

1.1. Qué es Java

Durante este curso exploraremos uno de los lenguajes que han dominado el mercado durante los últimos años. Las razones se explican en esta introducción y a lo largo del curso, el usuario encontrará motivos más que suficientes para explicar por qué una gran parte de las empresas punteras en el desarrollo de software han adoptado esta tecnología.

Características básicas:

Creado por Sun. Es multiplataforma. Lenguaje orientado a objetos

1.1.1. Historia

Con el nacimiento de la WWW, aparecen problemas porque existen multitud de plataformas y sistemas operativos.

El objetivo es crear una plataforma neutral, la primera idea es basarse en C++, que se amplía y se convierte en Oak.

En 1992 Oak pasa a llamarse Java, y se utiliza por primera vez con este nombre en el proyecto "Green" - *7. En los años 94 y 95 estalla la popularidad de la Web y es en este año cuando se comienza a hablar de Java como uno de los lenguajes más utilizados.

1.1.2. Características principales

- Además de ser potente es un lenguaje fácil de aprender.
- Abandona muchas de las características de C:
 - La aritmética de punteros, que generaban código no seguro.
 - Los tipos de datos registro (struct).
 - Los datos definidos por el usuario (typedef).
 - El acercamiento a la memoria, con funciones como malloc y free. El programador no tiene que pensar en estos detalles
- Implementa los conceptos fundamentales del paradigma de Programación Orientado a Objetos: jerarquía de clases, herencia, polimorfismo, encapsulación, ocultamiento de información.

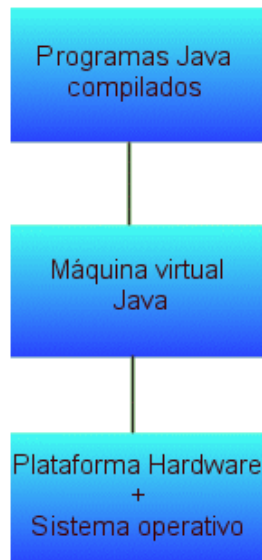
1.1.3. Es un lenguaje seguro

- Además de ser potente es un lenguaje fácil de aprender.
- Facetas:
 - En el lenguaje, características como los punteros de C y C++ se eliminan para prevenir el acceso ilegal a la memoria.
 - Cuando se usa Java para ejecutar aplicaciones que han bajado de la red se impide que puedan realizar acciones no previstas, como acceder a los recursos locales.
- Con esto evitamos ataques como Caballos de Troya.
- Verificador de byte-codes.
- El Cargador de Clases separa el espacio de nombres del sistema de ficheros local, del de los recursos procedentes de la red. Las clases importadas de la red se almacenan en un espacio de nombres privado, asociado con el origen.

1.2. Características del lenguaje

1.2.1. Multiplataforma

La virtud de ser multiplataforma es la característica más importante de este lenguaje, ahora veremos cómo se consigue la portabilidad de los programas desarrollados en este lenguaje.



JVM (Java Virtual Machine)

- La JVM es una computadora abstracta que ejecuta programas Java compilados.
- Debemos tener en cuenta que nuestros programas Java sólo funcionarán si tenemos la máquina virtual instalada.

Qué quiere decir: Computadora virtual

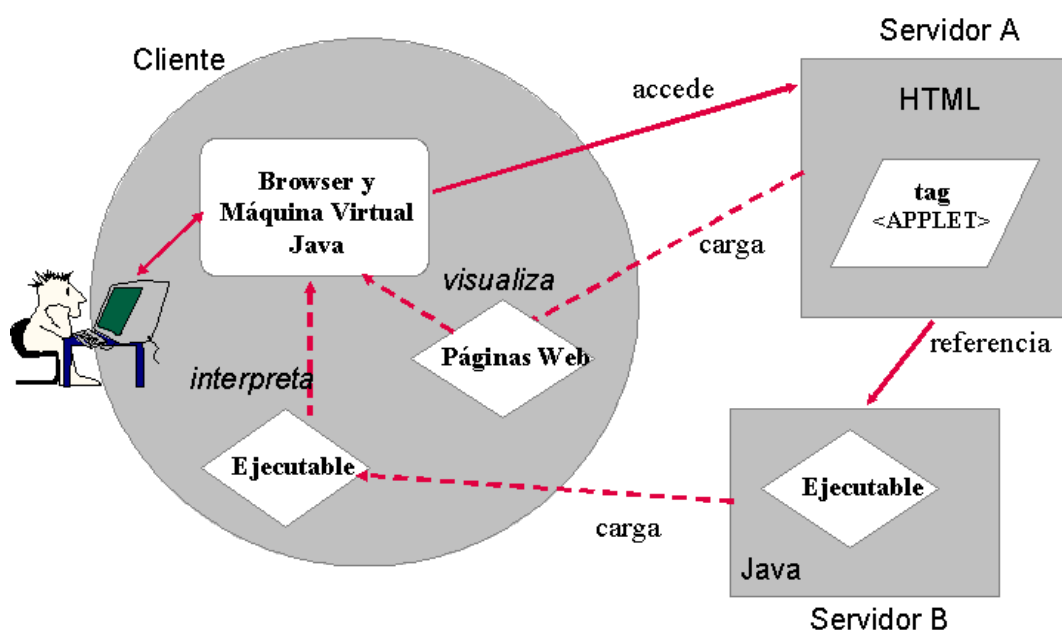
- Está implementada mediante software, que funciona sobre un hardware con Sistema Operativo.
- Tenemos que pensar que nuestros programas Java se ejecutarán gracias a "otro programa", la JVM.

Nuestros programas son multiplataforma:

Gracias a la idea de máquina virtual, si nosotros tenemos un programa Java, correrá sobre una máquina con sistema operativo Windows, o con sistema operativo Linux, lo que cambiará será la máquina virtual.

1.2.2. Para la Red

- **Puede estar incorporado en documentos Web:** mediante los conocidos Applets. A lo largo de los años Java se ha convertido en una referencia en el desarrollo Web.
- **Permite acceder a Sockets, URL, scripts CGI y otros elementos de red TCP/IP.**
- En Java podemos desarrollar aplicaciones Web de forma sencilla gracias a tecnologías como Applets, Servlets, JSP, etc.



1.3. Entorno de desarrollo

1.3.1. ¿Qué necesitamos?

¿Qué necesitamos para programar en Java?:

- El JDK (Java Development Kit).
- Un editor de texto, para crear programas.
- Y el appletviewer, para visualizar los applets, que viene incluido en el JDK.

Pero normalmente se utilizan los llamados IDE, que son herramientas que integran todos los elementos anteriores, es decir:

- Incorporan un JDK
- Son potentes editores de texto, que suelen asistir al programador sobre los métodos y atributos de las clases.
- También suelen incorporar utilidades como el appletviewer, y rmi.

1.3.2. IDEs populares

Entre los IDEs más populares podemos citar:

- JBuilder.
- JCreator.
- Eclipse.
- JDeveloper.
- NetBeans.

1.3.3. NetBeans, manejo básico.

Durante el desarrollo del curso se va a utilizar NetBeans.