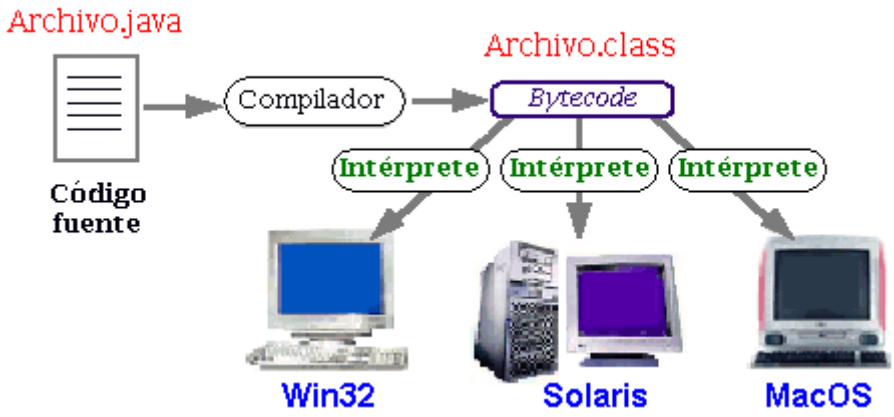
Java Virtual Machine

La mayoría de los lenguajes de programación se caracterizan por ser interpretados o compilados, lo que determina la manera en como serán ejecutados en una computadora.

Java tiene la característica de ser al mismo tiempo compilado e interpretado. El compilador es el encargado de convertir el código fuente de un programa en un código intermedio llamado bytecode que es independiente de la plataforma en que se trabaje y que es ejecutado por el intérprete de Java que forma parte de la Máquina Virtual de Java (Java Virtual Machine).

Compilación y ejecución de programas en Java.



Es por eso que se dice que Java tiene la característica de ser multiplataforma, pues el mismo bytecode (o archivo *.class), es interpretado por una máquina virtual diferente y adecuada para cada plataforma, es decir que lo que cambia no es el archivo compilado, si no la máquina virtual que lo va a interpretar; a diferencia de C, por ejemplo, que el código fuente tiene que ser compilado para cada plataforma.

El Kit de Desarrollo de Java (Java Development Kit)

El Kit de desarrollo de Java (JDK) contiene las herramientas y librerías necesarias para crear y ejecutar applets y aplicaciones en Java.

A continuación se listas algunas de las utilidades que se pueden encontrar en el JDK:

- **javac**. Es el compilador de Java. Se encarga de convertir el código fuente escrito en Java a *bytecode*.
- **java**. Es el intérprete de Java. Ejecuta el *bytecode* a partir de los archivos class.
- **appletviewer**. Es un visor de applets. En la mayoría de las ocasiones puede utilizarse en lugar de un Navegador Web.
- **javadoc**. Se utiliza para crear documentación en formato HTML a partir de el código fuente Java y los comentarios que contiene.
- **javap**. Es un desensamblador de Java.
- **jar**. Es una herramienta utilizada para trabajar con los archivos JAR.

Variables de entorno y Java

Una vez instalado el JDK, es necesario definir unos valores, en las llamadas variables de entorno, con el objeto de configurar la aplicaciones del JDK, para su correcto funcionamiento.

Una variable de entorno es una variable disponible a nivel del sistema operativo, y a la que cualquier aplicación puede tener acceso.

En el caso del JDK, es necesario definir dos variables de entorno para su correcto funcionamiento.

La primera es la variable PATH, que es una variable donde se almacenan las rutas donde se encuentran los archivos ejecutables. Para definir la variable PATH, se necesita dar la ruta absoluta del directorio de instalación del jdk, y agregar la carpeta bin, que es donde se encuentran los ejecutables como javac y java, como se muestra a continuación:

PATH=%PATH%;(directorio de instalación del jdk)\bin

Cabe señalar que en windows se utiliza %PATH% para dar el valor de la variable PATH, antes de la nueva asignación, por ejemplo si el PATH inicial es:

PATH=C:\mysql\bin

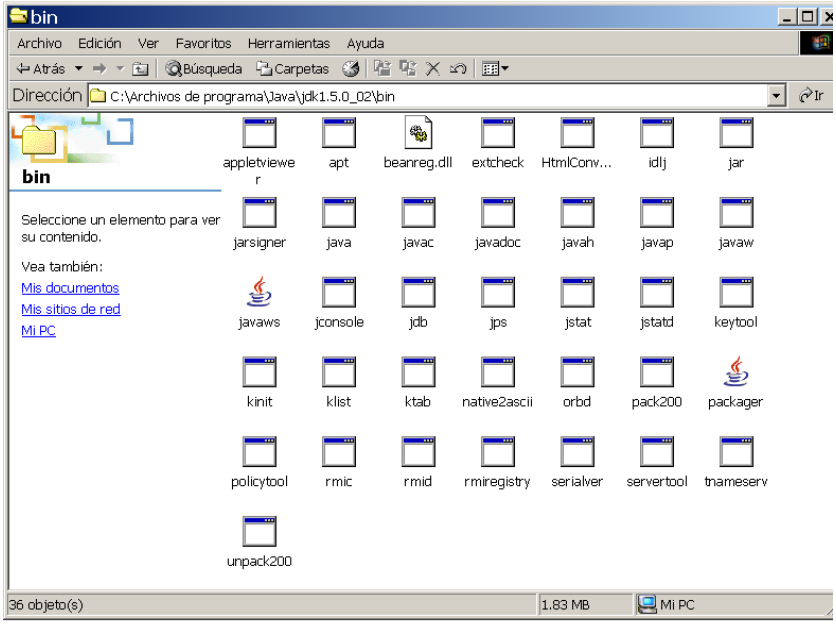
Al hacer

PATH=%PATH%;C:\Archivos de programa\Java\jdk1.5.0_02\bin

El nuevo valor de PATH será

PATH=C\mysql\bin\; C:\Archivos de programa\Java\jdk1.5.0_02\bin

Que es precisamente un ejemplo de PATH. Es conveniente recalcar que el PATH anterior solo es a nivel de ejemplo, dado que el valor que debe de tomar es el del directorio de instalación particular para cada PC, ya que puede variar dependiendo de cómo fue instalado el jdk, incluso de la versión.



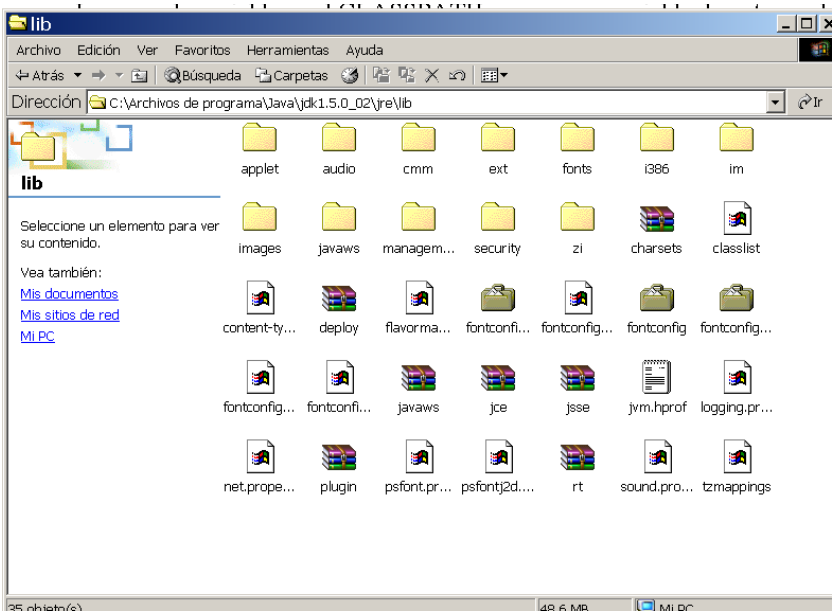
La segunda variable es el CLASSPATH, que es una variable de entorno donde se almacena la ruta donde se encuentran las clases de java que utilizará la máquina virtual para ejecutar la aplicación.

CLASSPATH=(directorio de instalación del jdk)\jre\lib;.

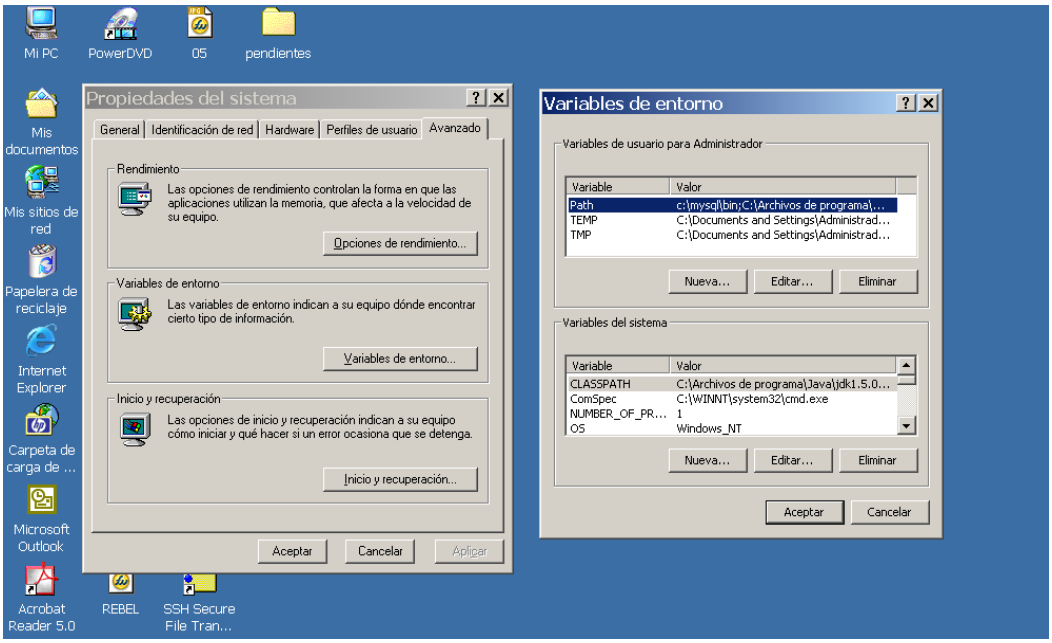
En la primera ruta especificamos donde encontrar clases que se encuentran dentro del jdk, y con el punto, especificamos que tambien en el directorio actual, hay clases, que comúnmente son las clases propias que creamos.

Un ejemplo de CLASSPATH puede ser

CLASSPATH= C:\Archivos de programa\Java\jdk1.5.0_02\jre\lib;.

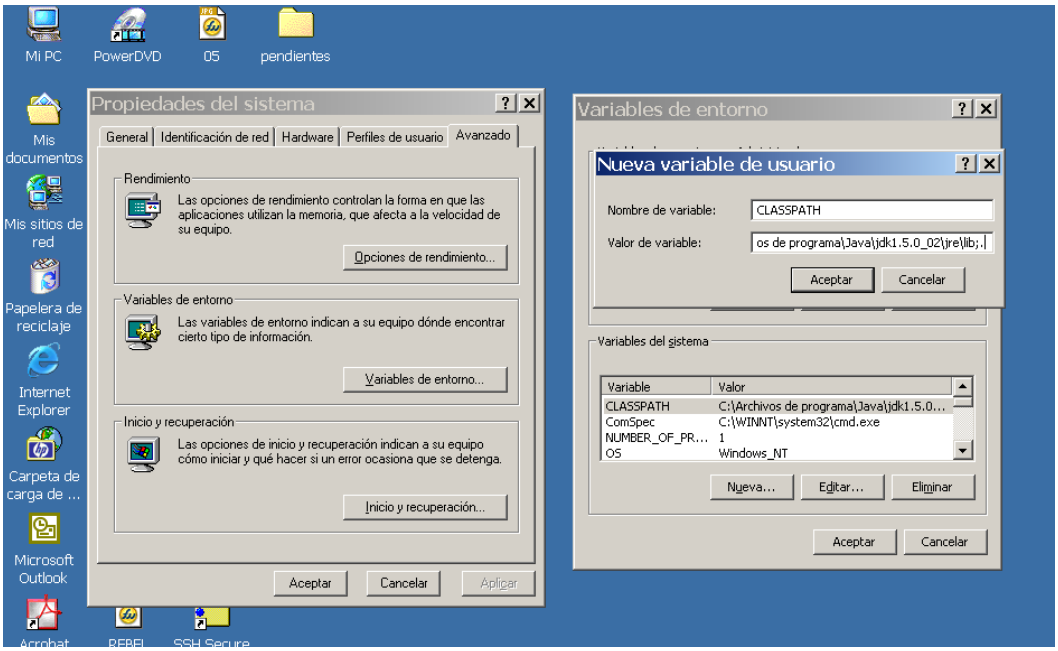


Para definir una variable de entorno en windows, podemos hacer clic derecho en MiPc, y elegir –propiedades-- del menu emergente. Una vez hecho cambiamos a la pestaña que dice –avanzado—y damos clic en variables de entorno.



Una vez en este punto podemos ver dos listas de variables de entorno con diferentes propósitos. La lista superior son variables que se definen por cada usuario del sistema. La lista inferior son variables que se definen a nivel de sistema y que por lo tanto se aplicarán para todos los usuarios del equipo, y solo por mencionarlos se necesitan privilegios de administrador para definirlas.

Para definir una variable de entorno, basta con buscarla en la lista y agregarle la ruta necesaria (como en el caso del PATH), o en su defecto, dar clic en el boton –Nueva y simplemente escribir el nombre y el valor que va a ir asociado a la variable de entorno.



Una vez definidas los valores de las variables de entorno PATH y CLASSPATH, podemos probar si fueron correctamente definidas en una consola de sistema, simplemente ejecutando javac y java, obteniendo una salida similar a la siguiente.

En el caso de los sistemas UNIX (GNU/Linux, BSD, Solaris ...), es algo muy similar, ya que después de instalar el JDK adecuado para cada plataforma, es necesario definir las variables de entorno PATH y CLASSPATH, pero atendiendo a la forma en como se definen las variables de entorno para citadas plataformas.