

Historia y características de Java



Historia de Java

Java se creó como una herramienta de programación para ser usada en un proyecto de set-top-box en una pequeña operación denominada the Green Project en Sun Microsystems en 1991. El equipo (Green Team), compuesto por trece personas y dirigido por James Gosling, trabajó durante 18 meses en Sand Hill Road, en Menlo Park, para desarrollarlo.

El lenguaje se denominó inicialmente Oak, luego pasó a llamarse Green y finalmente se le renombró Java.

Los objetivos de Gosling eran implementar una máquina virtual y un lenguaje con una estructura y sintaxis similar a C++. Entre junio y julio de 1994 el equipo reorientó la plataforma hacia la Web incluyendo el navegador HotJava en el proyecto.



OAK



Historia de Java

Durante las conferencias de SunWorld, Java y HotJava (el navegador Web) vieron por primera vez la luz el 23 de mayo de 1995 pero sería en 1996 cuando pasa, por algunos problemas legales, a llamarse Java oficialmente.

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.

En el año 2010 la compañía Oracle compra Sun Microsystems y desde entonces ha continuado con el soporte del lenguaje y de su plataforma.



ORACLE

Características de Java (I)

Es simple

Java ofrece la funcionalidad de un lenguaje potente, derivado de C y C++, pero sin las características menos usadas y más confusas de estos, haciéndolo más sencillo.

Orientado a objetos

La Programación Orientada a Objetos (POO) es un paradigma de programación, es decir, un modelo o un estilo de programación que nos da unas guías sobre cómo trabajar con él. Se basa en el concepto de clases y objetos. Este tipo de programación se utiliza para estructurar un programa de software en piezas simples y reutilizables de planos de código (clases) para crear instancias individuales de objetos.

Es distribuido

Java proporciona una gran biblioteca estándar y herramientas para que los programas puedan ser distribuidos.

Características de Java (II)

Independiente a la plataforma

Esto significa que programas escritos en el lenguaje Java pueden ejecutarse en cualquier tipo de hardware, lo que lo hace portable.

Recolector de basura

Cuando no hay referencias localizadas a un objeto, el recolector de basura de Java borra dicho objeto, liberando así la memoria que ocupaba. Esto previene posibles fugas de memoria.

Es seguro y solido

Proporcionando una plataforma segura para desarrollar y ejecutar aplicaciones que, administra automáticamente la memoria, provee canales de comunicación segura protegiendo la privacidad de los datos y, al tener una sintaxis rigurosa evita que se quiebre el código, es decir, no permite la corrupción del mismo.

Características de Java (III)

Es multihilo

Java logra llevar a cabo varias tareas simultáneamente dentro del mismo programa. Esto permite mejorar el rendimiento y la velocidad de ejecución..

Extensamente utilizado

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 diez millones de usuarios reportados.

El Bytecode

Todo código Java se compila en un lenguaje intermedio que no es 100% binario, está justo entre medio del lenguaje máquina y el código fuente y se conoce como bytecode. Éste, puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura del equipo.

Características de Java (IV)

La promesa inicial de Gosling era:

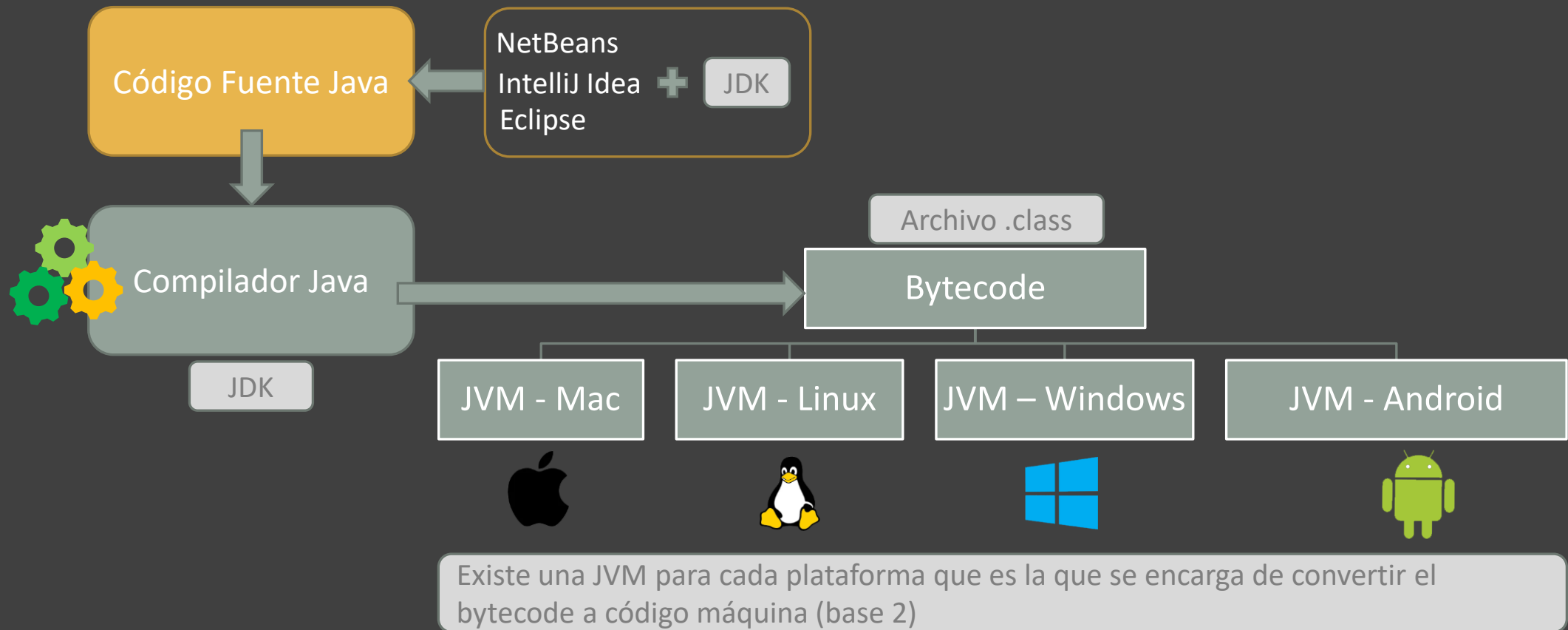
Write Once, Run Anywhere

(Escríbelo una vez, ejecútalo en cualquier lugar)

Proporcionando un lenguaje independiente del sistema operativo y un entorno de ejecución:
JVM (Máquina virtual de Java).

Java es rápido, seguro y fiable.

El Bytecode de Java



Java Environment

JRE – Java Runtime Environment

JVM – Java Virtual Machine

SDK – Software Development Kit

JDK – Java Development Kit

JSR – Java Specification Request

Java EE

SDK (Java EE)

JSR (Java EE)

Java SE



JRE

Librería JRE

Clases del núcleo
de la plataforma
Java

Bibliotecas de la
plataforma Java
de soporte

JVM



JDK
(Java SE)

JSR
(Java SE)

Java FX



SDK
(Java FX)

Java ME

SDK
(Java ME)



Java Card

SDK
(Java Card)

Es lo que
obtenemos al
descargar el
software de
Java

Java Environment

WWW



Servidores



Escritorio



Sist.Integrados



Tv



Móviles



Tarjetas

Keys APIs

Java EE

Java FX

Java TV

MSA

Plataforma

Java SE

Java ME

Java Card

Lenguaje

Lenguaje de programación Java

Java Environment

JVM – Java Virtual Machine

El corazón de la Plataforma Java es el concepto común de un procesador “virtual” que ejecuta programas escritos en el lenguaje de programación Java. En concreto, ejecuta el código resultante de la compilación del código fuente, conocido como bytecode. Este “procesador” es la máquina virtual de Java o JVM (Java Virtual Machine), que se encarga de traducir (interpretar o compilar al vuelo) el bytecode en instrucciones nativas de la plataforma destino. Esto permite que una misma aplicación Java pueda ser ejecutada en una gran variedad de sistemas con arquitecturas distintas

SDK – Java ME

El Java Me SDK es un conjunto de herramientas que permite el desarrollo de software sobre la plataforma Java ME (Micro Edition) para dispositivos con recursos limitados tales como teléfonos móviles, pagers, PDAs, etc. En la última versión (3.0) se integra CDC, CLDC y la tecnología Java BD-J (Blu-ray Disc Java) en un solo SDK. Para utilizarlo es necesario instalar previamente el JDK de Java SE.

SDK – Java FX

El kit de desarrollo de software (SDK) de JavaFX proporciona las herramientas y tecnologías de línea de comandos para desarrollar contenidos expresivos para aplicaciones desplegadas en navegadores, ordenadores de sobremesa y dispositivos móviles. Incluye JavaFX Desktop runtime, JavaFX Mobile Emulator and runtime (sólo para Windows), JavaFX API documentation y ejemplos.

Java Environment

SDK – Java EE

La plataforma Java Enterprise Edition (Java EE) define las APIs y requerimientos necesarios para poder ejecutar aplicaciones Java servidoras, con todo lo que ello supone: cliente-servidor, multiusuario, transaccionalidad, escalabilidad, etc...en definitiva, características que no eran importantes o imprescindibles en aplicaciones de escritorio.

Se apoya en la plataforma Java SE, por lo que es imprescindible conocer y dominar dicha plataforma antes de aventurarse en esta otra. Utiliza la misma Máquina Virtual Java (JVM).

SDK – Java Card

La plataforma Java Card define las APIs y requerimientos necesarios para poder ejecutar aplicaciones Java en los chips de las tarjetas.

Debido a las mínimas prestaciones del entorno de ejecución contiene el API más escueto.

Java Environment

JRE – Java Runtime Environment

Un programa destinado a la Plataforma Java necesita dos componentes en el sistema donde se va a ejecutar: una máquina virtual de Java (JVM), y un conjunto de bibliotecas para proporcionar los servicios que pueda necesitar la aplicación. La JVM que proporciona Sun Microsystems, junto con su implementación de las bibliotecas estándar, se conocen como Java Runtime Environment (JRE) o Entorno en tiempo de ejecución para Java. El JRE es lo mínimo que debe contener un sistema para poder ejecutar una aplicación Java sobre el mismo.

JDK – Java Development Kit

La escritura de aplicaciones y applets de Java necesita herramientas de desarrollo como JDK. JDK incluye Java Runtime Environment, el compilador Java y las API de Java. Familiarizarse resulta fácil para los programadores nuevos y con experiencia.

JSR – Java Specification Request

El Java Community Process, establecido en 1998, es un proceso formalizado que permite a las partes interesadas a involucrarse en la definición de futuras versiones y características de la plataforma Java.

El proceso JCP conlleva el uso de Java Specification Request (JSR), las cuales son documentos formales que describen las especificaciones y tecnologías propuestas para que sean añadidas a la plataforma Java.