



Recibe una cálida:

¡Bienvenida!

Te estábamos esperando 😊 +

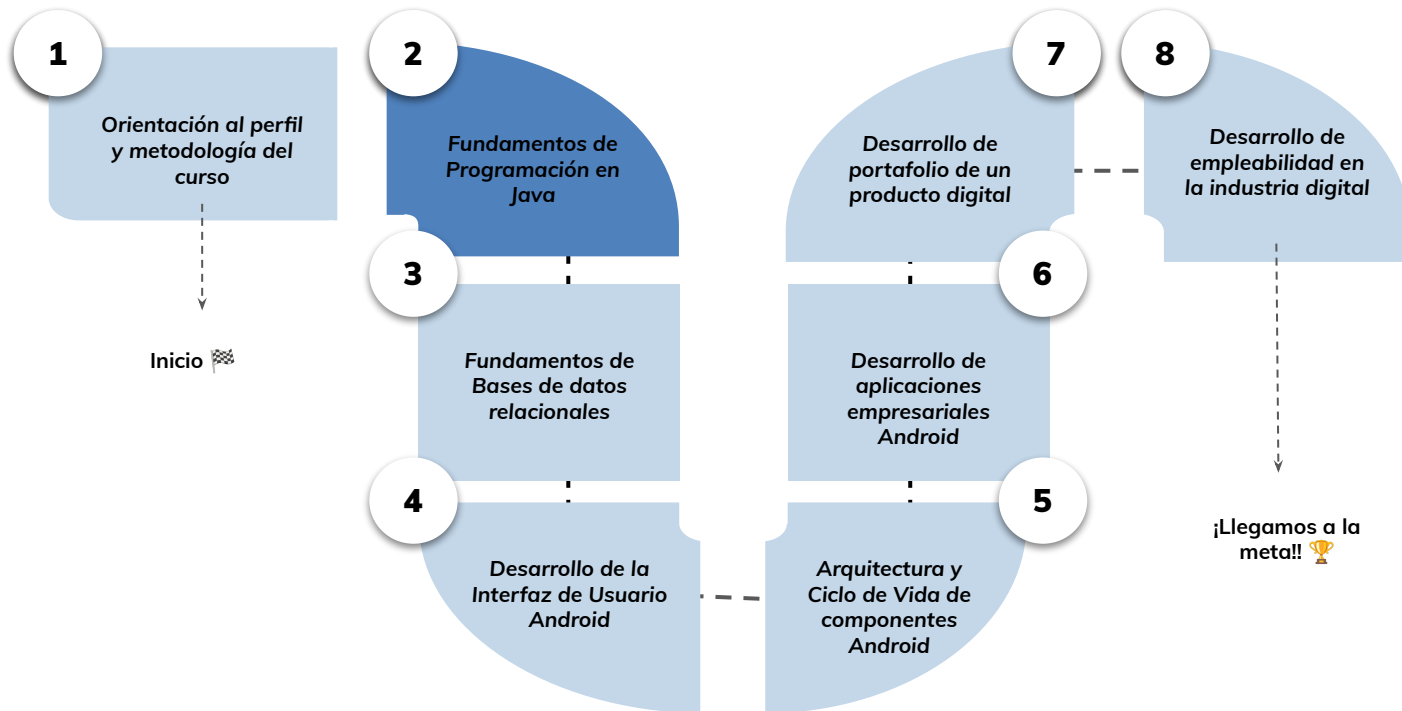


› El Lenguaje Java y su entorno

AE1.1: Analizar las características fundamentales del lenguaje java para el desarrollo de aplicaciones empresariales.

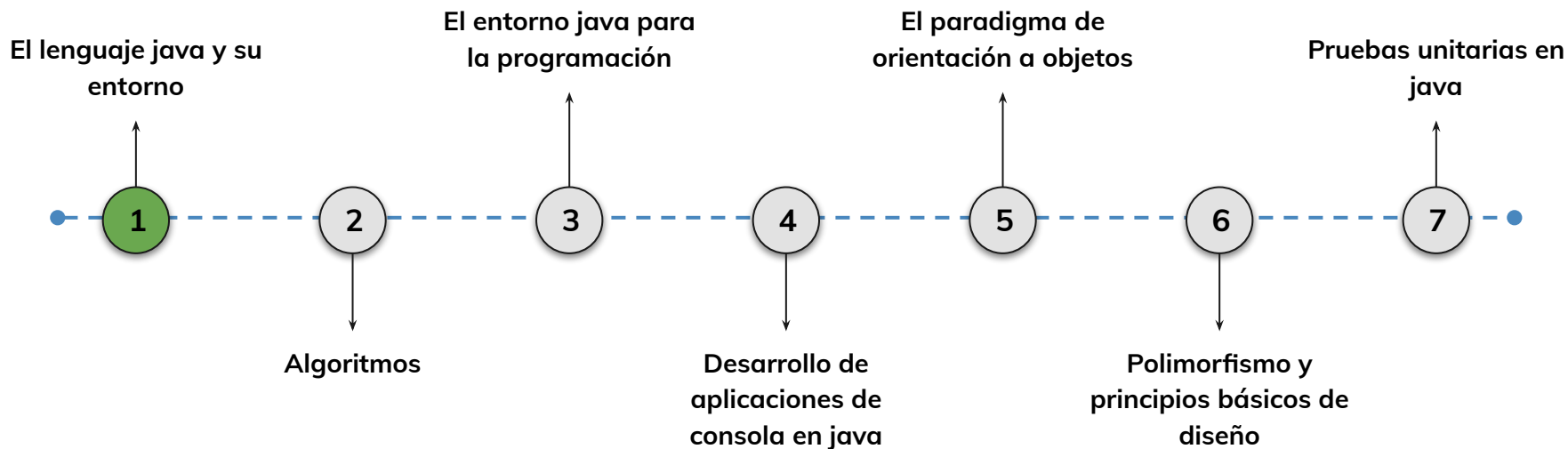
Hoja de ruta

¿Cuáles **módulos** conforman el programa?



Roadmap de lecciones

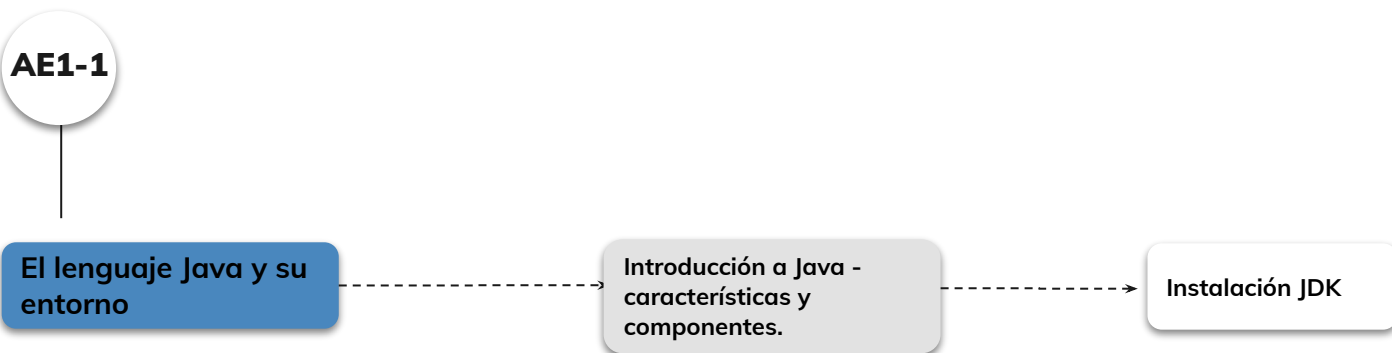
¿Cuáles **lecciones** estaremos estudiando en este módulo?





Learning Path

¿Cuáles temas trabajaremos hoy?



Reconocer las características fundamentales del lenguaje Java para el desarrollo de aplicaciones.



Objetivos de aprendizaje

¿Qué aprenderás?



- Comprender el concepto de Java y sus características.
- Entender la implementación de los componentes JVM, JRE y JDK.



Rompehielo



Compartamos experiencias:

- ¿Qué aplicaciones o software conocen que estén desarrolladas con Java?
- ¿Qué imaginas que podrás crear al aprender a programar en Java?

Les recomendamos abrir el mic 🎤 o escribir por el chat 😊

¿Que es Java?



Introducción a Java

- Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases. Fue creado por Sun Microsystems en 1995 y actualmente es propiedad de Oracle.
- Este lenguaje se considera fuertemente tipado (también llamado de tipado estático).



Características de Java



Algunas características distintivas de este lenguaje:

- **Portable:** El código Java se ejecuta en cualquier plataforma que tenga instalada la Java Virtual Machine (JVM).
- **Orientado a Objetos:** Usa el paradigma de Programación Orientada a Objetos.
- **Seguro:** Tiene un sistema de seguridad para evitar ejecución de código malicioso.
- **Multiplataforma:** Permite crear aplicaciones multiplataforma.
- **Amplio soporte:** Cuenta con gran cantidad de bibliotecas y frameworks.



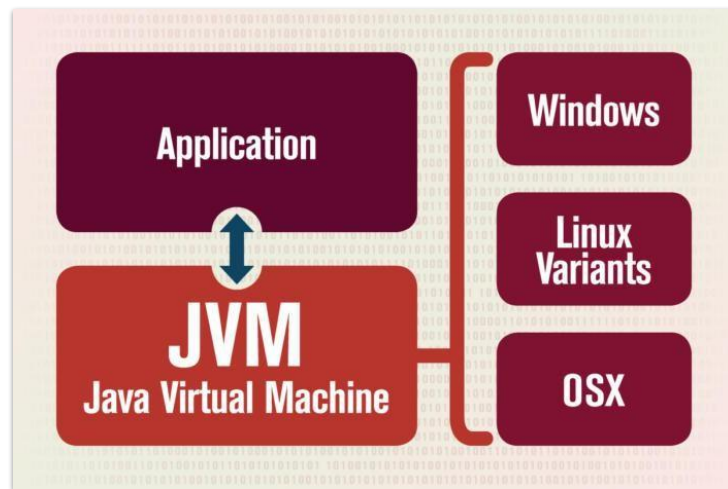
Java Virtual Maching

¿Qué es la JVM?:

La **JVM** es una máquina virtual que ejecuta bytecode Java. Proporciona una capa de abstracción entre el código y el hardware.

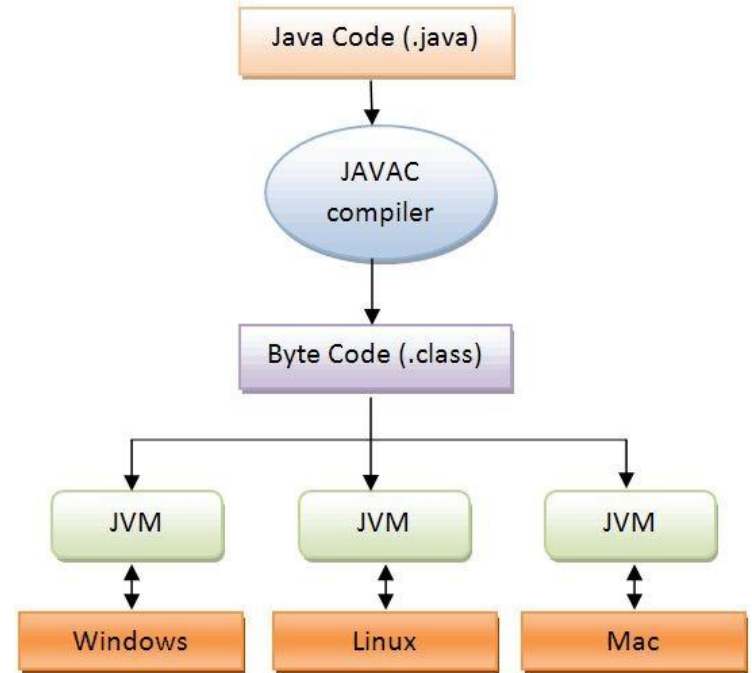
Permite la portabilidad de Java entre diferentes plataformas.

Gestiona automáticamente la memoria y proporciona seguridad.



Bytecode

- Es un código en formato binario, generado por el compilador Java al compilar el código fuente. También se le conoce como "código máquina virtual Java".
- Es interpretado y ejecutado por la JVM.
- Es independiente de plataforma, permite la portabilidad entre sistemas.



JDK y JRE

JDK: Java Development Kit

El Kit de Desarrollo de Java es un conjunto de herramientas para desarrollar aplicaciones Java.

Incluye el **compilador** javac, las bibliotecas de clases de Java (**API** de Java), entre otras herramientas.

Permite **compilar** código fuente Java a **bytcodg**.

Java Runtime Environment (JRE)

El Entorno de Ejecución de Java proporciona el entorno necesario para **sjsuñar** aplicaciones Java compiladas.

Contiene la Java Virtual Machine (**JVM**), las bibliotecas de clases de Java y otros componentes necesarios.

No incluye el compilador javac ni otras herramientas de desarrollo.



Usos y aplicaciones de Java



Algunas aplicaciones de este lenguaje:

- Desarrollo de aplicaciones móviles y web
- Aplicaciones empresariales y Big Data.
- Machine Learning
- Desarrollo de videojuegos.
- Internet de las Cosas (IoT).



Versiones de Java



Algunos hitos importantes en diferentes versiones Java:

- Java 1.0 (1996) - Primera versión estable de Java. Introdujo conceptos centrales como la JVM.
- Java 1.1 (1997) - Mejoras en AWT, introducción de JavaBeans, JDBC.
- Java 2 (1998) - Renombrado a Java 1.2. Mejoras en rendimiento y API de colecciones.
- Java 6 (2006) - Mejoras en la plataforma, internacionalización.
- Java 8 (2014) - Programación funcional con Lambdas.
- Java 11 (2018) - Finalización del ciclo de lanzamientos de 6 meses. LTS.
- Java 17 (2021) - Versión LTS actual. Enfoque en estabilidad y rendimiento.



Versiones de Java



Algunos hitos importantes en diferentes versiones Java:

- **Java 18 (2022)** - Introducción de mejoras en rendimiento y características experimentales como la nueva API de código byte y la optimización de la JVM.
- **Java 19 (2022)** - Nuevas características experimentales, optimización de la JVM, e introducción de Project Loom y mejoras en el rendimiento.
- **Java 20 (2023)** - Mejora en la eficiencia de las APIs, optimización de la JVM, y nuevas características de concurrencia y manejo de seguridad.
- **Java 21 (2023) (LTS)** - Versión LTS con mejoras en el rendimiento, nuevas herramientas para desarrolladores, y optimización de seguridad y patrones de coincidencia. -> **Esta versión utilizaremos en este curso**
- **Java 22 (2024)** - Enfoque en la mejora de la eficiencia de las APIs, seguridad, y optimización de rendimiento. Nuevas características experimentales.
- **Java 23 (2024)** - Optimización avanzada de la JVM, mejoras en concurrencia, desarrollo modular y nuevas herramientas para desarrolladores.
- **Java 24 (2025)** - Continuación de innovaciones en rendimiento, concurrencia, y modularidad. Mejoras en la seguridad y estabilidad de la plataforma.



LIVE CODING

Conocer los componentes de Java

Para trabajar con Java, se necesita:

- **JDK (Java Development Kit):** El conjunto de herramientas necesarias para desarrollar en Java.
 - Incluye el compilador `javac` y la JVM para ejecutar programas.
- **JRE (Java Runtime Environment):** Es un entorno que permite ejecutar programas Java (pero no compilar).
- **IDE (Entorno de Desarrollo Integrado):** Herramientas como Eclipse, IntelliJ IDEA o NetBeans hacen que sea más fácil escribir, compilar y depurar programas en Java.

.

LIVE CODING

Instalar lo necesario para preparar el entorno de ejecución: JDK y Eclipse IDE

Vamos a instalar todo lo necesario para programar en Java:

Paso 1: Instalar JDK

1. Ve a la página oficial de Java: [Descargar JDK](#) o puedes usar OpenJDK (una alternativa de código abierto).
2. Descarga la versión más reciente de JDK compatible con tu sistema operativo (Windows, macOS o Linux).
3. Durante la instalación, asegúrate de agregar el JDK al **PATH** del sistema para que puedas ejecutarlo desde la línea de comandos.

LIVE CODING

Paso 2: Instalar Eclipse IDE

1. Ve a la página oficial de Eclipse: Eclipse IDE.
2. Descarga la versión "Eclipse IDE for Java Developers".
3. Instala Eclipse siguiendo los pasos proporcionados por el instalador..
4. **Crear y ejecutar tu primer proyecto Java**

Ahora vamos a crear nuestro primer proyecto en Eclipse y ejecutar un programa simple de Java.

Paso 3: Crear un nuevo proyecto en Eclipse

1. Abre Eclipse.
2. En la pantalla de bienvenida, selecciona "Create a new Java project" o haz clic en **File > New > Java Project**.
3. Ponle un nombre a tu proyecto, por ejemplo: **MiPrimerProyectoJava**.
4. Haz clic en "Finish".

LIVE CODING

Paso 4: Crear una nueva clase Java

1. En el **Package Explorer**, haz clic derecho sobre el proyecto recién creado y selecciona **New > Class**.
2. Pon el nombre de la clase: **HolaMundo** (asegúrate de marcar la opción "public static void main(String[] args)" para crear el método principal automáticamente).
3. Haz clic en "Finish".



1. Solución

```
public class HolaMundo {  
    public static void main(String[] args) {  
        // Imprimir un mensaje en la consola  
        System.out.println("¡Hola, Mundo!");  
    }  
}
```



Momento:

Time-out!

 5 -10 min.





Ejercicio N° 1

Instalación del JDK



Instalación del Java Development Kit (JDK)

Contexto: 🙌

Para poder desarrollar aplicaciones en el lenguaje Java, es necesario tener instalado el JDK. En este ejercicio práctico vamos a instalar el Kit de desarrollo Java.

Consigna: 🛠️

Buscar e instalar el JDK 8 en tu computadora.

Tiempo 🕒 : 25 minutos

Instalación del Java Development Kit (JDK)

Paso a paso:



1. Ir a la página web de [Oracle JDK](#).
2. Buscar la sección de descargas de JDK 8 y seleccionar el instalador según tu sistema operativo.
3. Hacer clic en el botón de descarga y esperar a que se complete.
4. Cuando finalice la descarga, ejecutar el instalador.
5. Seguir los pasos del asistente de instalación. Aceptar la licencia.
6. Elegir la ruta de instalación en el equipo. Recomendado en Archivos de Programa.
7. Una vez completada la instalación, abrir la terminal o símbolo del sistema (CMD).
8. Escribir el comando "java -version" y presionar Enter.
9. Se deberá mostrar información sobre la versión instalada.
10. Listo, ¡el JDK está instalado correctamente!

¿Alguna consulta?





Resumen

¿Qué logramos en esta clase?



- ✓ Conocer el lenguaje Java y sus características.
- ✓ Conocer los componentes de Java.
- ✓ Crear y ejecutar tu primer proyecto Java.



¡Ponte a prueba!

Momento de ejercitación

Te invitamos a aprovechar esta última sección del espacio sincrónico para realizar de manera individual las **actividades disponibles en la plataforma**. Estas propuestas son clave para afianzar lo trabajado y **forman parte obligatoria del recorrido de aprendizaje**.

👉 **Análisis de caso** ————— 👉 **Selección Múltiple**

👉 **Comprensión lectora**

Si al resolverlas surge alguna duda, compartela o tráela al próximo encuentro sincrónico.

< **¡Muchas gracias!** >

