

Introducción

1. Describe brevemente la relación que hay entre el hardware principal de un ordenador y el almacenamiento y la ejecución de un programa. Puedes basarte en la máquina de Von Newman.

- **Disco Duro:** Almacena de forma permanente los ficheros ejecutables y los ficheros de datos.

- **Memoria RAM:** Almacena temporalmente el código binario de los ficheros ejecutables y los ficheros de datos necesarios.

- **CPU:** Lee y ejecuta instrucciones almacenadas en la memoria RAM, así como sus datos necesarios.

- **Hardware E/S (De entrada y salida):** Recoge los nuevos datos desde la entrada, se muestran los resultados y se leen/guardan en el disco.

2. Define los siguientes conceptos:

- **Código fuente** = Fichero de texto legible escrito en un lenguaje de programación.

- **Código objeto** = Fichero binario no ejecutable.

- **Código ejecutable** = Fichero binario ejecutable.

Ciclo de vida de un programa

1. Define “Ciclo de vida de un programa”

- El ciclo de vida de un programa es la estructura que contiene los procesos, actividades y tareas relacionadas con el desarrollo y mantenimiento de un producto de software, abarcando la vida completa del sistema, desde la definición de los requisitos hasta la finalización de su uso.

2. Nombra las principales fases del desarrollo de programas y explica brevemente qué se hace en cada uno de ellos.

- **Análisis:** Se determina y define claramente las necesidades del cliente y se especifican los requisitos que ha de cumplir el programa a desarrollar.

- **Diseño:** Se descompone y se organiza el sistema en diferentes elementos que puedan elaborarse por separado, para así trazar una estructura interna del software.

- **Codificación:** Se escribe el código fuente de cada componente (se pueden utilizar diferentes lenguajes de programación)
- **Pruebas:** El principal objetivo de las pruebas es conseguir que el programa funcione de manera incorrecta para así descubrir los defectos del programa.
- **Mantenimiento:** Se destinan recursos para el mantenimiento y correcto funcionamiento del programa una vez sea instalado por el usuario final. Podremos modificar, mejorar y corregir errores o mejorar ciertos aspectos del programa para que cumpla con las expectativas.

3. Ventajas e inconvenientes del modelo en cascada.

- Ventajas:

- Estructura sencilla gracias a unas fases de proyecto claramente diferenciadas que se deben hacer en un orden específico, el resultado de una fase es la entrada de la siguiente fase.
- Los costes y las cargas de trabajo se pueden estimar al comenzar el proyecto.

- Inconvenientes:

- Poco margen para realizar ajustes a lo largo del proyecto debido a un cambio en las exigencias.
- En ocasiones los fallos solo se detectan una vez finalizado el proceso de desarrollo.
- Los proyectos más complejos no permiten su división en fases de proyecto claramente diferenciadas.

4. Explica cómo funciona el modelo de desarrollo mediante la creación de prototipos.

- El modelo de desarrollo mediante la creación de prototipos se centra en un diseño rápido que representa las características principales del programa que el usuario podrá ver o utilizar. De esta manera pueden probarlo y dar su opinión sobre distintos aspectos como la usabilidad, la utilidad o el rendimiento, etc.. El prototipo se puede modificar cuando sea necesario y todos los resultados obtenidos de las presentaciones y pruebas se deben anotar para utilizar posteriormente como ayuda en el desarrollo del producto final.

5. ¿Qué 4 principios rigen el desarrollo ágil expresados en el manifiesto ágil?

- Individuos e interacciones por encima de procesos y herramientas.
- Software funcionando por encima de documentación exhaustiva.
- Colaboración con el cliente por encima de negociación contractual.
- Respuesta ante el cambio por encima de seguir un plan.

6. ¿Qué es una historia de usuario? Consulta el siguiente enlace

(https://es.wikipedia.org/wiki/Historias_de_usuario) y pon un ejemplo propio.

- Es una representación de un requisito escrito en una o dos frases que centra la atención en el usuario, con el objetivo de ofrecer un mejor servicio.

Ejemplo: Historia: Agregar comentarios en las entradas del blog y recibir alertas cuando otros lo hagan, Como: Pedro Pérez.

7. KANBAN. Estudia las ventajas y los inconvenientes de tener una pizarra web digital para la metodología Kanban. Puedes consultar los siguientes enlaces:

- <https://www.planview.com/resources/guide/introduction-to-kanban/what-is-kanban-board/>
- <https://trello.com/es>
- <https://www.taiga.io/es>
- <https://kanbantool.com/es/>

- Ventajas:

- Promueve la concentración en un mismo objetivo y el trabajo en equipo.
- Aumenta la productividad.
- Incrementa la organización.

- Inconvenientes:

- No le encuentro inconvenientes.

12. KANBAN. Haz un resumen de la metodología Kanban e indica en qué se diferencia de SCRUM. Puedes consultar el siguiente enlace (<https://www.atlassian.com/es/agile/kanban>).

- La metodología Kanban se implementa por medio de tableros Kanban. Se trata de un método visual de gestión de proyectos que permite a los equipos visualizar sus flujos de trabajo y la carga de trabajo siguiendo un backlog, centrándose únicamente en el trabajo que está en curso.

- En Scrum existen los roles de Scrum Master, de Product Owner y del equipo, mientras que Kanban no existen roles, en Scrum se trabaja con iteraciones de tiempo fijo, con unos ciclos fijos que se denominan Sprints, mientras que en Kanban tenemos un trabajo continuo y no tenemos esas interacciones o esos ciclos durante el desarrollo, no se permiten cambiar las tareas del Sprint.

13. SCRUM. Explica cómo funciona Scrum. Consulta los siguientes enlaces:

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

Scrum es un framework que se utiliza dentro de equipos que manejan proyectos de alta incertidumbre. Se trata de un marco de trabajo por el cual las personas pueden abordar problemas complejos adaptativos, a la vez que entregar productos del máximo valor posible productiva y creativamente. Scrum es liviano y fácil de entender pero, a la vez, difícil de dominar del todo.

14. SCRUM. Define los siguientes términos:

- **Product backlog** = El product backlog (o pila de producto) es un listado de todas las tareas que se pretenden hacer durante el desarrollo de un proyecto.
- **Sprint backlog** = Conjunto de tareas seleccionadas del product backlog durante el sprint planning para el Sprint actual, es decir, las tareas necesarias para realizar un incremento de producto.

15. SCRUM. En la terminología Scrum qué términos se utilizan como sinónimo de:

- **Cabeza de proyecto** = Product owner.
- **Cliente** = User.
- **Equipo de desarrollo** = Development Team.

16. XP. ¿Cuáles son las características distintivas de XP respecto a otras metodologías ágiles? Puedes consultar el siguiente enlace (<https://www.davidvalverde.com/blog/introduccion-a-la-programacion-extrema-xp/>)

- Software que funciona por encima de una buena documentación.
- Interacción constante entre el cliente y el equipo de desarrollo.
- Planificación flexible y abierta.
- Rápida respuesta a cambios.

Lenguajes de programación

1. ¿Qué diferencias hay entre los lenguajes declarativos y los imperativos? Menciona al menos 2 de cada tipo.

- En un lenguaje imperativo tienes que hacer una secuencia de operaciones a realizar para llegar a cierto resultado, mientras que en un lenguaje declarativo se especifica el resultado que se desea y no el cómo lograrlo.

Ejemplos de lenguaje **imperativo**: Java, Python, etc..

Ejemplos de lenguaje **declarativo**: SQL, Prolog, etc..

2. ¿Qué es compilar?, ¿Qué es interpretar?

- **Compilar** = Compilar es un proceso que traduce el código fuente de un proyecto de software (Legible para humanos) a código máquina (Legible para máquinas).

- **Interpretar** = Interpretar es analizar el código de un programa y realizar los cálculos que éste indica de forma directa.

3. Ventajas de los lenguajes compilados.

- Es más rápido que un lenguaje interpretado ya que no necesita trabajar sobre la marcha mientras la aplicación se está ejecutando.
- Durante su compilación se revisa si hay errores

4. Ventajas de los lenguajes interpretados.

- Son multiplataforma ya que el intérprete suele estar en varios sistemas operativos, por lo que no tienes que adaptar tu código a una plataforma en concreto.

5. Menciona 2 lenguajes compilados y otros 2 interpretados.

- Lenguajes compilados: C, C++.
- Lenguajes interpretados: PHP, Javascript.

6. ¿Se puede considerar código objeto el bytecode generado en Java después de la compilación? Justifica tu respuesta.

- Si, se puede considerar código objeto, pero destinado a la máquina virtual de Java en vez de código objeto nativo.

7. Pon un ejemplo de lenguaje de los siguientes tipos:

- **Bajo nivel**: Lenguaje máquina
- **Medio nivel**: C
- **Alto nivel**: Java

8. ¿Qué paradigma de programación siguen los siguientes lenguajes?

- **C** = Imperativo
- **C++** = Imperativo

- **SQL** = Declarativo
- **Java** = Imperativo
- **Javascript** = Imperativo
- **Lisp** = Declarativo
- **Prolog** = Declarativo

Puedes consultar el siguiente enlace:

(https://es.wikipedia.org/wiki/Paradigma_de_programaci%C3%B3n)

9. Explica qué criterios se pueden seguir a la hora de elegir un lenguaje de programación para el desarrollo de programas.

- El campo de aplicación, experiencia previa, herramientas de desarrollo, documentación disponible, base de usuarios, reusabilidad, transportabilidad, imposición del cliente.

ACTIVIDAD FINAL

Contesta las preguntas de esta unidad y guarda el archivo en GitHub en un repositorio llamado “Activitats_ED”.