**ENTORNOS DE DESARROLLO**

**Software**: son programas y hay 3 tipos

* De sistema: interactúa directamente con el hardware. (so, drivers)
  + Drivers: controladores que tienen la función de que funcionen correctamente componentes como hardware.
  + Firmware: software de sistema más cercano al hardware.
* De aplicación: programas más independientes del hardware.
  + Ofimática
  + Navegadores.
* Desarrollo: programas que permiten crear otros programas.
  + Editores
  + Compiladores
  + Intérpretes

**Hardware**: parte física del ordenador.

* Disco duro: memoria secundaria.
* Memoria RAM: memoria principal y es volátil (si le quitas la corriente se borra la información).
* CPU: cerebro del ordenador y se encarga de leer y ejecutar las instrucciones almacenadas en la memoria RAM.
* Entrada y salida de datos: recoge y muestran datos. Teclado, ratón, monitores (si no es táctil).

Al abrir un programa se accede primero a la ram, y si no se encuentra ahí accede al disco duro.

**Código fuente**: permite modificar un programa de forma sencilla, código del programa

**Código objeto**: se genera a partir del código fuente y no se puede ejecutar. Se le llama bytecode.

**Código ejecutable**: archivo que se puede ejecutar, ocurren instrucciones que han sido escritas en el código fuente.

**Ciclo de vida del software**

**Ingeniería de software**

Disciplina que estudia los principios y metodologías para el desarrollo y mantenimiento de

sistemas software.

**Desarrollo de software**

* Análisis: analizar requisitos
* Diseño: ideas para implementar
* Codificación: programacion codigo
* Pruebas: detectar fallos
* Mantenimiento:

**Análisis:**

Se determina y define claramente las necesidades del cliente y se especifica los

requisitos que debe cumplir el software a desarrollar

La especificación de requisitos debe:

* Ser completa y sin omisiones
* Ser concisa y sin trivialidades
* Evitar ambigüedades. Utilizar lenguaje formal.
* Evitar detalles de diseño o implementación
* Ser entendible por el cliente: sencillo de comprender por el cliente
* Separar requisitos funcionales y no funcionales: funcionales(color agradables, botones,etc) no funcionales (apariencia)
* Dividir y jerarquizar el modelo:distintas partes de la aplicación, dividir trabajo
* Fijar criterios de validación: que le valga al cliente como quiere, por ejemplo tiempo de cargas al usar la aplicación

**Diseño:**

Se descompone y organiza el sistema en elementos componentes que pueden ser

desarrollados por separado. Dividir trabajo, intentar hacer muy bien diseñado las distintas partes que va a tener la aplicación.

Se especifica la interrelación y funcionalidad de los elementos componentes.

Diseño arquitectónico: partes del programa,diferentes métodos

Diseño detallado: definir por bloques,ej: entradas páginas etc

Diseño de datos:

Diseño de interfaz de usuario

\*mock up: a static design of a web page or application that features many of its final design elements but is not functional. ejemplo:prototipo

**Codificación:**

Se escribe el código fuente de cada componente.

**Pruebas:**

Identificar fallos

**Mantenimiento:**

correctivo: corregir fallos

preventivo: antes de que falle

perfectivo: se mejora la funcionalidad

evolutivo: se añaden nuevas funcionalidades, se paga aparte

Adaptativo: se adapta a nuevos entornos, optimizar.

Modelos de desarrollo

Modelos en cascada