

# Conceptos iniciales

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

Entorno que integra editor, compilador/intérprete, depurador y herramientas para programar, con el propósito de optimizar el proceso de programación, evitando que el desarrollador tenga que usar aplicaciones separadas para escribir código, compilarlo, ejecutarlo y depurarlo.

Los más usados serían: Visual Studio Code, IntelliJ IDEA, Eclipse.

## Ciclo de vida del software.

El ciclo de vida del software describe todas las etapas que atraviesa un producto desde que nace la idea hasta que deja de usarse. Marca el camino completo de creación, evolución y retiro del sistema.

## Fases en el desarrollo y ejecución del software.

- **Análisis de requisitos**  
Se identifica qué necesita el usuario y qué debe hacer el sistema.
- **Diseño**  
Se define cómo se va a construir: arquitectura, base de datos, módulos, interfaces, flujos.
- **Implementación (programación)**  
Se escribe el código que materializa el diseño.
- **Pruebas**  
Se verifica que todo funcione bien, sin errores y cumpliendo lo esperado.
- **Despliegue**  
El software se instala, publica o distribuye para que los usuarios lo utilicen.
- **Mantenimiento**  
Se corrigen fallos, se mejoran funciones y se adapta a nuevas necesidades reales.

Una vez ya está programado, el ciclo “operativo” es más simple pero igual de importante:

### Traducción del código (si aplica)

- Lenguajes compilados: el código se convierte a ejecutable.
- Lenguajes interpretados: se analiza y ejecuta línea a línea.
- Lenguajes mixtos: pasa por bytecode y una máquina virtual.
- **Carga en memoria**  
El sistema operativo prepara el programa para ejecutarse.
- **Ejecución**  
Se ejecutan las instrucciones, se gestionan datos, memoria, entradas y salidas.
- **Finalización**  
Al terminar, libera memoria y recursos del sistema.

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

- Código fuente: es lo que escribes en un lenguaje de programación.
- Código objeto: es el resultado parcial tras compilar, instrucciones máquina aún sin enlazar.
- Código ejecutable: archivo final que puede ejecutarse directamente en el sistema.

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

- Algoritmo: pasos lógicos para resolver un problema.
- Pseudocódigo: representación del algoritmo en lenguaje cercano al natural, no ejecutable.
- Código: implementación real en un lenguaje que entiende la máquina.

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

Metodología basada en dividir el programa en módulos lógicos, con control de flujo claro.

Se apoya en tres estructuras:

- Secuencia
- Selección (condiciones)
- Iteración (bucles)

## **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.**

Conjunto de reglas y sintaxis para escribir programas.

Clasificación por nivel:

- Bajo nivel: ensamblador.
- Medio nivel: C.
- Alto nivel: Python, Java, JavaScript.

Clasificación por ejecución:

- Compilados: C, C++.
- Interpretados: Python, JavaScript.
- Mixtos (compilan a bytecode + VM): Java, C#.

## **Diferencia entre compilador e intérprete.**

- Compilador: traduce todo el código a ejecutable antes de correrlo (rápido en ejecución).
- Intérprete: ejecuta línea por línea directamente (más flexible, más lento).

## **¿QUÉ ES BYTECODE?**

El bytecode es un tipo de código intermedio generado por ciertos lenguajes de programación tras compilar el código fuente, pero antes de convertirlo a instrucciones nativas de una máquina física. No es directamente ejecutable por el hardware; en su lugar, se ejecuta en una máquina virtual.

**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.**

### **Clase**

Plantilla o modelo que define atributos y comportamientos comunes para crear objetos. Es la abstracción.

### **Objeto**

Instancia concreta de una clase. Representa un elemento real o conceptual con sus propios valores y estado.

### **Método**

Función definida dentro de una clase que describe una acción o comportamiento que los objetos pueden realizar.