

# CONCEPTOS DE SOFTWARE, PROGRAMA Y APLICACIÓN

## Software (SW)

- También denominado componente lógico.
- Es intangible, a diferencia del hardware que es tangible.
- Realiza una tarea determinada.
- Se comunica con el hardware para llevar a cabo su tarea.
- Se desarrolla, no se fabrica.
- Incluye los datos utilizados por los programas.
- Puede ser:
  - A medida.
  - Enlatado.
- Categorías:

## Según el tipo de tarea que realiza

- De sistema:
  - Hace que funcione el HW.
  - Administra los recursos HW.
  - Intermediario: Humano – HW
  - Ejemplo: SS. OO., Controladores (driver)
- De aplicación:
  - Nos ayudan a realizar tareas específicas → automatizan.
  - Ejemplo: CAD, SW médico, contabilidad, ofimática, ...
- De Programación o Desarrollo:
  - Proporciona herramientas al programador.
  - Suelen contar con un IDE (Entornos de Desarrollo Integrados).
  - Ejemplos: Eclipse.

## Según el método de distribución

- Shareware:
  - El usuario evalúa el producto por tiempo determinado.
  - {Adquirir su licencia → € | Precio cero}.
  - Ejemplo: Winrar.
- Freeware:
  - Se distribuye sin €.
  - Incluye una licencia:
    - Permite su redistribución.
    - Restricciones: vender, modificar.
  - No es software de libre distribución.
- Adware:
  - Shareware + se descarga automáticamente + publicidad.
  - Al comprar la licencia → desaparece la publicidad.
- Multimedia:
  - Integra textos, gráficos, sonidos, animaciones.
  - En distintos ámbitos como el educativo.
- De uso específico:
  - Resuelven un problema concreto.
  - Ejemplo: bancos, educación, ...

## Licencias de SW

- Licencia = contrato (desarrollador del sw, usuario).
- SW Libre:
  - Utilizado con cualquier fin.
  - Podemos estudiar su funcionamiento y modificarlo.
  - Distribuir copias.
  - Mejoras → hacerlas públicas → distribuir al público.
- SW Propietario:
  - El código fuente no es accesible.
  - Su licencia prohíbe 1 o varias posibilidades:
    - Redistribución
    - Copia
    - Uso en varias máquinas
    - Transferencia de titularidad
    - Difusión de fallos y errores
  - Variantes: Freeware y Shareware.
- SW de Dominio Público:
  - Se desconoce su autor o fin de propiedad de autor.
- GPL:
  - Licencia + Utilizada
  - SW Libre + fuente abierta.
  - Derecho a: usar, modificar haciendo públicas.

## Programa

- Es una serie de órdenes o instrucciones.
- Siguen un orden o secuencia para cumplir una finalidad determinada.
- Ejemplo: Holamundo.c

## Aplicación

- Puede estar formada por uno o varios programas.
- Está en contacto con el usuario, no con el hardware.
- Ejemplo: BB. DD., procesadores de texto, hojas de cálculo, correo electrónico.
- Finalizar: automatizar o ayudar a realizar tareas.
- Suite o Paquete Integrado:
  - Es el conjunto de programas que se ejecutan unos independientes del otro.
  - Ejemplo: Microsoft Office, Libre Office, Open Office.
  - Suelen compartir elementos comunes como librerías, datos, etc.
  - Son compatibles entre ellos.
- Tipos:
  - A medida:
    - Adaptado a los requerimientos de la empresa u organización.
    - Se desarrolla en función de las necesidades, actividad y forma de trabajar.
    - No suele ser compatible con empresas del sector.
    - Mantenimiento: es la etapa para subsanar y corregir errores.
    - Es más costoso, ya que va dirigido a un solo cliente.
    - Necesita tiempo de desarrollo.
  - Estándar o Enlatados:
    - Es software genérico.
    - Resuelve múltiples necesidades.

- Puede parametrizarse o adaptarse a las necesidades → herramientas de configuración.
- Pueden faltarle funciones → Se complementa con otras herramientas.
- Ya está hecho.
- Menor nº de errores → comprobado por varios clientes.
- Suele ser más barato.
- Pueden tener funciones que no usamos.

## CICLOS DE VIDA

- Son las fases o etapas en las que se desarrolla el software.

### Cascada

- Entrega el documento completo.
- Solo en el mantenimiento se vuelve a etapas anteriores (codificación → pruebas).
- Con realimentación:
  - Cuando debido a fallo, ambigüedades, etc., se vuelve a etapas anteriores.
- Ventajas, Inconvenientes, Cuando se recomienda.

### Evolutivos

- Los requisitos del usuario y del producto cambian cuando se desarrolla.
- Surgen versiones distintas.

### Iterativo incremental

- Basado en cascada.
- El SW se entrega en partes pequeñas → incrementos.
- Los incrementos son reutilizables.
- Ej: procesador de texto
  - 1º se desarrollan las funciones básicas como las de edición.
  - En cada incremento se desarrollarían nuevas funciones como la gramática u ortografía.
  - En el siguiente, funciones avanzadas de paginación.
  - ...

### En espiral

- Mezcla el modelo en Cascada y el Iterativo de construcción de prototipos.
- Cada ciclo → se divide en 4 fases.
- Es similar al iterativo incremental, pero este si analiza riesgos.
- Fases:
  - Determinar objetivos, alternativas, restricciones, ...
  - Análisis de riesgos: identificar los riesgos, como resolverlos, ...
  - Desarrollar y probar.
  - Planificación: se revisa lo hecho, si se continúa, se planifican las fases del ciclo siguiente.
- Muy utilizado para sistemas orientados a objetos.