

1. EVOLUCIÓN JAVA

1.1. ¿Qué es Java?

Java es un lenguaje de programación de propósito general que fue creado por Sun Microsystems en 1995 (ahora propiedad de Oracle Corporation). Java se destaca por ser un lenguaje orientado a objetos y por su capacidad de ejecutarse en múltiples plataformas, lo que significa que el código Java puede ejecutarse en diversos sistemas operativos así como computadoras personales, dispositivos móviles, servidores y sistemas embebidos, todo esto sin necesidad de modificaciones significativas. Esto se logra a través del uso de la máquina virtual Java (JVM), que interpreta el código Java en tiempo de ejecución.

Asimismo, Java es un lenguaje compilado, lo que significa que el código fuente es convertido a código máquina antes de ser ejecutado. Es utilizado por desarrolladores para crear aplicaciones web, aplicaciones móviles, juegos, software empresarial, etc.

1.2. Plataforma Java

La plataforma Java es un conjunto de componentes que permiten el desarrollo y ejecución de aplicaciones Java. Los componentes principales de la plataforma Java son:

- La JVM (Java Virtual Machine), que es un software que interpreta el código Java y lo convierte en código máquina para la plataforma específica en la que se está ejecutando.
- El entorno de ejecución Java (JRE, Java Runtime Environment)
- El JDK (Java Development Kit), que es un conjunto de herramientas que permiten desarrollar aplicaciones Java.
- La API (Application Programming Interface), que es un conjunto de clases y métodos que proporcionan funcionalidades para el desarrollo de aplicaciones Java.

1.1. Ediciones y versiones de Java

Java se divide en diferentes ediciones (Editions) y versiones (Versions), las más conocidas son:

- Java Standard Edition (Java SE): Destinada a aplicaciones de escritorio y desarrollo en general.
- Java Enterprise Edition (Java EE): Orientada a aplicaciones empresariales y web.
- Java Micro Edition (Java ME): Diseñada para aplicaciones en dispositivos móviles y embebidos.

1.2. Plataforma Java

La plataforma Java es el entorno para desarrollar y gestionar applets y aplicaciones Java. Consta de tres componentes principales: el lenguaje Java, los paquetes Java y la máquina virtual Java.

El lenguaje Java y los paquetes son similares a C++ y sus bibliotecas de clases. Los paquetes Java contienen clases, que están disponibles en cualquier implementación Java compatible. La interfaz de programación de aplicaciones (API) debe ser la misma en cualquier sistema que soporte Java.

Java difiere de un lenguaje tradicional, como C++, en la forma en que compila y ejecuta. En un entorno de programación tradicional, usted escribe y compila el código fuente de un programa en código objeto para un sistema operativo y un hardware específicos. El código

objeto se enlaza con otros módulos de código objeto para crear un programa en ejecución. El código es específico con respecto a un conjunto determinado de hardware de sistema y no funciona en otros sistemas si no se realizan cambios. Esta figura muestra el entorno de despliegue de un lenguaje tradicional.

- Aplicaciones y applets Java

Un applet es un programa Java diseñado para incluirse en un documento web HTML. Puede escribir su applet Java e incluirlo en una página HTML, de la misma forma que se incluye una imagen. Cuando utiliza un navegador habilitado para Java para ver una página HTML que contiene un applet, el código del applet se transfiere al sistema y lo ejecuta la máquina virtual Java del navegador.

- Máquina virtual Java

La máquina virtual Java es un entorno de ejecución que puede añadir a un navegador web o a cualquier sistema operativo, como por ejemplo IBM i. La máquina virtual Java ejecuta las instrucciones que genera un compilador Java. Consta de un intérprete de código de bytes y un tiempo de ejecución que permiten que los archivos de clase Java se ejecuten en cualquier plataforma, independientemente de la plataforma en la que se desarrollaron originalmente.

- Archivos JAR y de clase Java

Un archivo JAR (Java ARchive) es un formato de archivo que combina muchos archivos en uno. El entorno Java difiere de otros entornos de programación en que el compilador Java no genera código de máquina para un conjunto de instrucciones específico de hardware. En su lugar, el compilador Java convierte el código fuente Java en instrucciones de máquina virtual Java, que almacenan los archivos de clase Java. Puede utilizar archivos JAR para almacenar los archivos de clase. El archivo de clase no tiene como destino una plataforma de hardware específica, sino que tiene como destino la arquitectura de máquina virtual Java.

- Hilos Java

Un hilo es una única secuencia independiente que se ejecuta dentro de un programa. Java es un lenguaje de programación multihilos, por lo que puede haber más de una hebra en ejecución en la máquina virtual Java a la vez. Los hilos Java proporcionan una forma para que un programa Java realice varias tareas al mismo tiempo. Un hilo es esencialmente un flujo de control de un programa.

- Java Development Kit

Java Development Kit (JDK) es software para desarrolladores Java. Incluye el intérprete Java, las clases Java y las herramientas de desarrollo Java: compilador, depurador, desensamblador, appletviewer, generador de archivos de resguardo y generador de documentación.

1.3. Historia de Java

Java fue creado por James Gosling y su equipo en Sun Microsystems a principios de la década de 1990. La primera versión pública de Java, conocida como "Java 1.0", se lanzó

en 1996. El objetivo de Java era crear un lenguaje de programación que fuera portátil, seguro y robusto. Desde entonces, Java ha evolucionado con nuevas versiones y características. Fue adquirido por Oracle Corporation en 2010.

Los Inicios de Java

El lenguaje Java fue desarrollado en sus inicios por James Gosling, en el año 1991. Inicialmente Java era conocido como Oak o Green.

La primera versión del lenguaje Java es publicada por Sun Microsystems en 1995. Y es en la versión del lenguaje JDK 1.0.2, cuando pasa a llamarse Java, corría el año 1996.

En las primeras versiones de Java 1.1, 1.2 y 1.3 es en la que el lenguaje va tomando forma, con la inclusión de tecnologías como JavaBeans, JDBC para el acceso a base de datos, RMI para las invocaciones en remoto, Collections para la gestión de múltiples estructuras de datos o AWT para el desarrollo gráfico, entre otros.

Java Community Process (JCP)

La versión Java 1.4 pasa a ser la primera versión gestionada por la comunidad mediante el Java Community Process (JCP).

Se trabaja con Java Specification Requests (JSRs) que son las nuevas funcionalidades que se busca que tenga el lenguaje.

Java 1.4 se liberaba como JSR 59, corría el año 2002. Algunas de las características que contenía eran: librería NIO para IO no bloqueante, JAXP para el procesamiento de XML y XSLT o el API para preferencias.

Java 5

En 2004 se estaba trabajando con la versión Java 1.5, pero con vistas a reflejar el nivel de madurez de la plataforma Java se renombra a Java 5.

A partir de este momento se identifica el JDK con la versión 1.x, mientras que la plataforma Java sigue con la nueva política de versionado.

Así JDK 1.5 corresponde con Java 5, JDK 1.6 corresponde con Java 6, ... y así sucesivamente.

Dentro de Java 5 podemos encontrar el uso de genéricos, el autoboxing/unboxing entre tipos de datos primitivos y sus clases, el uso de enumerados y la aparición del bucle for-each.

Java 6

En el año 2006 aparece la versión Java 6 en la que podíamos encontrar cosas como el soporte de lenguajes de script, facilidades para la exposición y consumo de webservices mediante JAX-WS, nuevos tipos de drivers con JDBC 4 y la versión 2 de JAXB.

Java como Open Source

Una de las cosas que sucede en noviembre 2006 es que Sun Microsystems lo convierte en Open Source mediante una licencia GNU General Public License (GPL).

Dando lugar en mayo 2008 a lo que se conoce como OpenJDK, con OpenJDK 6.

Java 7

Llegado julio de 2011 ve la luz Java 7, la cual trae como novedades el soporte de lenguajes dinámicos, dotando a la JVM de un soporte de múltiples lenguajes y una nueva librería I/O para el manejo de ficheros.

También aparecen cosas menores, pero muy útiles como el manejo de String dentro de la validación en una estructura switch o la capacidad de poner subrayados en los números para que se puedan leer mejor.

Java 8

Versión aparecida en marzo de 2014.

Entre las características de Java 8 tenemos el soporte expresiones Lambda y uso de Streams, que permiten un estilo más funcional para los programas Java. Dentro de este enfoque más funcional también aparecen las transformaciones MapReduce.

Ve la luz el Proyecto Nashorn para disponer de un engine Javascript y así poder incluir este lenguaje dentro de las aplicaciones Java.

Otras cosas son un nuevo API Date y Time y la inclusión de JavaFX 8 dentro de la JDK de Java.

1.4. JDK

(Java Development Kit) es un conjunto de herramientas que los desarrolladores pueden utilizar para crear aplicaciones Java. El JDK incluye las siguientes herramientas:

- javac, que es un compilador que convierte el código fuente Java en bytecode.
- java, que es un intérprete que ejecuta el bytecode Java.
- jar, que es una herramienta que empaqueta los archivos Java en un archivo JAR.
- javadoc, que es una herramienta que genera documentación para las clases Java.

1.5. JVM

Es responsable de ejecutar el código compilado en bytecode y proporcionar una capa intermedia entre el software y el hardware.

La JVM (Java Virtual Machine) es un software que interpreta el código Java y lo convierte en código máquina para la plataforma específica en la que se está ejecutando. La JVM es responsable de la ejecución de las aplicaciones Java.

La JVM tiene las siguientes características:

- Portabilidad: La JVM permite que el mismo código Java se ejecute en diferentes plataformas.
- Seguridad: La JVM proporciona un entorno de ejecución aislado en el que se ejecuta el código Java.

Existen varias implementaciones de la JVM, incluyendo la JVM de Oracle, OpenJDK y otras desarrolladas por terceros.

1.6. Documentación API

Se define tres plataformas en un intento por cubrir distintos entornos de aplicación. Así, ha distribuido muchas de sus API (Application Program Interface) de forma que pertenezcan a cada una de las plataformas:

- Java ME (Java Platform, Micro Edition) o J2ME — orientada a entornos de limitados recursos, como teléfonos móviles, PDAs (Personal Digital Assistant), etc.
- Java SE (Java Platform, Standard Edition) o J2SE — para entornos de gama media y estaciones de trabajo. Aquí se sitúa al usuario medio en un PC de escritorio.
- Java EE (Java Platform, Enterprise Edition) o J2EE — orientada a entornos distribuidos empresariales o de Internet.

Las clases en las API de Java se organizan en grupos disjuntos llamados paquetes. Cada paquete contiene un conjunto de interfaces, clases y excepciones relacionadas. La información sobre los paquetes que ofrece cada plataforma puede encontrarse en la documentación de ésta.

El conjunto de las API es controlado por Sun Microsystems junto con otras entidades o personas a través del programa JCP (Java Community Process). Las compañías o individuos participantes del JCP pueden influir de forma activa en el diseño y desarrollo de las API, algo que ha sido motivo de controversia.