

Java Profesional

Bloque 1: Introducción

Tatiana Borda

<https://tatianaborda.com/>

<https://www.linkedin.com/in/tatiana-borda/>

<https://www.youtube.com/@AlienExplorer>



Hola soy... 🖐️

¡Hola a todos! Antes de iniciar la clase de hoy me gustaría que me conozcan un poco: **Soy Tatiana Borda algunos me conocen como Alien Explorer**, soy una fullstack dev autodidacta con más de 5 años de experiencia en Java, JavaScript y Solidity. Me apasiona desarrollar aplicaciones web y explorar el mundo Blockchain a través de hackathones. Además, disfruto mucho de enseñar y compartir tips de programación en mi canal de YouTube, AlienExplorer. Soy curiosa y estoy siempre buscando nuevos desafíos!



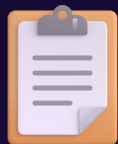


Tabla de contenido

/01

Bienvenida

/03

Cómo se ejecuta un programa en Java?

/05

Tu primer programa

/02

Que es Java?

/04

Instalación y configuración del entorno

/06

Qué vas a aprender?

/02

Qué es Java y por qué aprenderlo?

Java es un lenguaje de programación compilado, fuertemente tipado y orientado a objetos, diseñado para ser portable, concurrente y seguro.

Además, Java se basa en el principio de **“write once, run anywhere”** esto significa que podés escribir un programa en Java y ejecutarlo en cualquier sistema operativo, gracias a la Máquina Virtual de Java (JVM)



Un poco un historia:

Java nació en 1991 como un proyecto llamado Oak, desarrollado por el "Equipo Verde" en Sun Microsystems, liderado por James Gosling. Su idea era crear un lenguaje orientado a objetos, más simple que C/C++, pero con una sintaxis similar para que fuera fácil de adoptar.

Inicialmente, Java fue pensado para programar televisores inteligentes y decodificadores. Pero ese mercado no estaba listo todavía. En cambio, Java encontró su lugar en el desarrollo para Internet, que crecía rápidamente en los años 90.



Un poco un historia:

En 1995, Oak cambió su nombre a Java (el nombre "Oak" ya estaba registrado). Según Gosling, el nombre se inspiró en el café Java que tomaba mientras programaba.

En 1996, Sun lanzó Java 1.0 con la promesa de "Write Once, Run Anywhere": escribir el código una sola vez y ejecutarlo en cualquier plataforma. Esa portabilidad, junto con mayor seguridad y accesibilidad, convirtió a Java en un estándar de facto en la programación para la web y aplicaciones empresariales.



Qué podés desarrollar con JAVA?



- Desarrollo Empresarial
- Desarrollo Mobile Android
- Big Data y Ciencia de Datos
- Sistemas Embebidos e IoT
- Desarrollo de juegos
- Cloud Computing y DevOps
- Cyber Seguridad
- Arte generativo
- Herramientas de desarrollo
- Aplicaciones Desktop

Principales features de Java

- * Sintaxis clara y estructurada
- * Programación Orientada a Objetos (OOP)
- * Gran ecosistema de librerías y herramientas
- * Seguridad, robustez y rendimiento
- * Soporte para programación funcional (desde Java 8)
- * Actualizaciones frecuentes y backward compatibility



```
androidx.appcompat.app.AppCompatActivity extends AppCompatActivity
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}
public void onClick(View view) {
    Intent i = new Intent(this, MainActivity.class);
    startActivity(i);
}
```


Por qué Java?

Aprender Java hoy es una inversión sólida porque es:

- *Conocido como uno de los lenguajes de programación más versátiles del mundo
- *Estable y maduro, con más de 25 años de desarrollo.
- *Multiplataforma.
- *Ampliamente usado en el mercado laboral (especialmente en backend y grandes sistemas).
- *Compatible con herramientas modernas y arquitecturas como microservicios



```
android.support.design.widget.Snackbar
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}
public void onClick(View view) {
    Intent i = new Intent(this, MainActivity.class);
    startActivity(i);
}
```

/03

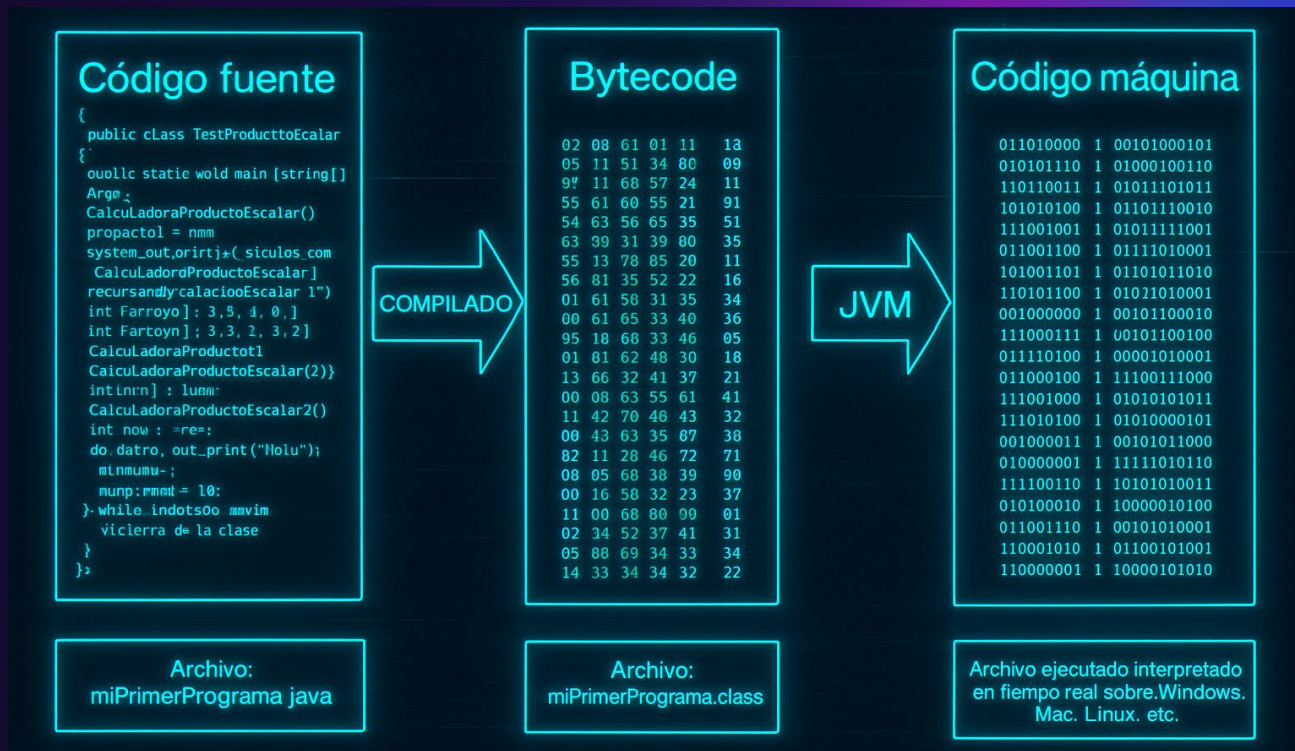
Cómo se ejecuta un programa en Java?

Para que un programa en Java funcione, necesitamos entender tres conceptos clave: el código fuente, la compilación y la ejecución.

- **Código fuente (.java):** es lo que escribimos como desarrolladores
- **Compilador (javac):** transforma ese archivo en bytecode, un formato intermedio que entiende la JVM
- **JVM (Java Virtual Machine):** interpreta ese bytecode y lo ejecuta en cualquier sistema operativo (Windows, Linux, Mac Os)



El proceso:



JVM, Java Virtual Machine:

La JVM, Máquina Virtual de Java, es el componente central del ecosistema de Java que permite ejecutar aplicaciones escritas en JAVA*. Al ser un lenguaje compilado Es una máquina virtual que traduce el bytecode (el código intermedio que genera el compilador de Java) en instrucciones entendibles por el sistema operativo y el hardware donde se está ejecutando.

*Existen otra máquinas virtuales para otros lenguajes compilados



Funcionamiento de la JVM:

1. Compilación del código fuente:

Cuando escribimos un programa en Java (archivo .java), este es compilado por el compilador de Java (javac) en bytecode (archivo .class).

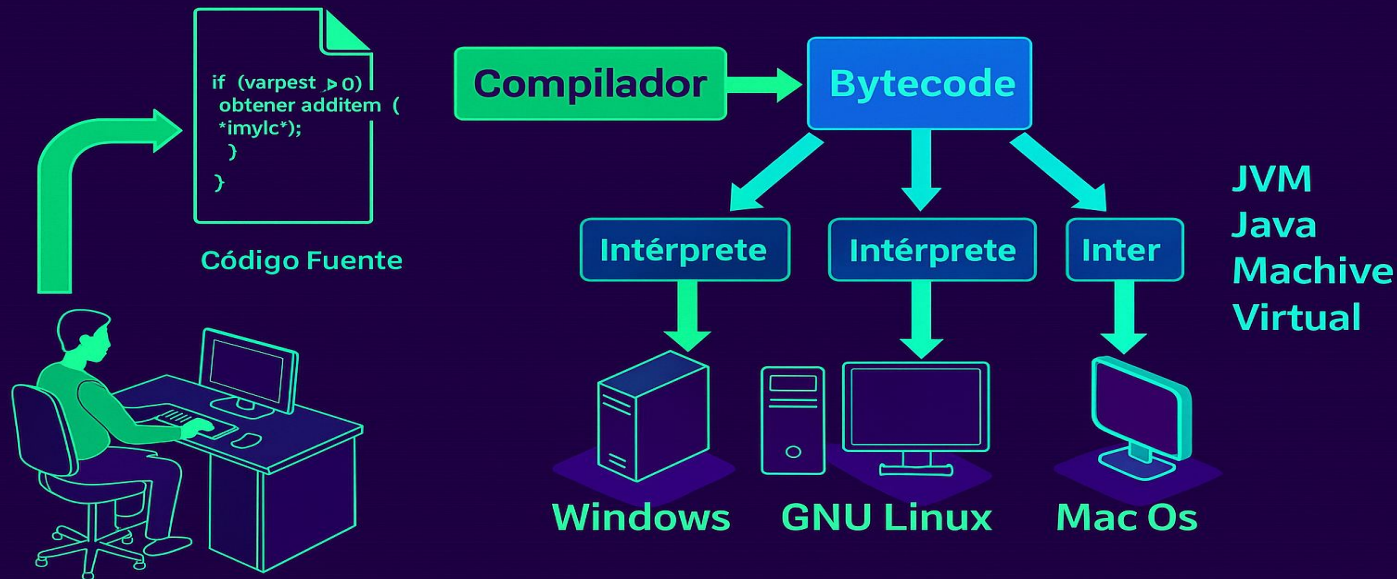
El bytecode es el mismo sin importar el sistema operativo o hardware.

2. Ejecución del bytecode:

La JVM interpreta o compila este bytecode y lo ejecuta en el hardware subyacente.



Funcionamiento de la JVM:



Java Development Kit:

Para poder hacer todo este proceso que permite que Java sea tan flexible y multiplataforma, necesitamos instalar el JDK (Java Development Kit), que incluye:

- *El compilador javac
- *La JVM
- *Herramientas para desarrollar y ejecutar nuestros programas
- *JRE: java runtime environment

Más adelante vamos a instalarlo juntos, pero por ahora es importante que entiendas esta idea: cuando escribís en Java, no estás generando directamente un programa ejecutable, sino algo que va a correr sobre la JVM



/04

Instalación y configuración del entorno

*JDK (Java Development Kit): es lo primero. En este curso vamos a usar la versión LTS más reciente, que al momento de grabar es Java 21 esto incluye el JRE- java runtime enviroment o entorno de ejecución.

*Editor de código o IDE: vamos a descargar IntelliJ IDEA, muy usado profesionalmente en proyectos Java

*Terminal o línea de comandos, para compilar y ejecutar tus archivos

En este video te muestro paso a paso cómo descargar, instalar y verificar que todo está funcionando, vamos!



/05

Tu primer programa



/06

Qué vas a aprender?

A lo largo del curso vamos a ir avanzando en bloques temáticos:

- los fundamentos: variables, operadores, condicionales, ciclos.
- funciones y métodos.
- programación orientada a objetos: clases, herencia, encapsulamiento.
- colecciones como listas y mapas.
- características modernas de Java como lambdas, var, y pattern matching

Todo esto lo vamos a aplicar paso a paso en un mini proyecto práctico: un gestor de tareas por consola. Cada módulo va a sumar una nueva funcionalidad al programa, para que no solo entiendas los conceptos, sino también los veas funcionando en algo real.

