

# Activitats UD1-ED

---

## Introducció

1. Descriu breument la relació que hi ha entre els components de maquinari principals d'un ordinador i l'emmagatzematge i l'execució del programari.

La relación entre hardware y software de un ordenador, es la siguiente:

El disco duro almacena de forma permanente todos los ficheros, la memoria RAM se trata de una memoria volátil, que almacena los ficheros de datos y ejecutables durante el tiempo necesario. Por otro lado el CPU se encarga de procesar todas las instrucciones del ordenador, leyendo y ejecutando ordenes y requisitos del sistema operativo, componentes y aplicaciones. Por ultimo, la puertos de entrada y salida se encargan de transmitir una serie de datos entre el ordenador y los periféricos.

2. Defineix els conceptes següents:
  - Codi font: es un fichero de texto escrito en algún lenguaje de programación, que indica a un dispositivo como debe ejecutar un programa.
  - Codi objecte: es un fichero traducido a binario, y que es entendido directamente por el ordenador.
  - Codi executable: es un código binario, que se encuentra empaquetado y listo para ser ejecutado en cualquier ordenador.

## Cicle de vida del programari

---

1. Defineix "Cicle de vida del programari".

Son las etapas por las que pasa un software para garantizar que este cumpla los requisitos para la aplicación y verificación de los procedimientos de desarrollo.

2. Anomena les fases principals del desenvolupament de programari i explica breument que es fa a cadascuna.

- **Análisis:** En esta fase, se recopilan y analizan las necesidades y requisitos del sistema. Los analistas de sistemas y los clientes trabajan juntos para identificar qué debe hacer el software. Se identifican las características, funcionalidades y restricciones del sistema y se documentan en una especificación de requisitos. El objetivo es comprender y definir claramente lo que se necesita que el software haga.
- **Diseño:** En esta fase, se crea un plan o diseño para el software basado en los requisitos establecidos en la fase de análisis. Esto implica la creación de una arquitectura de sistema, el diseño de interfaces de usuario, la definición de bases de datos y la planificación de la estructura general del programa. El resultado es un conjunto de documentos y diagramas que guiarán el desarrollo.
- **Codificación:** Esta fase implica la creación real del software mediante la escritura de código fuente. Los desarrolladores siguen las especificaciones de diseño y escriben el programa en uno o más lenguajes de programación. Esta es la parte más técnica del desarrollo de software, donde se transforman los conceptos y diseños en un producto funcional.

- **Pruebas:** Después de la codificación, se realizan pruebas para asegurarse de que el software cumple con los requisitos y funciones especificadas en la fase de análisis. Esto implica probar el programa en diferentes escenarios y validar que funcione correctamente. Se pueden identificar y corregir errores y deficiencias en esta fase.
- **Mantenimiento:** Una vez que el software está en uso, se continúa con el mantenimiento, que puede incluir corrección de errores, actualizaciones para acomodar cambios en los requisitos o el entorno, y mejora de las funcionalidades existentes. Esta fase tiene el objetivo de mantener el software en buen estado de funcionamiento durante su vida útil.

### 3. Avantatges i inconvenients del model en cascada.

#### Ventajas:

1. Estructura clara: El modelo en cascada sigue un proceso lineal y estructurado que es fácil de entender y seguir.
2. Fácil gestión y planificación: Permite una gestión y planificación más sencillas debido a su naturaleza secuencial con objetivos y entregables definidos.
3. Documentación temprana: Facilita la generación de documentación en cada fase, lo que mejora la revisión y comprensión de los requisitos y el diseño.

#### Desventajas:

1. Inflexible ante cambios: Carece de flexibilidad para adaptarse a cambios en los requisitos, lo que puede ser costoso y complicado.
2. No adecuado para proyectos complejos: Puede ser inadecuado para proyectos con requisitos inciertos, cambiantes o altamente innovadores.
3. Largo tiempo de entrega: Tiende a tener un tiempo de entrega largo, ya que todas las fases deben completarse antes de entregar el producto final.

### 4. Explica com funciona el model de desenvolupament mitjançant la creació de prototips.

El enfoque de desarrollo mediante la creación de prototipos permite una mayor interacción entre los desarrolladores y los usuarios desde las etapas iniciales del proyecto. Pues esto implica diseñar y construir una versión simplificada del producto antes de crear la versión final.

### 5. Quins quatre principis regeixen el desenvolupament àgil expressats al Manifest Àgil.

- Personas e interacciones sobre procesos y herramientas.
- Software funcionando sobre documentación extensa.
- Colaboración con el cliente sobre negociación contractual.
- Respuesta ante el cambio sobre seguir un plan.

### 6. Què és una història d'usuari? Consulta el següent enllaç i posa un exemple propi.

Una historia de usuario es una técnica utilizada en el desarrollo de software para capturar y describir las necesidades y requisitos de los usuarios de manera concisa y comprensible.

Como: usuario de correo electrónico, quiero: filtrar automáticamente los mensajes no deseados, para: evitar tener que eliminarlos manualmente.

Les podem formular de la següent manera: “Com a [perfil], vull [objectiu del software], per a poder [resultat]”.

[https://es.wikipedia.org/wiki/Historias\\_de\\_usuario](https://es.wikipedia.org/wiki/Historias_de_usuario)

---

7. KANBAN. Estudia els avantatges i els inconvenients de tenir una pissarra web digital per a la metodologia Kanban. Pots consultar els següents enllaços:

Ventajas:

- **Historial y seguimiento:** Las pizarras digitales suelen registrar el historial de las tareas y los cambios, lo que facilita el seguimiento del progreso y el análisis retrospectivo para mejorar el flujo de trabajo.
- **Visibilidad en tiempo real:** Las pizarras digitales ofrecen una vista en tiempo real del estado de las tareas y del flujo de trabajo. Esto ayuda a los miembros del equipo a estar siempre al día y a tomar decisiones informadas.
- **Personalización:** Muchas pizarras digitales permiten personalizar los tableros Kanban para adaptarlos a las necesidades específicas del equipo y del proyecto.

Desventajas:

- **Dependencia de la tecnología:** Las pizarras digitales dependen de la tecnología y la conexión a Internet. Si hay problemas técnicos, puede interrumpirse el flujo de trabajo.
- **Seguridad y privacidad:** Se deben tomar precauciones adicionales para garantizar la seguridad y la privacidad de los datos de la empresa cuando se utiliza una pizarra web digital.
- **Posible distracción:** Si no se utiliza de manera adecuada, una pizarra digital puede convertirse en una distracción en lugar de una herramienta de productividad.

8. KANBAN. Fes un resum de la metodologia Kanban i indica les seves diferències davant de SCRUM. Pots consultar el següent enllaç:

Kanban es una metodología de gestión de proyectos y flujos de trabajo que se basa en la visualización, el control del trabajo en curso y el flujo continuo. Utiliza tableros visuales para representar tareas, establece límites de trabajo en curso para aumentar la eficiencia y promueve la mejora continua del proceso.

Kanban se basa en un flujo continuo de trabajo, es altamente adaptable y se centra en la eficiencia y la mejora continua del proceso, mientras que Scrum utiliza iteraciones fijas y roles definidos, y se enfoca en la planificación a corto plazo.

<https://ca.atlassian.com/agile/kanban>

---

9. SCRUM. Explica com funciona Scrum. Consulta els enllaços següents:

Scrum funciona como modelo incremental, estableciendo iteraciones regulares, de 2 a 4 semanas. Además, al comenzar la iteraciones se fijan una serie de objetivos prioritarios(sprint backlog).

<https://proyectosagiles.org/que-es-scrum/>

<https://proyectosagiles.org/como funciona-scrum/>

---

10. SCRUM. Defineix els termes següents:

- Product backlog: Una lista priorizada de requisitos y elementos de trabajo para el producto en su conjunto.
- Sprint backlog: Es una selección específica de elementos del Product Backlog que el equipo se compromete a completar durante un sprint determinado.

---

11. SCRUM. A la terminologia Scrum quins termes s'utilitzen com a sinònim de:

- Cap de projecte: Product Owner
- Client: Scrum Master
- Equip de desenvolupament: Team

12. XP. Quines són les característiques distintives de XP davant d'altres metodologies àgils? Explica-les. Pots consultar el següent enllaç:

Las principales distinciones de XP respecto el resto de metodologías ágiles, son las siguientes:

- Diseño sencillo
- Pequeñas mejoras continuas
- Pruebas y refactorización
- Integración continua
- Programación por parejas
- El cliente se integra al equipo de desarrollo
- Propiedad del código compartida
- Estándares de codificación
- 40 horas semanales

<http://www.davidvalverde.com/blog/introduccion-a-la-programacion-extrema-xp/>

---

## Llenguatges de programació

1. Quina diferència hi ha entre els llenguatges declaratius i els imperatius? Anomena almenys 2 de cada tipus.

La principal diferencia entre estos dos lenguajes es su manera para obtener un resultado.

En los lenguajes declarativos se indica el resultado que queremos obtener sin especificar los pasos. Mientras que en los lenguajes imperativos se indica los pasos que se deben realizar para obtener un resultado.

Los lenguajes declarativos son lógicos y funcionales, mientras que los imperativos son estructurados y multiparadigmas.

2. Explica què és compilar? Explica què és interpretar?

La compilación transforma el código fuente en código de máquina antes de la ejecución, mientras que la interpretación ejecuta el código fuente línea por línea en tiempo real.

3. Avantatges dels llenguatges compilats.

- Su ejecución es muy eficiente
- Detección temprana de errores
- Optimización

4. Avantatges dels llenguatges interpretats

- El código fuente se interpreta directamente
- Facilidad de desarrollo y pruebas
- Facilidad de distribución

5. Anomena 2 llenguatges compilats i altres 2 interpretats.

Lenguajes compilados:

- C
- C++

Lenguajes interpretados:

- PHP
- Javascript

6. Es pot considerar codi objecte el **bytecode** generat a Java després de la compilació? Explica'n la resposta.

Se puede considerar como una forma de código objeto pero destinado a la maquina virtual de Java en vez de código objeto nativo.

7. Posa un exemple de llenguatge dels tipus següents:

- Baix nivell: ensamblador
- Nivell mitjà: C
- Alt nivell: Java

---

8. Quin paradigma de programació segueixen els llenguatges següents?

- C: Programación imperativa.
- C++: Combina la programación imperativa con la programación orientada a objetos.
- SQL: Lenguaje declarativo.
- Java: Programación orientada a objetos.
- Javascript: Programación imperativa y también es compatible con la programación orientada a objetos y la programación funcional.
- Lisp: Programación funcional.
- Prolog: Programación lógica.

---

9. Explica quins criteris es poden seguir a l'hora d'escollir un llenguatge de programació per al desenvolupament de programari.

Al momento de escoger un lenguaje de programación, se deben de tener en cuenta una serie de criterios:

- **Campo de aplicación:** Se refiere a qué tipo de problema o aplicación estás tratando de resolver.
- **Experiencia previa:** Si ya tienes experiencia en un lenguaje de programación, es más probable que lo elijas para un nuevo proyecto.
- **Herramientas de desarrollo:** La disponibilidad de herramientas de desarrollo, como IDEs, depuradores y editores de código
- **Documentación disponible:** Una amplia base de conocimientos y documentación puede facilitar la resolución de problemas.
- **Base de usuarios:** Una buena comunidad puede proporcionar soporte, bibliotecas adicionales y contribuir al crecimiento del lenguaje.
- **Reusabilidad:** La capacidad de reutilizar código previamente escrito es importante para la eficiencia y la velocidad en el desarrollo.
- **Transportabilidad:** La capacidad de ejecutar el código en diferentes plataformas o sistemas operativos puede ser un factor muy importante.
- **Imposición del cliente:** Puede ser que los clientes o las organizaciones puedan imponer la elección de un lenguaje en función de sus necesidades.