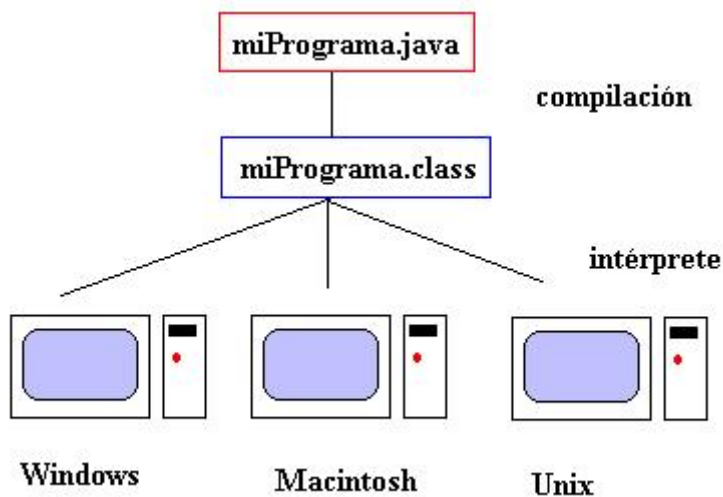


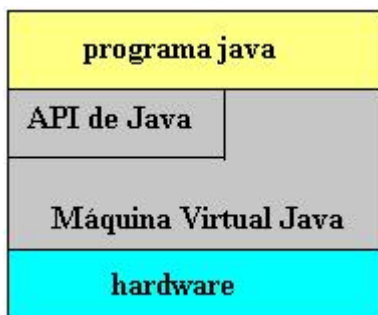
La Máquina Virtual Java

El lenguaje Java es a la vez compilado e interpretado. Con el compilador se convierte el código fuente que reside en archivos cuya extensión es **.java**, a un conjunto de instrucciones que recibe el nombre de *bytecodes* que se guardan en un archivo cuya extensión es **.class**. Estas instrucciones son independientes del tipo de ordenador. El intérprete ejecuta cada una de estas instrucciones en un ordenador específico (Windows, Macintosh, etc). Solamente es necesario, por tanto, compilar una vez el programa, pero se interpreta cada vez que se ejecuta en un ordenador.



Cada intérprete Java es una implementación de la Máquina Virtual Java (JVM). Los *bytecodes* posibilitan el objetivo de "write once, run anywhere", de escribir el programa una vez y que se pueda correr en cualquier plataforma que disponga de una implementación de la JVM. Por ejemplo, el mismo programa Java puede correr en Windows 98, Solaris, Macintosh, etc.

Java es, por tanto, algo más que un lenguaje, ya que la palabra Java se refiere a dos cosas inseparables: el lenguaje que nos sirve para crear programas y la Máquina Virtual Java que sirve para ejecutarlos. Como vemos en la figura, el API de Java y la Máquina Virtual Java forman una capa intermedia (Java platform) que aísla el programa Java de las especificidades del hardware (hardware-based platform).



La Máquina Virtual Java

La Máquina Virtual Java (JVM) es el entorno en el que se ejecutan los programas Java, su misión principal es la de garantizar la portabilidad de las aplicaciones Java. Define esencialmente un ordenador abstracto y especifica las instrucciones (*bytecodes*) que este ordenador puede ejecutar. El intérprete Java específico ejecuta las instrucciones que se guardan en los archivos cuya extensión es **.class**. Las tareas principales de la JVM son las siguientes:

- Reservar espacio en memoria para los objetos creados
- Liberar la memoria no usada (garbage collection).
- Asignar variables a registros y pilas
- Llamar al sistema huésped para ciertas funciones, como los accesos a los dispositivos
- Vigilar el cumplimiento de las normas de seguridad de las aplicaciones Java

Esta última tarea, es una de las más importantes que realiza la JVM. Además, las propias especificaciones del lenguaje Java contribuyen extraordinariamente a este objetivo:

- Las referencias a arrays son verificadas en el momento de la ejecución del programa
- No hay manera de manipular de forma directa los punteros
- La JVM gestiona automáticamente el uso de la memoria, de modo que no queden huecos.
- No se permiten realizar ciertas conversiones (casting) entre distintos tipos de datos.

Por ejemplo, cuando el navegador encuentra una página web con un applet, pone en marcha la JVM y proporciona la información que aparece en la etiqueta `<APPLET > ... </APPLET>`. El cargador de clases dentro de la JVM ve que clases necesita el applet. Dentro del proceso de carga, las clases se examinan mediante un verificador que asegura que las clases contienen código válido y no malicioso. Finalmente, se ejecuta el applet.

El lenguaje Java

El lenguaje Java no está diseñado solamente para crear applets que corren en la ventana del navegador. Java es un lenguaje de propósito general, de alto nivel y orientado a objetos.

Java es un lenguaje de programación orientado a objetos puro, en el sentido de que no hay ninguna variable, función o constante que no esté dentro de una clase. Se accede a los miembros de datos y las funciones miembro a través de los objetos y de las clases. Por

razones de eficiencia, se han conservado los tipos básicos de datos, **int**, **float**, **double**, **char**, etc, similares a los del lenguaje C/C++.

Los tipos de programas más comunes que se pueden hacer con Java son los applets (se ejecutan en el navegador de la máquina cliente) y las aplicaciones (programas que se ejecutan directamente en la JVM). Otro tipo especial de programa se denomina servlet que es similar a los applets pero se ejecutan en los servidores Java.

La API de Java es muy rica, está formada por un conjunto de paquetes de clases que le proporcionan una gran funcionalidad. El núcleo de la API viene con cada una de las implementaciones de la JVM:

- Lo esencial: tipos de datos, clases y objetos, arrays, cadenas de caracteres (strings), subprocesos (threads), entrada/salida, propiedades del sistema, etc.
- Applets
- Manejo de la red (networking)
- Internacionalización
- Seguridad
- Componentes (JavaBeans)
- Persistencia (Object serialization)
- Conexión a bases de datos (JDBC)

Java proporciona también extensiones, por ejemplo, define un API para 3D, para la edición y reproducción de audio y vídeo, Java Media Framework, para los servidores, telefonía, reconocimiento de voz,