

Capítulo 1: Introducción a Java

1.1. ¿Qué es un lenguaje de programación?

Un **lenguaje de programación** es un sistema de notación diseñado para escribir instrucciones que una computadora puede interpretar y ejecutar. Existen diferentes paradigmas de programación, entre los que se destacan:

1. **Lenguajes de bajo nivel:** Como el lenguaje ensamblador, que se comunica directamente con el hardware y tiene una sintaxis dependiente de la arquitectura del procesador.
2. **Lenguajes de medio nivel:** Como el lenguaje C, que permite una gestión más detallada de los recursos del sistema sin ser tan específico del hardware. Actualmente, Rust, se va posicionando como líder futuro en el segmento.
3. **Lenguajes de alto nivel:** Como Java, Python y C++, que son más comprensibles para los humanos y ofrecen abstracciones que facilitan la programación sin necesidad de conocer detalles específicos del hardware.

Como se puede deducir, el que sea más entendible para los humanos es un sobreesfuerzo para el lenguaje con lo que es por eso que son más lentos, en general, aunque poco a poco han ido mejorando en cuestiones de rendimiento. No obstante a día de hoy (2025, en el momento de redactar estos apuntes), Rust y C++ o C, siguen siendo los más empleados para sistemas embebidos (alta optimización)

4. **Lenguajes interpretados y compilados:** Los compilados (como C y C++) requieren una traducción previa a código máquina, mientras que los interpretados (como Python o JavaScript) se ejecutan directamente sin una fase de compilación previa. Java combina ambas estrategias a través de su Máquina Virtual de Java (JVM).

Ejemplo comparativo entre ensamblador, C y Java

A continuación, se muestra un ejemplo de código que realiza la misma funcionalidad en los tres lenguajes, incrementando un número en una variable.

Código en ensamblador (8086):

```
section .data
    num db 5

section .text
global _start

_start:
    mov al, [num]
    inc al
    mov [num], al

    mov eax, 1
    int 0x80
```

Código en C (con punteros):

```
#include <stdio.h>

int main() {
    int num = 5;
    int *ptr = &num;
    (*ptr)++;
    printf("Valor de num: %d\n", num);
    return 0;
}
```

Código en Java:

```
public class Incremento {
    public static void main(String[] args) {
        int num = 5;
        num++;
        System.out.println("Valor de num: " + num);
    }
}
```

Como se observa, en ensamblador se requiere manipular directamente registros y memoria, en C se usan punteros para acceder y modificar el valor, mientras que en Java el código es más abstracto y sencillo gracias a su alto nivel de abstracción.

1.2. Orígenes de Java

Java fue creado por **James Gosling** y su equipo en **Sun Microsystems** a principios de la década de 1990. Originalmente, se llamaba **Oak** y estaba destinado a ser utilizado en dispositivos electrónicos de consumo. Sin embargo, la posibilidad de ejecutar código en diferentes plataformas sin modificaciones lo llevó a convertirse en un lenguaje clave para el desarrollo de software.

En 1995, Java fue presentado oficialmente y desde entonces ha evolucionado con numerosas versiones y mejoras. Hoy en día es mantenido por **Oracle Corporation** y sigue siendo uno de los lenguajes de programación más utilizados en el mundo.

1.3. Instalación de Java en Windows

Para comenzar a programar en Java, es necesario instalar el **Kit de Desarrollo de Java (JDK)**. Este incluye el compilador **javac**, la JVM y herramientas adicionales para el desarrollo de aplicaciones.

Pasos para instalar el JDK en Windows:

1. Descargar el JDK

- Visita la página oficial de Oracle: <https://www.oracle.com/java/>
- Descarga la última versión del JDK compatible con Windows.

- Actualmente: https://download.oracle.com/java/23/latest/jdk-23_windows-x64_bin.msi

2. Instalar el JDK

- Ejecuta el instalador descargado y sigue las instrucciones en pantalla.
- Selecciona el directorio de instalación (se recomienda dejar la ruta por defecto).

3. Configurar las variables de entorno

- Abre el Explorador de archivos, haz clic derecho en "Este equipo" y selecciona "Propiedades".
- Haz clic en "Configuración avanzada del sistema" y en la pestaña "Opciones avanzadas" selecciona "Variables de entorno".
- En "Variables del sistema", busca **JAVA_HOME** (si no existe, créala) y establece su valor como la ruta de instalación del JDK (por ejemplo, **C:\Program Files\Java\jdk-17**).
- Agrega **C:\Program Files\Java\jdk-17\bin** a la variable **Path**.

4. Verificar la instalación

- Abre una ventana de **Símbolo del sistema** (**cmd**) y ejecuta los siguientes comandos:

```
java -version
javac -version
```

- Deberías ver información sobre la versión instalada de Java y del compilador.

1.4. Configuración de Java en Visual Studio Code

Para programar en Java con **Visual Studio Code (VS Code)**, es necesario instalar algunas extensiones adicionales.

Pasos para configurar Java en VS Code:

1. Instalar Visual Studio Code

- Descarga e instala **VS Code** desde su sitio oficial: <https://code.visualstudio.com/>.

2. Instalar la Extensión de Java

- Abre VS Code y ve al "Marketplace" de extensiones (**Ctrl + Shift + X**).
- Busca e instala **"Extension Pack for Java"**, que incluye:
 - Language Support for Java(TM) by Red Hat
 - Debugger for Java
 - Test Runner for Java
 - Maven for Java
 - IntelliCode for Java

3. Configurar la ejecución de Java en VS Code

- Abre VS Code y crea una nueva carpeta para tu proyecto.
- Dentro de la carpeta, crea un archivo **HolaMundo.java** con el siguiente contenido:

```
public class HolaMundo {  
    public static void main(String[] args) {  
        System.out.println("Hola, mundo!");  
    }  
}
```

- Guarda el archivo y presiona **Ctrl + Shift + P** para abrir la barra de comandos.
- Escribe **Java: Configure Runtime** y selecciona la versión del JDK instalada.
- Para ejecutar el código, abre el terminal en VS Code (**Ctrl + ñ**), y usa:

```
javac HolaMundo.java  
java HolaMundo
```