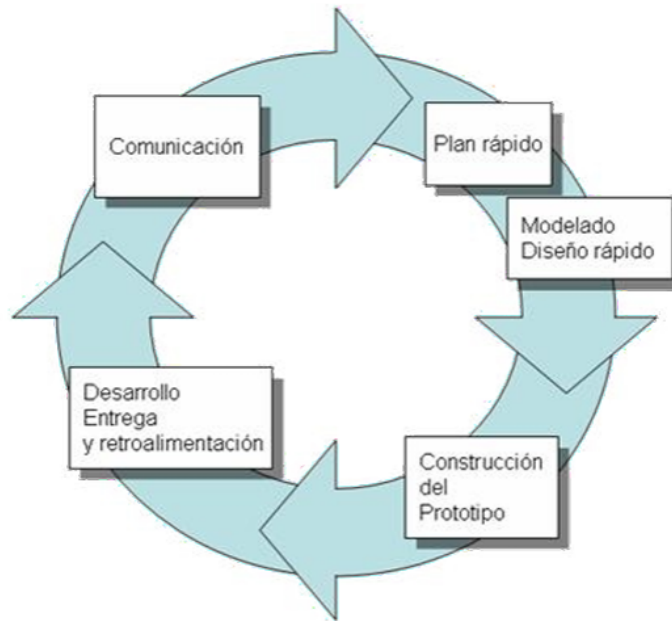
- Parecido al modelo en cascada
- Jerarquizada en distintos niveles
- Niveles superiores indican mayor abstracción
- Niveles inferiores indican mayor detalles
- El resultado de una fase es el inicio de la siguiente
- Existen diferentes variantes con mayores y menor cantidades de actividades



Prototipos

A Menudo los requisitos no están especificados claramente:

- por no existir experiencia previa
- por omisión o falta de concreción del usuario/cliente



Proceso:

·En la fase de análisis se crea un prototipo el cual es aprobado por el usuario/cliente para refinar el software.

·Se repite el proceso anterior todo lo que sea necesario.

Tipos de prototipos:

·Prototipos rápidos:

- Puede estar desarrollado usando lenguaje y/o herramientas
- Cuando se termina de usar se desecha

·Prototipos evolutivos

- Está diseñado con el mismo lenguaje y herramientas del proyecto
- El prototipo se usa como base para desarrollar el proyecto

Modelo en espiral