

# Actividades

## Introducción

1. Describe brevemente la relación que hay entre los componentes de maquinaria principales de un ordenador y el almacenamiento y la ejecución del programa. Puedes basar-te en la maquina de Von Newman.

- **Disco duro:** es un dispositivo de almacenamiento necesario para conservar tus archivos y datos a largo plazo. Cuando almacenas un archivo en tu ordenador, lo guardas en el disco duro, el cual podría considerarse un archivador para tus archivos digitales.
- **Memoria RAM:** es la memoria de corto plazo del ordenador. Su función principal es recordar la información que tienes en cada una de las aplicaciones abiertas en el ordenador, mientras este se encuentre encendido.
- **CPU:** es el componente más importante dentro del ordenador. Es el cerebro de todo el funcionamiento del sistema, el encargado de dirigir todas las tareas que lleva a cabo el equipo y de ejecutar el código de los diferentes programas.
- **E/S:** proporciona una interfaz homogénea para acceder a todos los sistemas de red que proporciona el sistema operativo. Además permite acceder a los manejadores de cada tipo de red particular de forma transparente.

2. Define los conceptos siguientes:

- **Código fuente.** Es todo texto legible por un ser humano y redactado en un lenguaje de programación determinado. El objetivo del código fuente es crear normas y disposiciones claras para el ordenador y que este sea capaz de traducirlas a su propio lenguaje.
- **Código objeto.** Conjunto de instrucciones y datos escritos en un lenguaje que entiende el ordenador directamente: binario o código máquina. Proviene de la traducción de cierto código fuente, es un fragmento del programa final y es específico de la plataforma de ejecución.
- **Código ejecutable.** Es un lenguaje creado por maquina que se corresponde a las unidades de programas, donde el ordenador realiza las instrucciones compiladas por medio del enlazador de librerías y el compilador.

# Ciclo de vida del programa

1. D¿Cómo hay que colocarse el cinturón de seguridad?

2.efine "Ciclo de vida del programa".

Es una tarea que consta de una serie de pasos de obligado cumplimiento que busca garantizar que los programas creados son eficientes, fiables, seguros y responden a las necesidades de los usuarios finales.

3. Nombra las fases principales del desarrollo del programa y explica brevemente que se hace a cada una.

## **Planificación**

Es necesario hacer ciertas tareas que influirán decisivamente en el éxito del mismo.

Algunas de las tareas de esta fase incluyen actividades como la determinación del ámbito del proyecto, la realización de un estudio de viabilidad, el análisis de los riesgos asociados, la estimación del coste del proyecto, su planificación temporal y la asignación de recursos a las diferentes etapas del proyecto.

## **Análisis**

Hay que averiguar qué es exactamente lo que tiene que hacer el software. Por eso, la etapa de análisis en el ciclo de vida del software corresponde al proceso a través del cual se intenta descubrir qué es lo que realmente se necesita y se llega a una comprensión adecuada de los requerimientos del sistema.

## **Diseño**

En esta fase se estudian posibles opciones de implementación para el software que hay que construir, así como decidir la estructura general del mismo.

Es posible que la solución inicial no sea la más adecuada, por lo que en tal caso hay que refinar-la.

## **Implementación**

En esta fase hay que elegir las herramientas adecuadas, un entorno de desarrollo que facilite el trabajo y un lenguaje de programación apropiado para el tipo de software a construir. Esta elección dependerá tanto de las decisiones de diseño tomadas como del entorno en el que el software deba funcionar.

Al programar, hay que intentar que el código no sea indescifrable siguiendo distintas pautas como las siguientes:

- Evitar bloques de control no estructurados.
- Identificar correctamente las variables y su alcance.
- Elegir algoritmos y estructuras de datos adecuadas para el problema.
- Mantener la lógica de la aplicación lo más sencilla posible.

- Documentar y comentar adecuadamente el código de los programas.
- Facilitar la interpretación visual del código utilizando reglas de formato de código previamente consensuadas en el equipo de desarrollo.

También hay que tener en cuenta la adquisición de recursos necesarios para que el software funcione, además de desarrollar casos de prueba para comprobar el funcionamiento del mismo según se vaya programando.

## **Pruebas**

La fase de pruebas del ciclo de vida del software busca detectar los fallos cometidos en las etapas anteriores para corregirlos. Por supuesto, lo ideal es hacerlo antes de que el usuario final se los encuentre. Se dice que una prueba es un éxito si se detecta algún error.

## **Instalación o despliegue**

La siguiente fase es poner el software en funcionamiento, por lo que hay que planificar el entorno teniendo en cuenta las dependencias existentes entre los diferentes componentes del mismo.

Es posible que haya componentes que funcionen correctamente por separado, pero que al combinarlos provoquen problemas. Por ello, hay que usar combinaciones conocidas que no causen problemas de compatibilidad.

## **Uso y mantenimiento**

Esta es una de las fases más importantes del ciclo de vida de desarrollo del software. Puesto que el software ni se rompe ni se desgasta con el uso, su mantenimiento incluye tres puntos diferenciados:

- Eliminar los defectos detectados durante su vida útil (mantenimiento correctivo).
- Adaptarlo a nuevas necesidades (mantenimiento adaptativo).
- Añadirle nuevas funcionalidades (mantenimiento perfectivo).

## **4. Ventajas e inconvenientes del modelo en cascada.**

**Ventajas:** Una estructura sencilla gracias a unas fases de proyecto claramente diferenciadas.

Buena documentación del proceso de desarrollo a través de unos hitos bien definidos.

Los costes y la carga de trabajo se pueden estimar al comenzar el proyecto.

Aquellos proyectos que se estructuran en base al modelo en cascada se pueden representar cronológicamente de forma sencilla.

**Inconvenientes:** Por norma general, los proyectos más complejos o de varios niveles no permiten su división en fases de proyecto claramente diferenciadas.

Poco margen para realizar ajustes a lo largo del proyecto debido a un cambio en las exigencias.

El usuario final no se integra en el proceso de producción hasta que no termina la programación.

En ocasiones, los fallos solo se detectan una vez finalizado el proceso de desarrollo.

#### 5. Explica como funciona el modelo de desarrollo mediante la creación de prototipos.

Se utiliza en el desarrollo de software para ofrecer al usuario una visión previa de cómo será el programa o sistema. Se le quiere diseñar una Base de Datos para almacenar todos los datos de un campeonato de fútbol sala que se organiza este año en la ciudad. Aquellos que quieran participar deberán formar un equipo (nombre, patrocinador, color de la 1ª camiseta, color de la 2ª camiseta, categoría, . . .) e inscribirse en el campeonato. A medida que transcurran los partidos se irán almacenando los resultados de éstos, así como qué equipos lo jugaron, en qué campo se jugó, quién lo arbitró y alguna incidencia que pudiera haber ocurrido (en caso de que no ocurran incidencias no se anotará nada. Además, los participantes deberán rellenar una ficha de suscripción con algunos datos personales (nombre, apellidos, edad, dirección, teléfono, . . .) dice de desarrollo evolutivo al modelo de prototipo porque evoluciona hasta convertirse en el producto final.

#### 6. Que cuatro principios rigen el desarrollo ágil expresado al Manifiesto Ágil?

- Los individuos y sus interacciones por encima a los procesos y las herramientas
- El producto funcionando por encima a la documentación exhaustiva
- La colaboración con el cliente por encima de la negociación contractual
- Adaptación al cambio por encima a seguir un plan rígido

#### 7. Que es una historia de usuario? Consulta el siguiente enlace y pon un ejemplo propio.

Los podemos formular de la siguiente manera: “Como a [perfil], quiero [objetivo del software], para poder [resultado]”.

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

Las historias de usuario son parte de un enfoque ágil que ayuda a cambiar el enfoque de escribir sobre los requisitos a hablar sobre ellos.

Ejemplo: Como usuario, quiero poder entrar a tu pagina todas las veces que quiera.

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

<https://leankit.com/learn/kanban/kanban-board/>

<https://trello.com/es>

<https://taiga.io/>

<https://kanbantool.com/es/>

**Ventajas:** Medición del rendimiento, organización del flujo de trabajo, cumplir los tiempos de entrega, distribución de tareas, mejorar la calidad del producto y evitar la acumulación de trabajo.

**Inconvenientes:** Coste, es limitado, Kanban no es óptimo para todo tipo de proyectos y no permite anticiparse a grandes aumentos de la demanda.

12. KANBAN. Haz un resumen de la metodología Kanban e indica sus diferencias delante de SCRUM. Puedes consultar el siguiente enlace:

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

Kanban se basa en el desarrollo y entrega continuos, abordando un pequeño número de tareas de forma fluida y simultánea.

Los equipos Kanban utilizan una herramienta de planificación visual, el tablero Kanban, que muestra cada historia de usuario en una tarjeta y mueve esas tarjetas a través de columnas que representan etapas progresivas de finalización.

Los equipos de Scrum se comprometen a generar software al final de los intervalos establecidos.

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

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

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

En Scrum no se hace una entrega final del proyecto sino que se van haciendo de entregas parciales, de forma que esto es lo que más beneficia al receptor del proyecto. Scrum está especialmente indicado para entornos complejos, donde los cambios se producen como mucha frecuencia y sobre la marcha y donde la rapidez, la flexibilidad, la adaptabilidad y la competencia juegan un papel fundamental.

14. SCRUM. Define los términos siguientes:

- Product backlog. Es una lista de todas las tareas que se quieren hacer durante el desarrollo.
- Sprint backlog. Es un periodo de tiempo en el que dentro de el se pretende hacer una determinada cantidad de cosas para el desarrollo.

15. SCRUM. A la terminología Scrum que términos se utilizan como a sinónimo de:

- Cabeza de proyecto. Project Owner
- Cliente. Product Owner
- Equipo de desarrollo. Development Team

16. XP. Cuales son las características distintivas de XP delante de otras metodologías ágiles? Explica-las. Puedes consultar el siguiente enlace:

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

**Comunicación:** Su objetivo es que hayan buenas interacciones con todos y promueve que todo se haga mediante comunicación y ni documentos.

**Simplicidad:** Centrarse primero en las partes mas sencillas del desarrollo y mientras va creciendo el código seguirá siendo simple.

**Retroalimentación:** Se trata cuando el cliente esta integrado y da su opinión sobre lo que hay, de esta forma el cliente puede decir que quiere y los programadores pueden centrarse en ellas.

**Valentía:** Trata de que cuando se comete algún error no dejarlo atrás porque de esta forma bajara la productividad, si se comete alguno hay que admitirlo que ha sido tuyo y solucionarlo.

## Lenguajes de programación

1. Que diferencia hay entre los lenguajes declarativos y los imperativos? Nombra al menos 2 de cada tipo.

Imperativo: es una secuencia de operaciones que se tienen que hacer. Fortran y Java

Declarativo: pone el resultado deseado, pero no como se consigue. SQL y Erlang

2. Explica que es compilar? Explica que es interpretar?

Compilar: Traducir código programable a código que pueda ejecutarse.

Interpretar: Ejecuta código directamente sin tener que traducirlo a otro.

3. Ventajas de los lenguajes compilados.

La mayor ventaja que puedes obtener es la velocidad de traducción de un código de programación a un código ejecutable, obteniéndolo casi de inmediato.

4. Ventajas de los lenguajes interpretes.

Tiene varias ventajas como estar en distintos sistemas operativos, que se pueda llevar a otras plataformas y el aumento de rendimiento que se obtiene.

5. Nombra 2 lenguajes compilados y otros 2 interpretados.

Compilados: C++ y Cobra Interpretados: Javascript y Python.

6. Se puede considerar código objeto el **bytecode** generado a Java después de la compilación? Explica la respuesta. Si se trata de código objeto ya que luego de compilarse se traduce al código el cual el ordenador entiende directamente

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

- Bajo nivel. Código Binario y Lenguaje de Maquina
- Nivel medio. C y Basic
- Alto nivel. C++ y Java

8. Que paradigma de programación siguen los lenguajes siguientes?

- C Paradigma Imperativo
- C++ Paradigma Orientado a Objetos
- SQL Paradigma Declarativo
- Java Paradigma Orientado a Objetos e Imperativo
- Javascript Paradigma Declarativo o Imperativo
- Lisp Paradigma Procedimental y Funcional
- Prolog Paradigma Logico

Puedes consultar el siguiente enlace:

- <https://ca.wikipedia.org/wiki/Paradig>Explica que criterios se pueden seguir a la hora de elegir un lenguaje de programación para el desarrollo del programama\_de\_programaci%C3%B3n

9. Explica que criterios se pueden seguir a la hora de elegir un lenguaje de programación para el desarrollo del programa.

Aplicación: A la hora de desarrollar juegos o aplicaciones móviles,etc.. Son lenguajes totalmente diferentes y por tanto tienes que aprenderlos bien.

Salario: Hay diferentes salarios para cada lenguaje y por tanto tienes que elegir bien cuales querrías aprender para cobrar por ellos.

Geografía: Tienes que tener en cuenta a la hora de elegir un lenguaje si se utiliza mucho por donde vives o también si quieres hacer que un lenguaje que se utiliza poco destaque.

Popularidad: El lenguaje que vayas a utilizar tienes que tener en cuenta que sea popular para que genere proyectos en los que puedas trabajar.

Actividad final (no realizar aun)

Elabora las respuestas de las preguntas de esta Unidad y guarda el archivo a GitHub en un repositorio llamado 'Activitats\_ED'.