

Introducción al JDK (Java Development Kit)



■ Guía sobre el JDK de Java

Introducción

Bienvenidos a esta lección sobre el **Java Development Kit (JDK)**, el conjunto de herramientas esenciales para desarrollar aplicaciones en Java. En esta guía, exploraremos su importancia, sus componentes principales y la mejor versión para trabajar actualmente.

Pasos para Ejecutar un Programa en Java:

1. **Instalar el JDK:** Necesitas el JDK para compilar y ejecutar programas Java.
2. **Configurar un IDE o Editor:** Un buen entorno de desarrollo hace que escribir y depurar código sea más fácil, como IntelliJ IDEA o Apache Netbeans.
3. **Escribir el Código:** Escribe tu programa en un archivo `.java`.
4. **Compilar el Código:** Usa el comando `javac` para compilar el archivo.
5. **Ejecutar el Programa:** Usa el comando `java` para ejecutar el programa compilado.

Datos Importantes

- **El JDK** incluye todas las herramientas necesarias para desarrollar y ejecutar programas Java.
- **Un IDE** como IntelliJ IDEA, Apache Netbeans o Eclipse proporciona herramientas adicionales para facilitar el desarrollo.

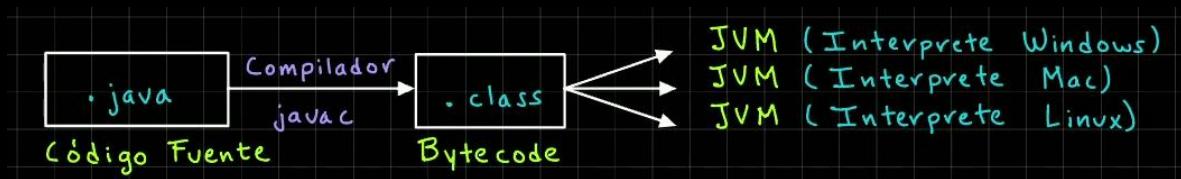
- **Compilar y Ejecutar:** Estos son los dos pasos clave para transformar tu código en una aplicación en funcionamiento.

¿Qué es un Compilador y por qué se necesita para trabajar con Java?

Un **compilador** es una herramienta que traduce el código fuente escrito por los programadores en un lenguaje de alto nivel (como Java) a un lenguaje de bajo nivel o código máquina que puede ser ejecutado directamente por el sistema operativo del computador. Este proceso es crucial porque las computadoras no entienden el código de alto nivel directamente; necesitan instrucciones en un formato que puedan ejecutar.

Por qué se Necesita un Compilador en Java

En el caso de Java, el compilador (llamado `javac`) convierte el código fuente Java (`.java`) en bytecode (`.class`). El bytecode es un formato intermedio que es independiente de la plataforma y puede ser ejecutado en cualquier sistema operativo que tenga una Máquina Virtual Java (JVM).



¿Qué es el JDK?

El **Java Development Kit (JDK)** es un paquete de herramientas necesarias para desarrollar aplicaciones en Java. Incluye:

1. **Compilador (`javac`):** Traduce el código fuente Java a bytecode.
2. **Java Runtime Environment (JRE):** Proporciona las bibliotecas y la JVM necesarias para ejecutar el bytecode.
3. **Herramientas de Desarrollo:** Incluye herramientas como el depurador (`jdb`), el empaquetador (`jar`), entre otras.

¿Qué es el JRE?

El **Java Runtime Environment (JRE)** es un subconjunto del JDK. Es todo lo que necesitas para ejecutar programas Java, pero no para desarrollarlos. Incluye:

1. **JVM (Java Virtual Machine):** Ejecuta el bytecode Java.
2. **Bibliotecas de Clase:** Conjunto de bibliotecas necesarias para ejecutar aplicaciones Java.
3. **Otros Componentes:** Archivos de configuración, bibliotecas nativas, etc.

Relación entre JDK y JRE

- **JDK:** Incluye el JRE más las herramientas de desarrollo necesarias (como el compilador).
- **JRE:** Incluye solamente lo necesario para ejecutar aplicaciones Java.

◆ ¿Qué versión del JDK usar?

Aunque existen versiones más recientes del JDK que se liberan cada 6 meses, recomendamos utilizar la versión **JDK 21**, ya que es la versión **LTS (Long-Term Support)** más reciente.

◆ ¿Qué es una versión LTS?

Una versión LTS es una **versión de soporte extendido**, lo que significa que recibirá actualizaciones de seguridad y correcciones de errores por varios años. Esto la hace ideal para proyectos a largo plazo y aplicaciones empresariales.

🔧 ¿Cómo funciona el JDK?

Cuando trabajamos con Java, el proceso se divide en dos fases principales:

1. Compilación:

- Se escribe el código en un archivo `.java`.
- El compilador (`javac`) lo convierte en bytecode y genera un archivo `.class`.

2. Ejecución:

- La JVM toma el bytecode y lo interpreta (ejecuta) en el sistema operativo correspondiente.
- Gracias a esto, Java es **independiente de la plataforma** (puede ejecutarse en Windows, Mac y Linux sin modificaciones en el código).

📌 Diferencia entre JDK y JRE

Característica	JDK (Java Development Kit)	JRE (Java Runtime Environment)
Incluye un compilador	✓ Sí	✗ No
Contiene herramientas de desarrollo	✓ Sí	✗ No
Permite ejecutar aplicaciones Java	✓ Sí	✓ Sí
Se usa para programar en Java	✓ Sí	✗ No

💡 **Conclusión:** Si solo deseas ejecutar aplicaciones Java, el **JRE** es suficiente. Pero si quieres desarrollar en Java, necesitas el **JDK**, por lo tanto, es lo que instalaremos en nuestro curso de Universidad Java.

🛠 Componentes del JDK

Al instalar el JDK, obtienes:

1. **Compilador** (`javac`) – Traduce código Java a bytecode.
2. **JVM (Java Virtual Machine)** – Interpreta y ejecuta el bytecode.
3. **Bibliotecas estándar** – Clases y métodos esenciales de Java.

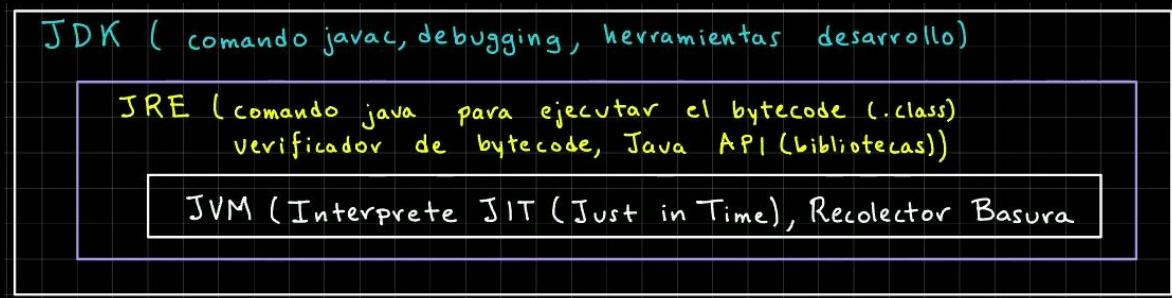
4. **Herramientas de desarrollo** – Depuradores, empaquetadores, etc.
5. **JIT (Just-In-Time Compiler)** – Optimiza el rendimiento del código en tiempo de ejecución.

¿Qué obtienes al instalar el JDK y JRE?

Cuando instalas el JDK, también obtienes el JRE como parte del paquete. Esto significa que al instalar el JDK, tendrás todas las herramientas necesarias tanto para desarrollar como para ejecutar programas Java.

Resumen

1. **Compilador**: Convierte el código fuente Java en bytecode que puede ser ejecutado por la JVM.
2. **JDK**: Incluye el compilador, el JRE, y otras herramientas de desarrollo.
3. **JRE**: Incluye la JVM y las bibliotecas necesarias para ejecutar aplicaciones Java.
4. **JVM**: Ejecuta el bytecode Java y permite la independencia de plataforma.



El intérprete JIT es una pieza clave en la JVM que permite a las aplicaciones Java ejecutar de manera eficiente y optimizada en tiempo de ejecución. Mediante la conversión dinámica de bytecode a código máquina nativo, el JIT mejora significativamente el rendimiento de las aplicaciones Java mientras mantiene la portabilidad y flexibilidad del lenguaje.

📌 Conclusión

- El **JDK** es esencial para programar en Java.
- La versión **JDK 21 (LTS)** es la mejor opción actualmente.
- Java sigue siendo uno de los lenguajes más potentes y versátiles para desarrollo web, móvil y empresarial.

📌 En la siguiente lección aprenderemos a configurar nuestro entorno de desarrollo y comenzar a escribir código en Java. ¡Nos vemos en la próxima lección! 🚀

¡Continuemos, el esfuerzo vale la pena

Ing. Ubaldo Acosta

Fundador de [GlobalMentoring.com.mx](https://www.globalmentoring.com.mx)