

## A1.1 Conceptos

### **Qué es un IDE y di al menos 3 de los más usados.**

Un IDE (*Entorno de Desarrollo Integrado*) es un programa que reúne en una sola aplicación las herramientas necesarias para desarrollar software: editor de código, compilador o intérprete, depurador y otras utilidades.

Tres de los más usados:

1. Visual Studio Code (VS Code) – popular por su ligereza y extensiones.
2. IntelliJ IDEA – muy usado en desarrollo Java y Kotlin.
3. Eclipse – común en proyectos Java y de código abierto.

### **Ciclo de vida de un software.**

El ciclo de vida del software es el conjunto de fases que sigue un proyecto desde que se idea hasta que se retira. Las principales etapas son:

1. Análisis de requisitos: se recopilan y definen las necesidades del cliente.
2. Diseño: se planifica la arquitectura y estructura del sistema.
3. Implementación (codificación): los programadores escriben el código.
4. Pruebas: se verifica que el software funcione correctamente.
5. Despliegue: se instala y pone en uso el sistema.
6. Mantenimiento: se corrigen errores y se realizan mejoras o actualizaciones.

#### **Fases en el desarrollo y ejecución del software.**

Análisis: se estudian los requisitos del sistema.

Diseño: se definen la estructura y componentes del software.

Codificación: se escribe el programa según el diseño.

Pruebas: se revisa que funcione sin errores.

Implementación o ejecución: se instala y utiliza el software en su entorno real.

Mantenimiento: se actualiza, corrige y mejora tras su uso.

### **Diferencia entre código fuente, código objeto y código ejecutable.**

- Código fuente: es el texto original escrito por el programador en un lenguaje de programación (por ejemplo, Python, Java o C++).
- Código objeto: es el resultado de compilar el código fuente; contiene instrucciones en lenguaje máquina parciales, aún no listas para ejecutarse.
- Código ejecutable: es el programa final ya enlazado y listo para ejecutarse directamente por el sistema operativo.

### **Diferencia entre algoritmo, pseudocódigo y código**

- Algoritmo: conjunto ordenado de pasos lógicos para resolver un problema. Se expresa en lenguaje natural o con diagramas.
- Pseudocódigo: descripción del algoritmo usando una notación parecida a un lenguaje de programación, pero sin seguir su sintaxis exacta.
- Código: implementación real del algoritmo en un lenguaje de programación que puede ser interpretado o compilado por un ordenador.

### **¿Qué es el DISEÑO ESTRUCTURADO y cuáles son los 3 tipos de construcciones en las que se basa?**

El diseño estructurado es una metodología de programación que organiza el código en módulos y usa estructuras lógicas claras para facilitar la comprensión, mantenimiento y reducción de errores.

Se basa en tres construcciones básicas:

1. Secuencia: ejecución ordenada de instrucciones una tras otra.
2. Selección: ejecución condicional de instrucciones (por ejemplo, *if-else*).
3. Iteración: repetición de un conjunto de instrucciones mientras se cumpla una condición (por ejemplo, *while* o *for*).

**Qué es un lenguaje de programación y clasificación de los lenguajes en función de su nivel de abstracción y según la forma de ejecución. Pon ejemplos. Diferencia entre compilador e intérprete.**

**Lenguaje de programación: conjunto de reglas y símbolos que permiten escribir instrucciones para que un ordenador ejecute tareas específicas.**

Clasificación según nivel de abstracción:

- Bajo nivel: cercano al hardware, difícil para humanos. Ejemplo: ensamblador.
- Alto nivel: fácil de entender y usar. Ejemplo: Python, Java, C++.

Clasificación según forma de ejecución:

- Compilados: el código fuente se traduce completamente antes de ejecutarse. Ejemplo: C, C++.
- Interpretados: se ejecutan línea a línea mediante un intérprete. Ejemplo: Python, JavaScript.

Diferencia entre compilador e intérprete:

- Compilador: traduce todo el programa a código máquina antes de ejecutarlo, generando un archivo ejecutable.
- Intérprete: traduce y ejecuta el código instrucción por instrucción en tiempo real, sin crear un ejecutable.

**Bytecode** es un código intermedio entre el código fuente y el código máquina.

Se genera al compilar un programa y puede ejecutarse en una máquina virtual (como la JVM en Java o la PVM en Python).

Permite que el mismo p¿QUÉ ES BYTECODE?

rograma funcione en distintos sistemas operativos sin recompilarlo.

En la programación orientada a objetos (POO), existen tres conceptos fundamentales: clase, objeto y método. Defínelos y pon un ejemplo práctico en un lenguaje que conozcas.

**ChatGPT Plus**

- **Clase:** plantilla que define las propiedades (atributos) y acciones (métodos) que tendrán los objetos.
- **Objeto:** instancia concreta de una clase; representa un elemento real con sus propios valores.

- **Método:** función definida dentro de una clase que describe el comportamiento de sus objetos.

### Ejemplo en Python:

```
class Coche:

    def __init__(self, marca, color):

        self.marca = marca

        self.color = color

    def arrancar(self):

        print(f"El {self.marca} de color {self.color} ha
arrancado.")

# Crear un objeto

mi_coche = Coche("Toyota", "rojo")

mi_coche.arrancar()
```

### Explicación:

- `Coche` es la **clase**.
- `mi_coche` es un **objeto** de esa clase.
- `arrancar()` es un **método** que realiza una acción sobre el objeto.