

Tarea 4

Lenguajes de programación

1. ¿Qué diferencia existe entre los lenguajes declarativos y los imperativos? Nombra al menos 2 de cada tipo.

La programación declarativa consiste en decirle a un programa qué tiene que hacer en lugar de decirle cómo debería hacerlo.

Los lenguajes **declarativos** se caracterizan por tener una sintaxis abreviada y abstracta.

Son ejemplos de lenguajes de programación declarativo Prolog, Haskell, entre otros.

Por su parte, la programación **imperativa** se centra en cómo hay que realizar una tarea.

Mediante las correspondientes sentencias le damos las órdenes de lo que queremos que haga.

Estos lenguajes destacan por presentar un código fuente largo pero en general fácil de entender.

Existen muchos ejemplos de lenguajes de programación imperativos entre ellos Java, C++, etc.

2. ¿Explica qué es compilar? ¿Explica qué es interpretar?

La tarea de **compilar** se refiere al proceso de traducción del código fuente. Esta compilación del código fuente de un programa se realiza debido a que el código trabajado por el lenguaje de programación no es ejecutable directamente por el ordenador. Es por eso que se traducen todas las instrucciones contenidas en el texto al llamado lenguaje de máquina, un código fácil de entender para el dispositivo y que luego pueda ejecutar las sentencias, órdenes e instrucciones determinadas por el código fuente.

Por otro lado, al **interpretar** un programa el código se ejecuta directamente línea por línea a la vez que se va leyendo, en este caso por un programa denominado intérprete.

3. Ventajas de los lenguajes compilados.

Ventajas de los lenguajes compilados

→ Son más eficientes y rápidos a la hora de ejecutarse, debido a que llegados a este punto, se encuentra en código máquina

→ No necesita de otro programa externo para ser ejecutado.

4. Ventajas de los lenguajes interpretados.

Ventajas de los lenguajes interpretados:

→ El ciclo de desarrollo es más rápido en un lenguaje interpretado

→ Ofrecen características como escritura dinámica y un tamaño de programa más pequeño.

5. Nombra 2 lenguajes compilados y otros 2 interpretados.

Dos ejemplos de lenguajes compilados pueden ser C y Rust.

En cuanto a lenguajes interpretados podemos destacar PHP y Ruby, entre otros.

6. ¿Puede considerarse código objeto el bytecode generado en Java tras la compilación? Explica la respuesta.

El bytecode de Java es el resultado de la compilación del código fuente, sin embargo, ya que este solo se puede ejecutar en la máquina virtual de Java, podríamos suponer que sí se puede considerar un código objeto.

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

- Bajo nivel → Ensamblador
- Nivel medio → C
- Alto nivel → PHP

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

- C → Imperativo estructurado
- C++ → Imperativo multiparadigma
- SQL → Declarativo algebraico
- Java → Imperativo orientado a objetos
- Javascript → Imperativo multiparadigma
- Lisp → Declarativo funcional
- Prolog → Declarativo lógico

Puedes consultar el siguiente enlace:

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

9. Explica qué criterios pueden seguirse a la hora de elegir un lenguaje de programación para el desarrollo software.

A la hora de elegir el lenguaje de programación correcto en un proyecto determinado, decimos que existen una serie de criterios a tener en cuenta:

– **Campo de aplicación:** Tener claro tanto el sector como a las personas a la que el producto final va dirigido.

– **Experiencia previa:** Detallar si se ha trabajado con el lenguaje antes o no.

- **Herramientas de desarrollo:** Qué tipo de herramientas se puede usar con el programa, con el fin de buscar una eficiencia y productividad óptima.
- **Documentación disponible:** Cuánta información existe del programa acerca de su arquitectura, codificación, procesos, etc.
- **Base de usuarios:** Puntualizar cuántas desarrolladores/usuarios están utilizando el programa en el momento.
- **Reusabilidad:** Destacar si las estructuras del lenguaje utilizado son aptas para reutilizarlas en otras partes del desarrollo o incluso en otros proyectos.
- **Transportabilidad:** Mirar la facilidad con la que el software puede ser ejecutado en un entorno distinto del que fue desarrollado originalmente.
- **Imposición del cliente:** Adaptarse a la elección del cliente.