Tipos de software

- **Software de Sistema**→ Interactúa directamente con el hardware.
- **Software de Aplicaciones**→ Son los Programas que nos va a permitir realizar tareas.
- **Software de Desarrollo**→ Aplicaciones o programas que nos permiten la creación de otros programas.

Relación hardware-software

- **Disco Duro**→ Almacena de forma permanente los archivos.
- **Memoria RAM**→ Almacena de forma temporal el código de los archivos ejecutables y de datos.
- **CPU**→ Lee y ejecuta las instrucciones que se almacenan en la memoria RAM.
- **E/S**→ Recoge los datos desde la entrada y los muestra como resultado.

Código fuente objeto y ejecutable

- **Código fuente**→ Es el código entendible por un ser humano para hacer un programa nos permite modificar el programa de forma sencilla.
- **Código objeto**→ Es aquel archivo binario y el código se genera a través del código fuente pero no se puede ejecutar.
- **Código ejecutable**→ Es aquel archivo que únicamente se puede ejecutar.

El compilador se encarga de hacer funcionar el programa independientemente de su procesador (intel amd o mac)

Ciclo de vida del Software

Ingeniería del Software

Disciplina encargada de estudiar los principios y metodologías para desarrollar y mantener un software

Algunos consideran que "*desarrollo de software*" es un termino mas apropiado que "*ingeniería de softwar*e" ya que este último implica niveles de rigor y pruebas de procesos que no se apropian para todos los tipos de desarrollo de software

Fases

- **Análisis**→ Analizar y definir los requisitos
- **Diseños**→ Ver las ideas y cómo la implementamos
- **Codificación**→ Programar el código
- **Pruebas**→ Se realizan test para verificar que todo esté correcto y según lo previsto
- **Documentación/Mantenimiento**→ Actualizaciones , parches , solución de los fallos que surjan , algo que se pueda mejorar o alguna funcionalidad extra para añadir…

Análisis

se determina y define las necesidades del cliente y especificación de los requisitos que debe cumplir el software

Características que deben cumplir los requisitos

- ➢ Deben ser completos y sin omisiones
- ➢ Concisos y sin trivialidades (toda la información necesaria e importante)
- ➢ Evitar ambigüedades y usar un lenguaje formal
- ➢ Evitar detalles de diseño o implementación
- ➢ Deben ser entendibles por el cliente
- ➢ Separar requisitos funcionales y no funcionales
- ➢ Dividir y jerarquizar el modelo (hacer categorías , partes en las que voy a dividir el trabajo)
- ➢ Fijar criterios de validación

Diseño

Aquí entran en juego las personas que imaginan la solución , hay que descomponer y organizar el sistema en elementos que puedan ser desarrollados por separado.

Se detallan las funcionalidades de las partes divididas anteriormente.

Las actividades habituales son las siguientes:

- ➢ Definir la arquitectura.
- ➢ Detallar qué es lo que va a hacer cada uno de los bloques.
- ➢ Ver qué tipos de datos usamos y cómo los repartimos.
- ➢ Definir la interfaz de usuario , es decir cómo interactúa el usuario con nuestro programa.

Codificación

Escribir el código fuente del programa, se pueden usar distintos lenguajes informáticos:

- ➢ Programación→ C, C++, Java, Javascript…
- ➢ Otro tipo→ HTML, XML, JSON….

Pruebas

Una vez ya hemos realizado el análisis , el diseño y la codificación llega el momento de realizar las pruebas

Su objetivo , es conseguir que el programa funcione incorrectamente para detectar los fallos del programa, sometiendo al programa al máximo número de situaciones diferentes.

Mantenimiento

Durante la explotación del sistema software es necesario realizar cambios ocasionales.

Tipos de mantenimiento

- **Correctivo**→ Corregir los defectos.
- **Perfectivo**→ Funciona correctamente pero hace que funcione mejor.
- **Evolutivo**→ Donde se añaden funcionalidades nuevas.
- **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.
- **Pruebas de unidades**→ Módulos utilizables.
- **Pruebas de integración**→ Sistema utilizables.
- **Pruebas del sistema**→ Sistema aceptado.
- **Documentación**→ Documentación técnica y de usuario.
- **Mantenimiento**→ Informes de errores y control de cambios.

Modelos de Desarrollo

Modelos de Desarrollo de Software

Distinguimos 3 grupos:

- Modelo Clásicos (Predictivos)
  - Modelos en Cascada
  - Modelo en V
- Modelo de Construcción de prototipos
- Modelo Evolutivo o Incrementales
  - Modelo en Espiral (Iterativos)
  - Metodologías Ágiles (Adaptativos)

Modelo en Cascada

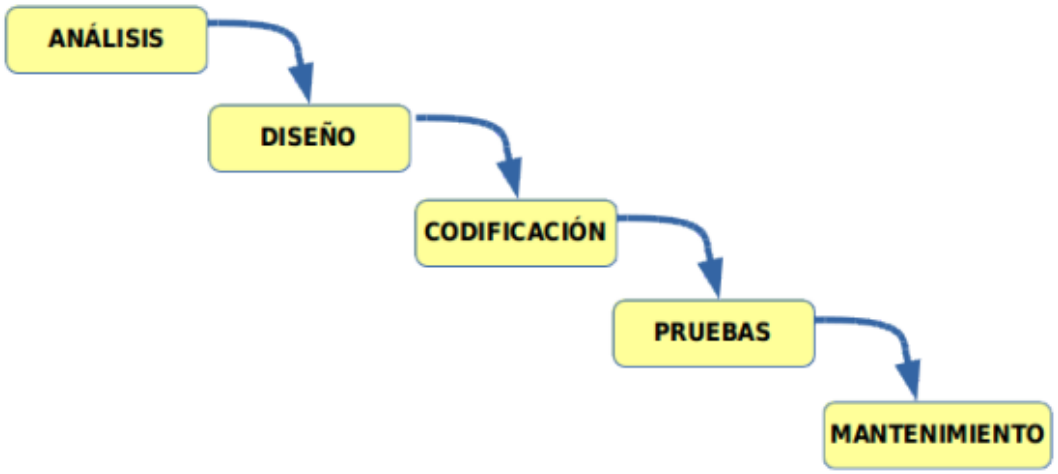
Es el modelo de mayor antigüedad , este identifica las fases principales del desarrollo del software:

- Análisis
- Diseño
- Codificación
- Pruebas
- Mantenimiento

Las fases deben realizarse en el orden especificado , el resultado de una fase es la entrada de la siguiente fase.

Es un modelo bastante rígido que se adapta mal al cambio continuo de especificaciones.

Existen diferentes variantes con mayor o menor cantidad de actividades.



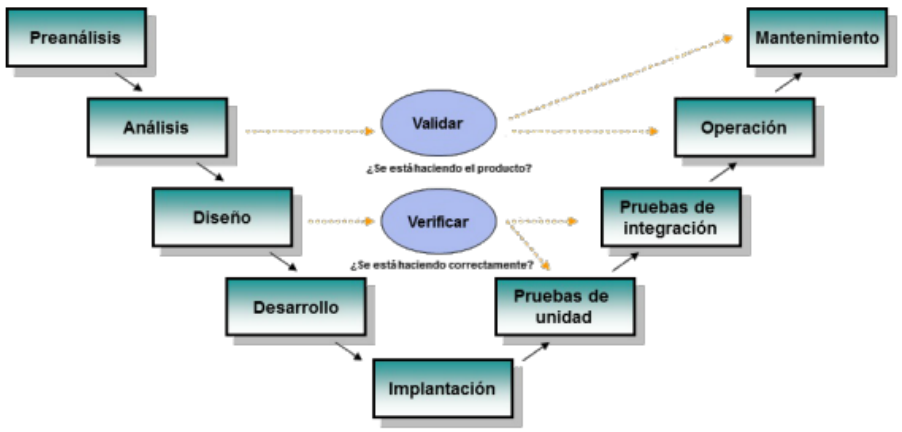
Modelo en V

Este modelo es muy parecido al modelo en cascada , nos da una visión jerarquizada con distintos niveles.

Los niveles superiores indican una mayor abstracción , mientras que, los niveles inferiores indican un mayor nivel de detalle.

El resultado de una fase es la entrada de la siguiente fase.

Existen diferentes variantes con mayor o menor cantidad de actividades.

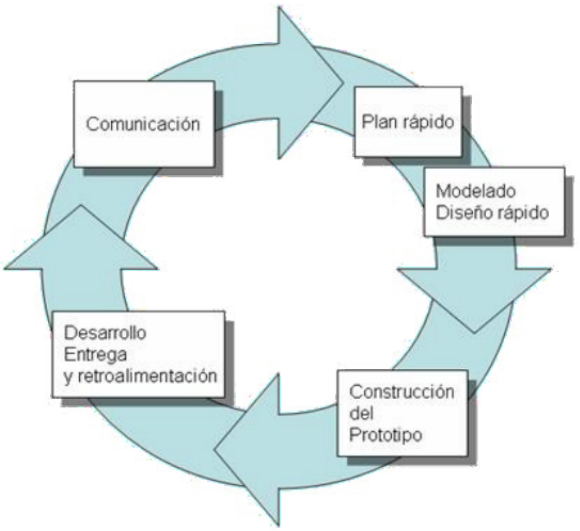


Prototipos

A menudo los requisitos no están especificados claramente ya sea:

- ➢ Por no existir una previa experiencia
- ➢ Por omisión o falta de concentración del usuario/cliente

Aquí un modelo de construcción de prototipos→



Prototipos II

El proceso es el siguiente:

- ➢ Primero se crea un prototipo durante la fase de análisis y es probado por el usuario/cliente para refinar los requisitos del software a desarrollar.
- ➢ Luego se repite el paso anterior las veces que sean necesarias.

Prototipos III

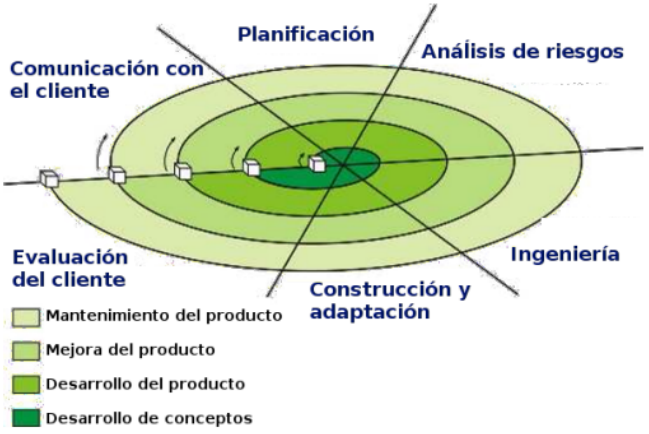
Tenemos varios tipos de prototipos:

- ➢ Prototipos Rápidos
  - ○ El prototipo puede estar desarrollado usando otro lenguaje o herramientas
  - ○ Finalmente el prototipo se desecha
- ➢ Prototipos Evolutivos
  - ○ Este prototipo está diseñado en el mismo lenguaje y herramientas del proyecto.
  - ○ El prototipo se usa como base para el desarrollo del proyecto.

Modelo en Espiral

Es desarrollado por Boehm en el año 1988 , En la actividad de ingeniería corresponde a las fases de los modelos clásicos:

- ➢ Análisis
- ➢ Diseño
- ➢ Codificación
- ➢ Pruebas
- ➢ Mantenimiento



Modelo en Espiral II

Aplicado a la programación orientada a objetos

- ➢ En La actividad de ingeniería se da gran importancia a la reutilización de código

Metodologías Ágiles