APRENDERAPROGRAMAR.COM

LA MÁQUINA VIRTUAL JAVA (JVM). COMPILADOR E INTÉRPRETE. BYTECODE, CÓDIGO FUENTE Y CÓDIGO MÁQUINA. (CU00611B)

Sección: Cursos
Categoría: Curso “Aprender programación Java desde cero”

Fecha revisión: 2029

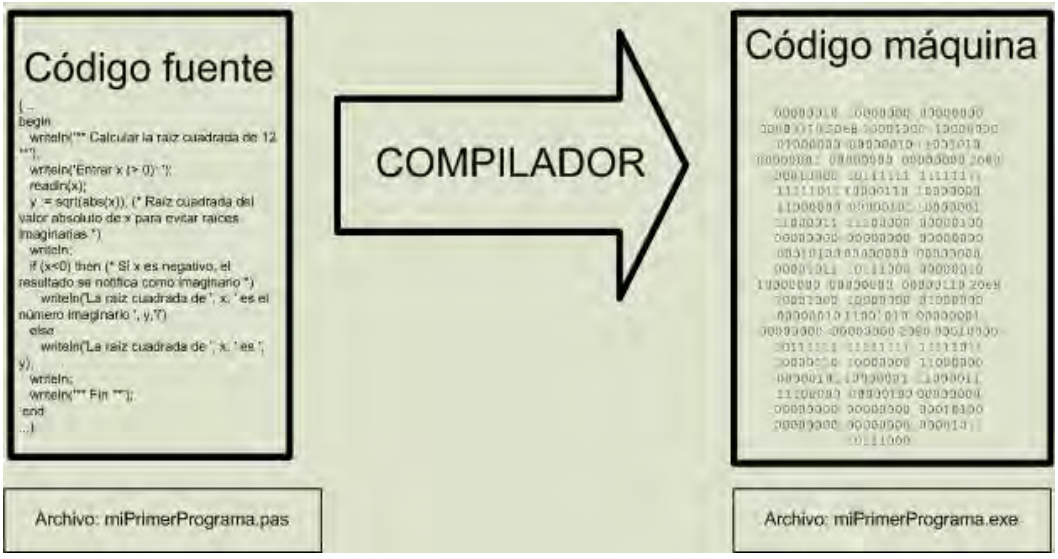
Resumen: Entrega nº11 curso Aprender programación Java desde cero.

Autores: Alex Rodríguez y Walter Sagástegui

MÁQUINA VIRTUAL JAVA (JAVA VIRTUAL MACHINE O JVM). COMPILADOR E INTÉRPRETE. BYTECODE.

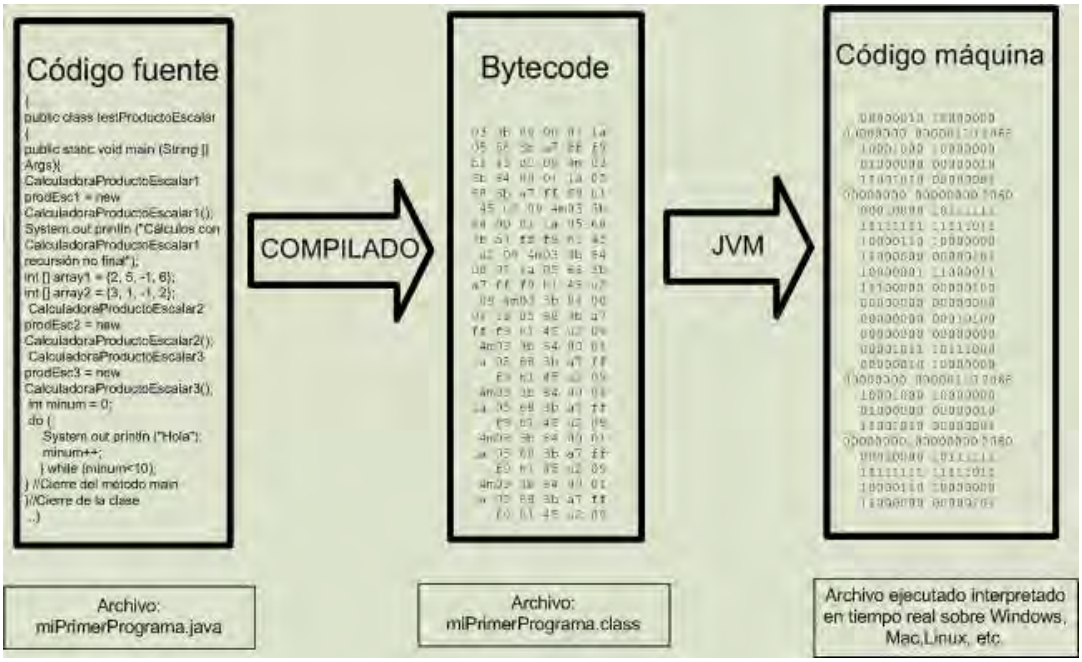
Vamos a crear nuestro primer programa, que nos servirá para comprobar si hemos instalado y configurado correctamente Java. Pero antes vamos a repasar algunos conceptos importantes que nos permitan entender lo que vamos haciendo.

El primer concepto a abordar es el de compilación. "Compilar" significa traducir el código escrito en "Lenguaje entendible por humanos" (por ejemplo Java, C, Pascal, Fortran), a un código en "Lenguaje Máquina", que entienden las máquinas, pero no entendible por nosotros. Se hace esto porque a los humanos nos resultaría casi imposible trabajar directamente con el lenguaje de los ordenadores. Es por eso por lo que usamos un lenguaje más asequible para nosotros (en nuestro caso Java) y luego empleamos un traductor (compilador). La creación de programas en muchos lenguajes se basa en el proceso: escribir código fuente → compilar y obtener programa ejecutable. El compilador se encarga de evitar que se pueda traducir un programa con código fuente mal escrito y de hacer otras verificaciones previas, de modo que el código máquina tiene ciertas garantías de que cumple cuando menos con los estándares de sintaxis obligatorios de un lenguaje.

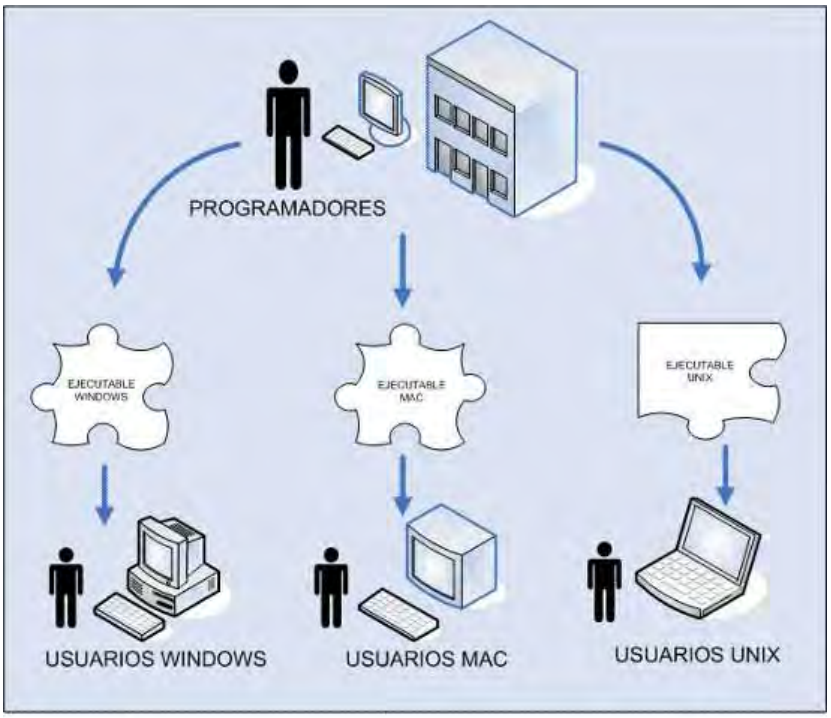


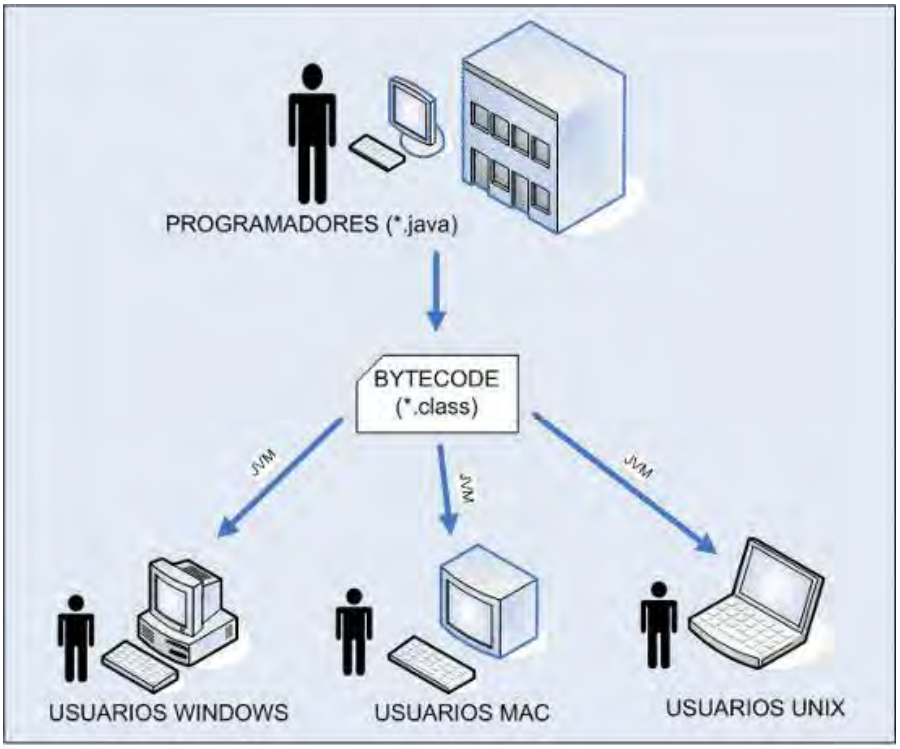
En este esquema, el archivo ejecutable no es válido para cualquier ordenador. Por ejemplo, si se ha generado el ejecutable para Windows, no podrá utilizarse en Macintosh. Sin embargo el proceso en Java no se corresponde con el gráfico anterior. Esta fue una característica novedosa de Java respecto a otros lenguajes cuando se lanzó la primera versión de Java. La novedad introducida fue que **Java se hizo independiente del hardware y del sistema operativo en que se ejecutaba**. En otros lenguajes existía el problema de compatibilidad descrito. Sin embargo, Java se hizo independiente de la plataforma añadiendo un paso intermedio: los programas Java no se ejecutan en nuestra máquina real (en nuestro ordenador o servidor) sino que Java simula una "máquina virtual" con su propio hardware y sistema operativo. En resumen, el proceso se amplía en un paso: del código fuente, se pasa a un código

intermedio denominado habitualmente “bytecode” entendible por la máquina virtual Java. Y es esta máquina virtual simulada, denominada Java Virtual Machine o JVM, la encargada de interpretar el bytecode dando lugar a la ejecución del programa.



Esto permite que Java pueda ejecutarse en una máquina con el Sistema Operativo Unix, Windows, Linux o cualquier otro, porque **en realidad no va a ejecutarse en ninguno de los sistemas operativos, sino en su propia máquina virtual** que se instala cuando se instala Java. El precio a pagar o desventaja de este esquema es que todo ordenador que quiera correr una aplicación Java ha de tener instalado Java con su máquina virtual. Las diferencias entre ambas concepciones podemos verlas en los siguientes esquemas.





ESQUEMA COMPILADO – INTERPRETADO JAVA

La máquina virtual era un aspecto importante que diferenciaba a Java de otros lenguajes cuando irrumpió en el mercado de los lenguajes de programación; permitía **escribir y compilar el programa una sola** vez en lugar de varias veces y ejecutar ese código en cualquier plataforma ("write once, run anywhere").

Otra razón de su gran éxito ha sido que cuando surgió se convirtió en un lenguaje más orientado a objetos que todos los otros lenguajes existentes. Además cabe destacar su potencia y el permitir crear programas de aspecto y funcionamiento muy similar al también muy popular "entorno Windows". Esto afianzó su reconocimiento como un lenguaje de programación innovador.

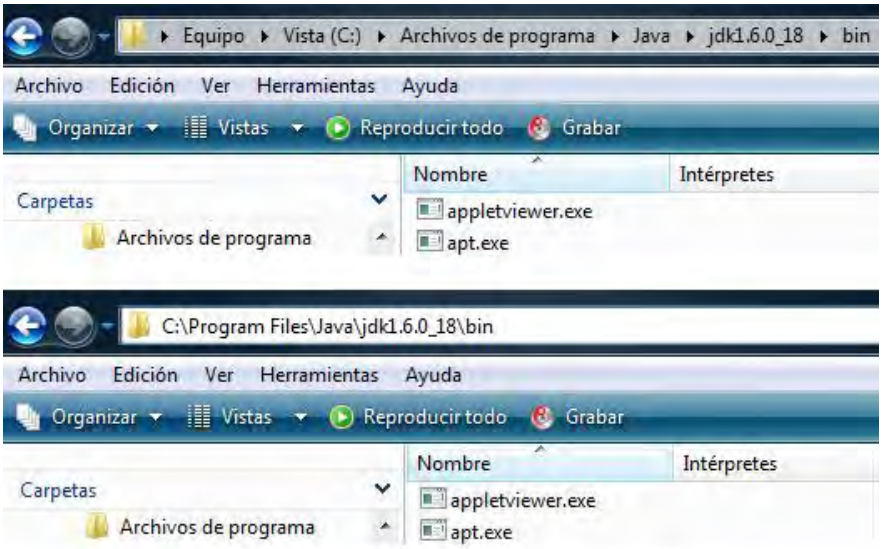
Aclarar que al ser Java un programa que se interpreta en una máquina virtual, el archivo resultante de la compilación es un archivo con la extensión .class interpretable por la máquina virtual. Este archivo .class está escrito en un *lenguaje de máquina virtual* (bytecode).

Para que la "Máquina Real" (nuestro ordenador) ejecute el programa, hay que "interpretar" (traducir) el archivo .class a un código en "Lenguaje de Máquina Real". Esta es la labor de lo que llamamos "intérprete" o traductor del lenguaje de la máquina virtual a la máquina real.

Los archivos respectivos que se encargan de estas tareas son:

- El compilador Java --- > javac.exe. Se encarga de compilar el código fuente.
- El intérprete Java --- > java.exe. Se encarga de interpretar los archivos .class (bytecode).

La ruta en la que se ubican ambos archivos es esta o una similar a esta: "C:\Program Files (x86)\Java\jdk1.7.0_51\bin" (o "C:\Program Files\Java\jdk1.7.0_51\bin", depende de la versión de Windows en caso de que usemos Windows). El explorador de Windows nos muestra una barra con la ruta en que nos encontramos (ruta aparente). Para conocer la ruta real basta pinchar sobre la ruta aparente.



En la próxima entrega veremos los pasos para compilar e interpretar nuestro primer programa escrito en lenguaje Java.

Próxima entrega: CU00612B

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:

http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=68&Itemid=188