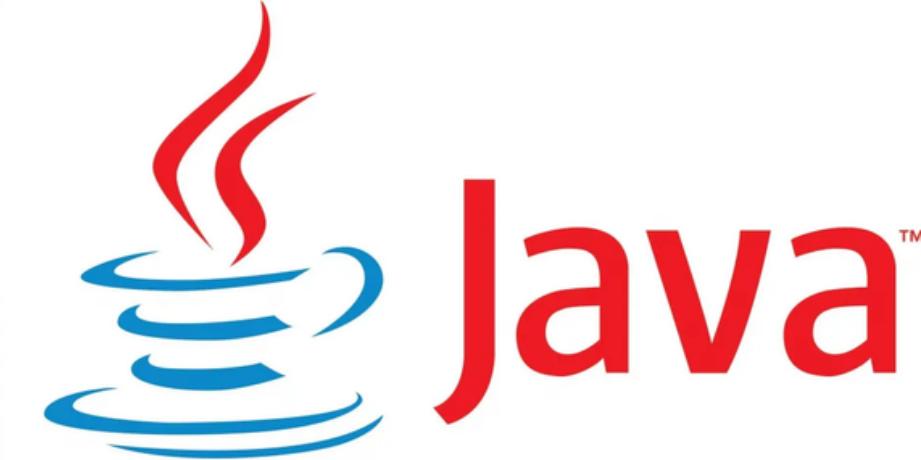


El lenguaje JAVA



Presentado por
Equipo 3

Danna Paola Ruiz Hernández 2064501 IAS
Alexis Francisco Mendoza Vázquez 2107087 ITS
Johan Emilio Vazquez Luna 2092470 ITS
Brayan Alexis López Guzmán 2035619 IAS

¿Qué es JAVA?



Java es un lenguaje de programación y una plataforma informática versátil y de propósito general, utilizada para crear software y aplicaciones en una amplia gama de dispositivos y sistemas operativos. Es conocido por ser multiplataforma, seguro, confiable y para desarrolladores de aplicaciones web, móviles, empresariales y de macrodatos.

Enfatiza su lema: "Write Once, Run Anywhere" (WORA). Explica que esto significa que el código compilado en Java puede ejecutarse en cualquier dispositivo que tenga una Máquina Virtual Java (JVM), sin necesidad de recompilarlo.



Orígenes de Java



Java fue creado en 1990 por James Gosling y su equipo en Sun Microsystems dentro del Proyecto Green. Inicialmente llamado Oak, fue diseñado para funcionar en diferentes dispositivos sin depender del hardware. En 1995 se renombró como Java y se lanzó oficialmente con el lema:
“Escribe una vez, ejecútalo en cualquier lugar”



Historia de Java



1990 - El inicio del proyecto Green

James Gosling, Mike Sheridan y Patrick Naughton, ingenieros de Sun Microsystems, comienzan un proyecto para desarrollar un lenguaje para dispositivos electrónicos domésticos. El cual se iba a llamar originalmente OAK



1992 - Creación de Oak

James Gosling desarrolla el primer prototipo del lenguaje Oak, diseñado para ser portátil y seguro. Se usaba en un dispositivo llamado Star7, con una interfaz gráfica simple

Historia de Java



1994 - Nace la idea de usarlo en la web

El equipo de Sun se da cuenta de que Oak podía usarse para crear aplicaciones web interactivas, algo revolucionario en esa época.



1995 - Nace oficialmente Java

El lenguaje se renombra a Java, ademas se presenta Java 1.0 que gracias a esto se presento el primer navegador con soporte JAVA

Historia de Java



1996 - Lanzamiento oficial de Java

1.0

Sun Microsystems libera la primera versión oficial de Java Development Kit y se empieza a popularizar gracias a los applets



1997 - Java 1.1

Se mejoran las bibliotecas, se agrega la API JDBC para conexión a bases de datos y se simplifica el manejo de eventos.

Historia de Java



● ● ●
1998 - Nace Java 2 (J2SE 1.2)

Gran actualización del lenguaje. Se divide en tres ediciones:

J2SE (Standard Edition): para aplicaciones de escritorio.

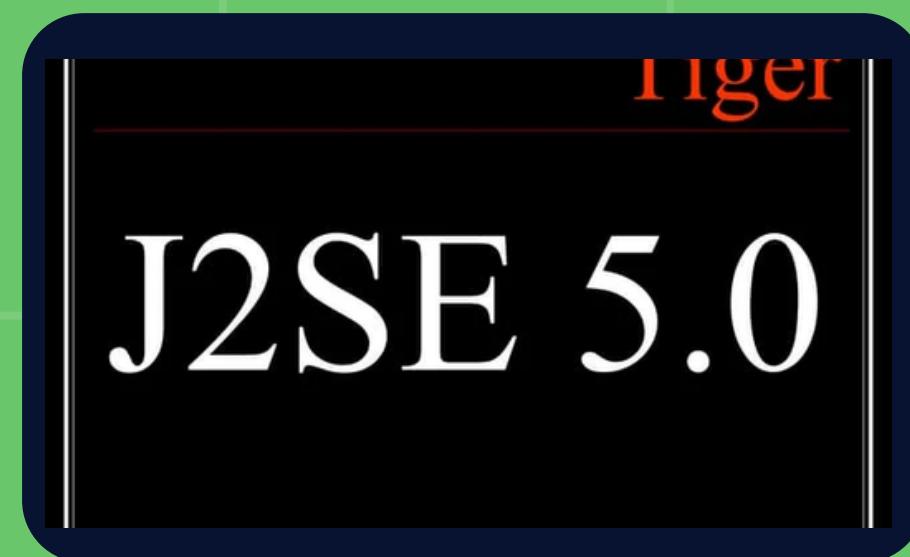
J2EE (Enterprise Edition): para aplicaciones empresariales.

J2ME (Micro Edition): para dispositivos móviles y embebidos.

● ● ●
2000 - Expansión móvil y empresarial

Java se vuelve el lenguaje base para muchos teléfonos móviles y servidores empresariales.

Historia de Java



2004 - Java 5

Se agregan nuevas características como genéricos, autoboxing, enumeraciones y bucle for mejorado.



2006 - Java 6

Se mejora el rendimiento, la integración con web services y el soporte para scripting.

Historia de Java



2009 - Oracle compra Sun

Microsystems

Oracle Corporation adquiere Sun y pasa a ser el propietario oficial de Java.



2011 - Java 7

Oracle lanza Java 7 con mejoras en rendimiento, manejo de recursos y soporte para lenguajes dinámicos.

Historia de Java



2014 - Java 8

Versión muy popular.

Introduce expresiones lambda, API de Streams y la nueva API de fecha y hora.



2017 - Java 9

Se introduce el Sistema de Módulos (Project Jigsaw), que permite dividir el JDK en módulos más pequeños.

A partir de aquí, Java adopta un ciclo de actualizaciones cada 6 meses.

Historia de Java



2018 - Java 10 y 11

Java 11 se convierte en la nueva versión de soporte a largo plazo (LTS). Se añade la palabra clave var y mejoras en el recolector de basura.



2021 - Java 17

Nueva versión LTS, más estable y moderna. Mejora la seguridad, rendimiento y compatibilidad con nubes y contenedores.

Historia de Java



2023 - Java 21

Última versión LTS (hasta ahora). Incluye mejoras como patrones de coincidencia, registros, text blocks, y Virtual Threads.



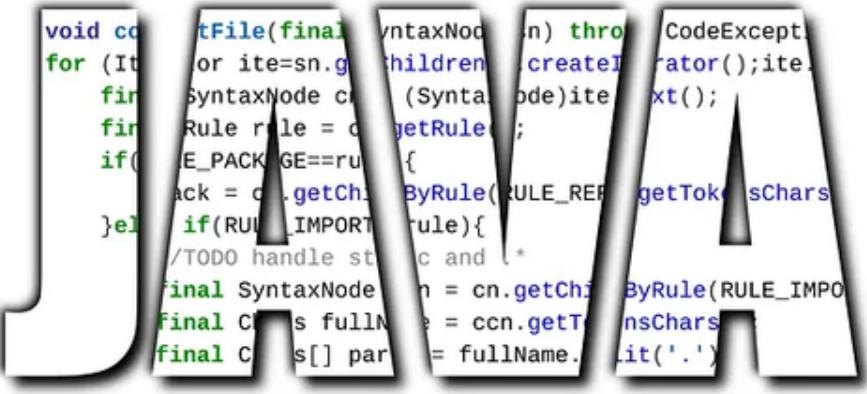
Actualidad (2025)

Java sigue siendo uno de los lenguajes más usados del mundo, especialmente en aplicaciones empresariales, backend web y sistemas Android.

Características: principales



- Orientado a Objetos: Java se basa en el paradigma de programación orientada a objetos, lo que significa que todo se organiza en clases y objetos, promoviendo la reutilización de código y la modularidad.



- Independiente de la Plataforma: Gracias a la Máquina Virtual de Java (JVM), el código Java puede ejecutarse en cualquier dispositivo o sistema operativo que tenga una JVM instalada, siguiendo el lema "escribe una vez, ejecuta en cualquier lugar".

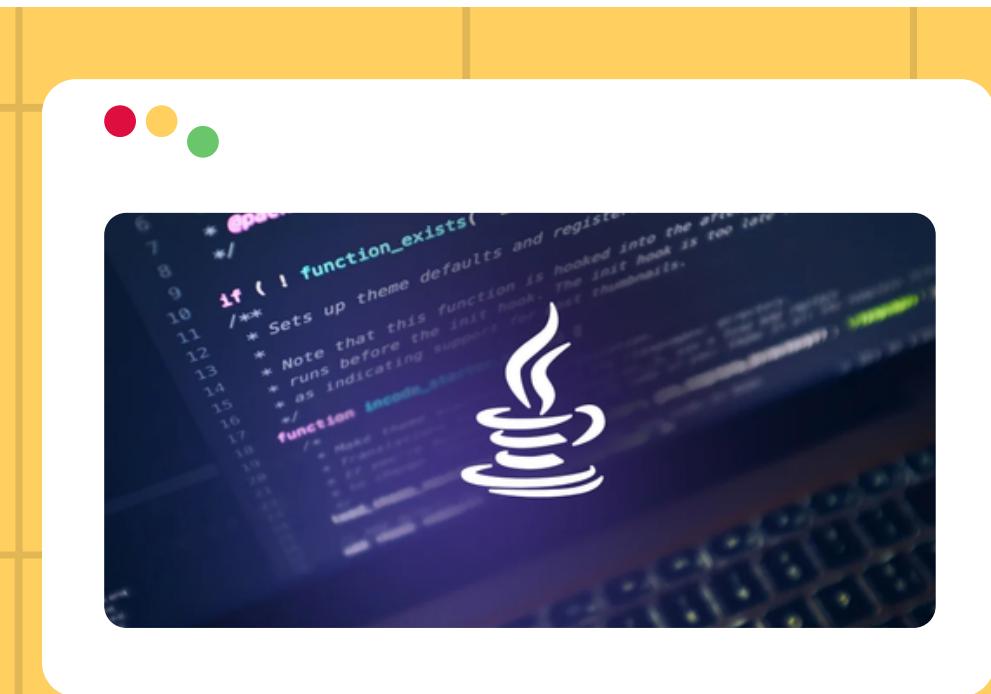
- Robusto y Seguro: Java incluye manejo de excepciones, recolección automática de basura (garbage collection) y verificación de tipos en tiempo de compilación, lo que reduce errores. Además, tiene características de seguridad como la ejecución en un entorno aislado (sandbox) para applets.



- Multihilo: Java soporta la programación concurrente, permitiendo que múltiples hilos (tareas) se ejecuten simultáneamente, ideal para aplicaciones que requieren alto rendimiento o multitarea.

Características principales

- Sintaxis Simple y Familiar: Su sintaxis está inspirada en C y C++, lo que la hace fácil de aprender para quienes ya conocen estos lenguajes, aunque elimina complejidades como los punteros.
- Alto Rendimiento: Aunque es un lenguaje interpretado, la JVM optimiza el rendimiento mediante la compilación en tiempo de ejecución (Just-In-Time Compilation, JIT), acercándose al rendimiento de lenguajes compilados.
- Gran Comunidad y Ecosistema: Java cuenta con una enorme comunidad de desarrolladores, una amplia biblioteca estándar (Java API) y frameworks populares como Spring y Hibernate, que facilitan el desarrollo de aplicaciones.
- Portabilidad y Escalabilidad: Java es ideal para aplicaciones de todo tamaño, desde aplicaciones móviles (Android) hasta sistemas empresariales complejos, gracias a su diseño flexible.



¿Cómo funciona? (JVM, JRE, JDK)



1. JDK (Java Development Kit)

- **Qué es:** El kit de desarrollo de Java, es decir, todo lo que necesita un programador para crear aplicaciones Java.
- **Contiene:**
- El JRE
- El compilador javac (convierte código fuente .java a bytecode .class)
- Herramientas para depurar, documentar y ejecutar programas
- **Función:** Permite desarrollar y compilar programas Java.



JDK

Development Tools : java, javac, javadoc, jar

¿Cómo funciona? (JVM, JRE, JDK)



2. JRE (Java Runtime Environment)

- **Qué es:** El entorno de ejecución de Java.
- **Contiene:**
- La JVM
- Librerías estándar de Java
- **Función:** Permite ejecutar programas Java ya compilados (no sirve para desarrollarlos).



JRE

Runtime Libraries, Byte Code Verifier, Libraries

3. JVM (Java Virtual Machine)

- **Qué es:** La máquina virtual de Java, el componente que realmente ejecuta el código Java.
- **Función:**
 - Toma el bytecode (.class) generado por el compilador
 - Lo interpreta y ejecuta en cualquier sistema operativo
- **Ventaja:** Gracias a la JVM, Java es multiplataforma.



JVM

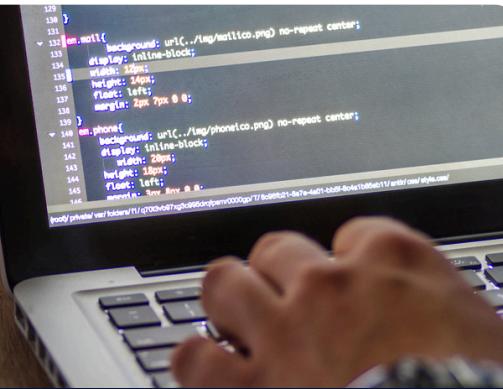
Java Interpreter, JIT, Garbage Collector

Ejemplo del lenguaje



1. Clase → El contenedor principal
2. main → Donde empieza el programa
3. Variables → Donde guardamos información
4. System.out.println() → Para mostrar cosas
5. if → Para tomar decisiones
6. for → Para repetir cosas

```
public class EjemploSimple {  
    Run | Debug  
    public static void main(String[] args) {  
        // Variables  
        String nombre = "Ana";  
        int edad = 25;  
  
        // Mostrar información  
        System.out.println("Nombre: " + nombre);  
        System.out.println("Edad: " + edad);  
  
        // Verificar si es adulto  
        if (edad >= 18) {  
            System.out.println(nombre + " es adulto");  
        }  
  
        // Contar del 1 al 3  
        for (int i = 1; i <= 3; i++) {  
            System.out.println("Número: " + i);  
        }  
    }  
}
```



Gracias

