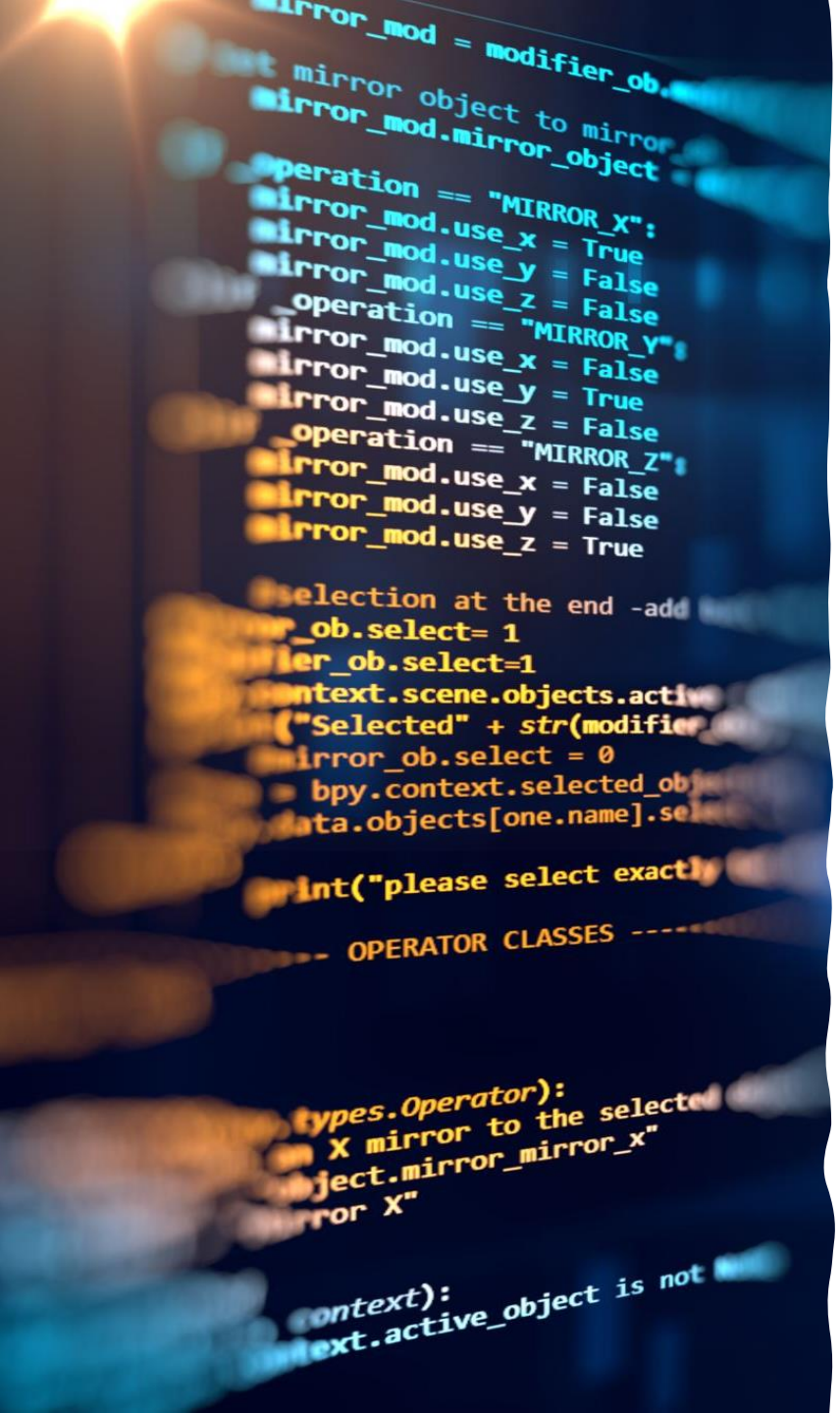
Programación en Java

Objetivo:

Introducir al alumno en el desarrollo de aplicaciones en JAVA.

Que el alumno conozca los principios básicos de la programación estructurada a partir de los siguientes conceptos:

- Tipos primitivos de datos
- Variables.
- Asignación.
- Operadores aritméticos
- Expresiones
- Estructuras de flujo de control if, switch.



Qué es Java

El lenguaje de programación Java fue originalmente desarrollado por James Gosling de Sun Microsystems (la cual fue adquirida por la compañía Oracle) y publicado en 1995 como un componente fundamental de la plataforma Java de Sun Microsystems. Su sintaxis deriva en gran medida de C y C++, pero tiene menos utilidades de bajo nivel que cualquiera de ellos.

Las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente.

Es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos 10 millones de usuarios reportados

Historia de Java

El nacimiento de Java se remonta al año 1991, cuando James Gosling y su equipo comenzaron a trabajar en la creación de un lenguaje para la empresa Sun Microsystems. Al poco tiempo de iniciar este proyecto, el equipo cambió su enfoque para alcanzar una meta más ambiciosa: diseñar un lenguaje para la recientemente creada World Wide Web. Finalmente, en 1995, Java fue lanzado al público para ser utilizado tanto para aplicaciones online como para la programación.

Desde su lanzamiento, Java se destacó como el único lenguaje en su tipo, debido a que su propia traducción funcionaba de forma diferente a las demás opciones de programación disponibles. Esta traducción radica en que compila e interpreta de forma paralela, cuando el resto de los lenguajes únicamente ejecutaba una de las dos operaciones. Esta sutil diferencia provocó un gran impacto e hizo de Java una de las opciones más populares de programación.

En un principio el lenguaje comenzó como una forma para conectar sistemas dentro de las oficinas y otros espacios de comunicación, pero al cambiar el enfoque hacia el mundo online Java trascendió esta frontera y entró de lleno en la web 2.0, o red interactiva.

No pasó mucho tiempo antes de que otras opciones en el mercado comenzaran a ganar popularidad, por lo que paulatinamente Java perdió terreno en el mundo online. Hasta que en el año 2016 Oracle la empresa que compró a Sun Microsystems en 2010) decidió disminuir el protagonismo de Java y en poco tiempo muchos de sus subprogramas fueron retirados.

Así fue que Java dejó de ser una de las opciones más populares para la programación web y pasó a ser una herramienta más relevante para los espacios de trabajo y para el desarrollo de tecnologías.

Ahora que conoces la historia breve de este lenguaje, es momento de ver qué rol desempeña en la actualidad tecnológica, así como las industrias y sectores que más lo utilizan.

¿Para qué sirve Java?

- Desarrollo de aplicaciones móviles
- Inteligencia Artificial
- Big Data
- Desarrollo de software
- Blockchain
- Internet de las cosas
- Desarrollo web

Diferencias entre Java y Javascript

- Debido a que Java y JavaScript son muy similares en sus nombres, estos dos lenguajes de programación han sido confundidos a lo largo de los años. Sin embargo, existen importantes diferencias entre ambos y solo una similitud: ambos son utilizados para la programación de software.
- JavaScript es un lenguaje exclusivamente interpretado, mientras que Java es un lenguaje compilado e interpretado a la vez. Esto puede parecer un detalle menor, pero en realidad implica algunas limitantes para los programadores, ya que con JavaScript se debe compilar el código para diferentes dispositivos, cuando con Java esto se hace automáticamente.
- Seguramente te estarás preguntando por qué ambos lenguajes comparten parte de su nombre. La respuesta no tiene que ver con sus características, sino con la historia de estas tecnologías y con una estrategia de marketing.
- Originalmente JavaScript llevaba por nombre LiveScript y era un lenguaje de programación creado por Netscape, compañía dedicada a la creación de software para la navegación en internet. Fue en el año de 1995 cuando Netscape anunció que su navegador soportaría el lenguaje Java, cambiando su nombre a JavaScript para atraer a más clientes. Estrictamente, estos dos lenguajes tenían diferentes funcionalidades y fueron creados por desarrolladores distintos.
- Eventualmente, Oracle compró Sun Microsystems, junto con los derechos de Java, mientras que Netscape fue adquirido por AOL, aunque JavaScript (ahora llamado oficialmente ECMAScript) se mantiene como un software libre.
- Veamos algunas de las funcionalidades de Java, así como la forma en que este lenguaje puede satisfacer tus necesidades de programación.

Características de Java

1. Simplicidad: Una de las principales ventajas de Java es que, debido a su sintaxis y reglas, es uno de los lenguajes más simples de utilizar. Por ejemplo, sus operadores y apuntadores son realmente sencillos.

2. Portabilidad: Java es un lenguaje de programación que no es exclusivo de una única plataforma. Esto significa que los desarrolladores pueden escribir el código una vez y correrlo en una gran diversidad de sistemas operativos, equipos y dispositivos.

3. Dirigido a objetos: En Java, todo es considerado como un objeto y este es uno de los principios de su enfoque basado en el polimorfismo. Esto implica que cada elemento dentro del código tiene un comportamiento y estado, lo cual añade estructura al lenguaje y permite aplicar diferentes implementaciones a los objetos. Las clases son las plantillas que Java utiliza para denominar a los objetos, mientras que las instancias permiten hacer referencia a ellos durante la ejecución del programa. Por su parte, la herencia permite usar el mismo código de una clase en otras clases que heredan de ella sus métodos (funciones) y sus atributos.

4. Ejecución en dos pasos: Como hemos visto, Java funciona mediante la compilación e interpretación simultánea durante el proceso de ejecución de órdenes. Esto hace que el código sea utilizable en muchos sistemas operativos, ya que se interpreta para cada uno de ellos.

Características de Java

5. Seguridad: Java ofrece un alto nivel de seguridad gracias a su ejecución en dos pasos. Debido a que el código tiene que pasar por ambos procesos, es mucho más difícil hackear o modificarlo de una forma maliciosa.

6. Dinamismo: Java está diseñado para adaptarse a los cambios, sin sacrificar velocidad y optimizando la memoria. Esto hace que el mismo código sea capaz de identificar errores y solucionarlos durante la compilación. Además, debido a que Java integra constructores propios, controla la forma en que un identificador se relaciona con una clase u objeto, asegurando que el programa corra de forma correcta.

7. Distributividad: Este lenguaje está pensado para distribuir operaciones entre diferentes equipos. Al emplear soluciones online, es posible dividir tareas y funcionalidades entre dos o más sistemas y compartir información para que el programa corra eficientemente.

8. Independencia: Java es uno de los sistemas con mayor independencia. Esto también significa que su sintaxis y objetos son ampliamente reconocidos y es mucho más fácil integrarlos a un equipo. Con un sistema de código cerrado, realmente fácil de aprender, ofrece una experiencia de uso simple y destaca por su rendimiento.

Por último, recuerda que Java y JavaScript no son lo mismo y que hoy en día hay una gran cantidad de empresas, sectores e industrias que utilizan este lenguaje de programación debido a sus ventajas.

Tipos primitivos de datos

TIPOS DE DATOS EN JAVA		NOMBRE	TIPO	OCUPA	RANGO APROXIMADO
	TIPOS PRIMITIVOS (sin métodos; no son objetos; no necesitan una invocación para ser creados)	byte	Entero	1 byte	-128 a 127
		short	Entero	2 bytes	-32768 a 32767
		int	Entero	4 bytes	$2*10^9$
		long	Entero	8 bytes	Muy grande
		float	Decimal simple	4 bytes	Muy grande
		double	Decimal doble	8 bytes	Muy grande
		char	Carácter simple	2 bytes	---
		boolean	Valor true o false	1 byte	---

Los primeros lenguajes de programación no usaban objetos, solo variables. Una variable podríamos decir que es un espacio de la memoria del ordenador a la que asignamos un contenido que puede ser un valor numérico (sólo números, con su valor de cálculo) o de tipo carácter o cadena de caracteres (valor alfanumérico que constará sólo de texto o de texto mezclado con números)

Tipos de Objetos

TIPOS DE DATOS EN JAVA	**TIPOS OBJETO (con métodos, necesitan una invocación para ser creados)**	Tipos de la biblioteca estándar de Java	String (cadenas de texto) Muchos otros (p.ej. Scanner, TreeSet, ArrayList...)
		Tipos definidos por el programador / usuario	Cualquiera que se nos ocurra, por ejemplo Taxi, Autobus, Tranvia
		arrays	Serie de elementos o formación tipo vector o matriz. Lo consideraremos un objeto especial que carece de métodos.
		Tipos envoltorio o wrapper (Equivalentes a los tipos primitivos pero como objetos.)	Byte
			Short
			Integer
			.

Cuáles sistemas operativos soportan java

Java es un lenguaje de programación versátil que se ejecuta en diversas plataformas. A continuación, te proporciono información sobre los sistemas operativos compatibles con Java:

Windows:
Windows 10 (8u51 y superiores)
Windows 8.x (escritorio)
Windows 7 SP1
Windows Vista SP2
Windows Server 2008 R2 SP1 (64 bits)
Windows Server 2012 y 2012 R2 (64 bits)

Requisitos:
RAM: 128 MB
Espacio en disco: 124 MB para JRE; 2 MB para Java Update

Exploradores compatibles: Internet Explorer 9 y superior, Firefox1.
Mac OS X:
Mac con procesador Intel que ejecuta Mac OS X 10.8.3+ o posterior
Privilegios de administrador para la instalación Se requiere un explorador de 64 bits (como Safari) para ejecutar Oracle Java en Mac1.

Linux:
Oracle Linux 5.5+
Oracle Linux 6.x (32 bits y 64 bits)
Oracle Linux 7.x (64 bits, 8u20 y superiores)

Red Hat Enterprise Linux 5.5+
Red Hat Enterprise Linux 6.x (32 bits y 64 bits)
Red Hat Enterprise Linux 7.x (64 bits, 8u20 y superiores)

Suse Linux Enterprise Server 10 SP2+, 11.x
Suse Linux Enterprise Server 12.x (64 bits, 8u31 y superiores)
Ubuntu Linux 12.04 LTS, 13.x, 14.x, 15.04, 15.10 (8u25 y superiores)

Explorador compatible: Firefox1.

Solaris: Consulta las configuraciones del sistema Java 8 soportadas para obtener información detallada sobre las plataformas, sistemas operativos y exploradores compatibles1.

En resumen, Java es compatible con una amplia variedad de sistemas operativos, lo que lo convierte en una excelente opción para desarrolladores y usuarios en todo el mundo2.

Diferencias en JDK y JRE

La **diferencia** entre el **JDK (Java Development Kit)** y el **JRE (Java Runtime Environment)** es fundamental para comprender cómo funcionan y cuándo debemos utilizar uno u otro:

JDK (Java Development Kit):

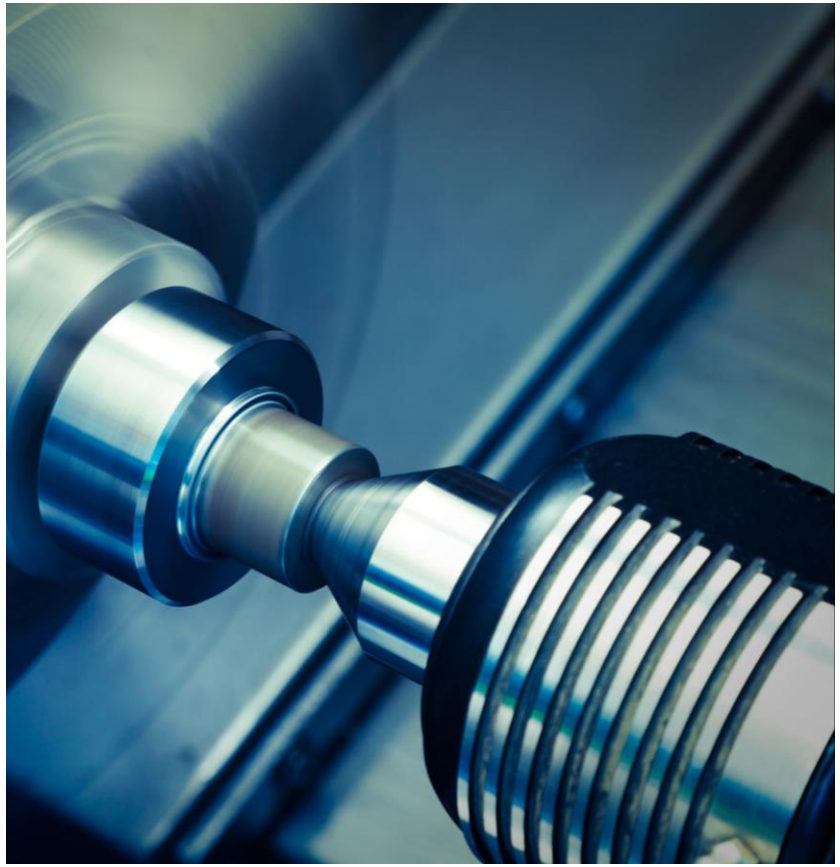
- El **JDK** es un conjunto de **herramientas de desarrollo de Java**.
- **Funciones**:
  - **Compilación**: Contiene el **compilador** (javac) que traduce el código fuente de Java a bytecode ejecutable.
  - **Depuración**: Proporciona herramientas para **depurar** y analizar el código.
  - **Desensamblador**: Incluye el **desensamblador de binarios** (javap) para examinar archivos de clase.
  - **Evaluación de rendimiento**: Ofrece herramientas como **VisualVM** y **Mission Control** para evaluar el rendimiento de las aplicaciones.
- **Contenido**: Un **JDK** ya contiene un **JRE** dentro de sus carpetas.

JRE (Java Runtime Environment):

- El **JRE** es el **Entorno de Ejecución de Java**.
- **Funciones**:
  - **Ejecución**: Contiene la **JVM (Java Virtual Machine)** y otras herramientas necesarias para ejecutar aplicaciones Java.
  - **Sin compiladores**: No incluye compiladores ni herramientas de desarrollo.
- **Uso**:
  - **Desarrollo**: Instala el **JDK** cuando deseas **desarrollar** aplicaciones Java.
  - **Ejecución**: Instala solo el **JRE** en equipos donde solo se ejecutarán aplicaciones Java.
- **En entornos productivos**:
  - **JDK**: Si necesitas herramientas de desarrollo en el servidor (por ejemplo, para pruebas básicas).
  - **JRE**: Si solo se ejecutarán aplicaciones Java sin necesidad de desarrollo.

Máquina Virtual de Java

La máquina virtual de Java (JVM) es un componente esencial en el mundo de la programación. Permíteme explicarte en qué consiste y por qué es tan relevante:



¿Qué hace la máquina virtual de Java?

La JVM es un entorno de tiempo de ejecución que permite ejecutar programas escritos en Java.

Funciona como un intérprete capaz de entender y ejecutar instrucciones expresadas en código Java.

Su característica más destacada es la portabilidad: permite que las aplicaciones Java se ejecuten en diferentes sistemas operativos y navegadores sin modificaciones.

Siguiendo el lema "escríbelo una vez, ejecútalo en cualquier parte", la JVM garantiza que el código Java sea independiente de la plataforma12.

¿Cómo funciona la JVM?

La JVM actúa como un puente entre el código Java y el sistema sobre el que se ejecuta.

Está implementada en diversos softwares, hardware y herramientas de desarrollo.

Es fundamental para los programadores, especialmente durante las pruebas de programas en fase de desarrollo.

La JVM interpreta el código Java y lo ejecuta en el sistema subyacente1.

En resumen, la máquina virtual de Java es la clave para que el lenguaje Java sea verdaderamente portátil y funcione en cualquier lugar donde esté presente la JVM

https://www.manualweb.net/java/tecnologias-java/

Tecnologías Java

Dentro de Java existen diferentes tecnologías de desarrollo, cada una enfocada a un fin diferente, ya sea la base del lenguaje Java, Java para el ámbito empresarial, Java para el desarrollo de aplicaciones móviles,…

Cada una de las tecnologías de desarrollo del lenguaje Java contiene:

Java Virtual Machine (JVM)

API de desarrollo de la plataforma

La aplicación se ejecuta dentro de la Java Virtual Machine (JVM) y tiene capacidad de accerder al API, que son las librerías con funcionalidades que nos ofrece Java.

Las tecnologías que existen en la plataforma Java son:

Java SE

Java SE es la plataforma estándar y objetivo de este tutorial sobre Java en la cual se recogen todas las funcionalidades básicas del lenguaje.

Dentro de estas funcionalidades básicas de Java encontramos: el uso de colecciones, acceso a ficheros con Java IO y NIO y bases de datos con JDBC, librerías para el desarrollo de aplicaciones de escritorio o web como Swing o JavaFX, librerías para la fecha y hora, posibilidad de crear aplicaciones multi-hilo, capacidades para realizar conexiones en red, manejo de contenido XML… incluso incluye la base de datos Java DB para el uso en memoria.

Si estás empezando con Java lo más normal es que te bajes las librerías de Java SE.

Puedes consultar todo el contenido de Java SE.

Java EE

Java EE se crea para poder realizar aplicaciones empresariales con Java. De esta forma se dota a Java EE con capacidades de desarrollo de aplicaciones de servidor con tecnologías como Servlets, JSP o EJB.

Java EE nos permite realizar el desarrollo de servicios, ya sean WSDL (con JAX-WS), REST (con JAX-RS), o la creación de websockets.

Además ofrece un API de persistencia de objetos con JPA, capacidades de mensajería con Java Message, de email con Java Mail o gestión de procesos batch.

Java ME

Java ME es la implementación de Java que nace para la creación de aplicaciones móviles.

Si bien con el paso del tiempo se ha ido enfocando más para el desarrollo de dispositivos IoT (Internet of Things): televisiones, sensores, impresoras,…

Así, dentro de Java ME podemos encontrar:

Java TV, para el desarrollo de aplicaciones en TV o en dispositivos multimedia.

Java Embedded, que nos permite crear diferentes perfiles de desarrollo de “aplicaciones incrustadas”, que además no tienen interface gráfica.

Java Cards

Es la tecnología de Java que nos sirve para el desarrollo de aplicaciones que vayan a ir en tarjetas inteligentes, aquellas que llevan un chip y poca capacidad de procesamiento y memoria.