

Introducción al lenguaje Java

Programación Orientada a Objetos
Facultad de Informática

Juan Pavón Mestras
Dep. Sistemas Informáticos y Programación
Universidad Complutense Madrid



¿Qué es Java?

- *brewed coffee*
la bebida preferida de los programadores
(al menos en USA)



Historia de Java

- Abril 1991: Proyecto Green (Sun) software para dispositivos electrónicos inteligentes y televisión interactiva. Se trataba de desarrollar un entorno de operación fiable, portable y eficiente.
 - El resultado: Lenguaje Oak (James Gosling)
- Principio 1994: El equipo se reduce ante el escaso crecimiento del mercado PDA y STB, y tras perder acuerdos con Time-Warner y 3DO, se reorienta hacia WWW:
 - Los requisitos del software son los mismos en WWW
- 1994: WebRunner (luego HotJava): navegador Web escrito en Java
- 23 mayo 1995 (SunWorld '95): Anuncio de la release alpha de Java
 - Java se distribuye gratuitamente
- Finales 1995: Primera integración Netscape+Java

¿Qué es Java?

- Un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems
 - Encapsulación, herencia y polimorfismo
 - Fuertemente tipado
 - Gestión automática de la memoria (recogida de basura)
 - Soporte para concurrencia (multihilo)
 - Gestión de excepciones
 - Constructores independientes de la arquitectura del procesador

¿Qué es Java?

- ... acompañado por un conjunto de bibliotecas (packages, conjuntos de clases)
 - Desarrollo de interfaces gráficas de usuario (AWT, Swing)
 - Conectividad (sockets, RMI, IDL, JDBC)
 - Utilidades (Vector, Stack, Hashtable, Date, etc.)
 - Funciones matemáticas
 - Gráficos 2D y 3D
 - Interacción y multimedia (Media, JMF, Speech, Animation, Collaboration, Telephony)
 - Etcétera, etcétera, ...

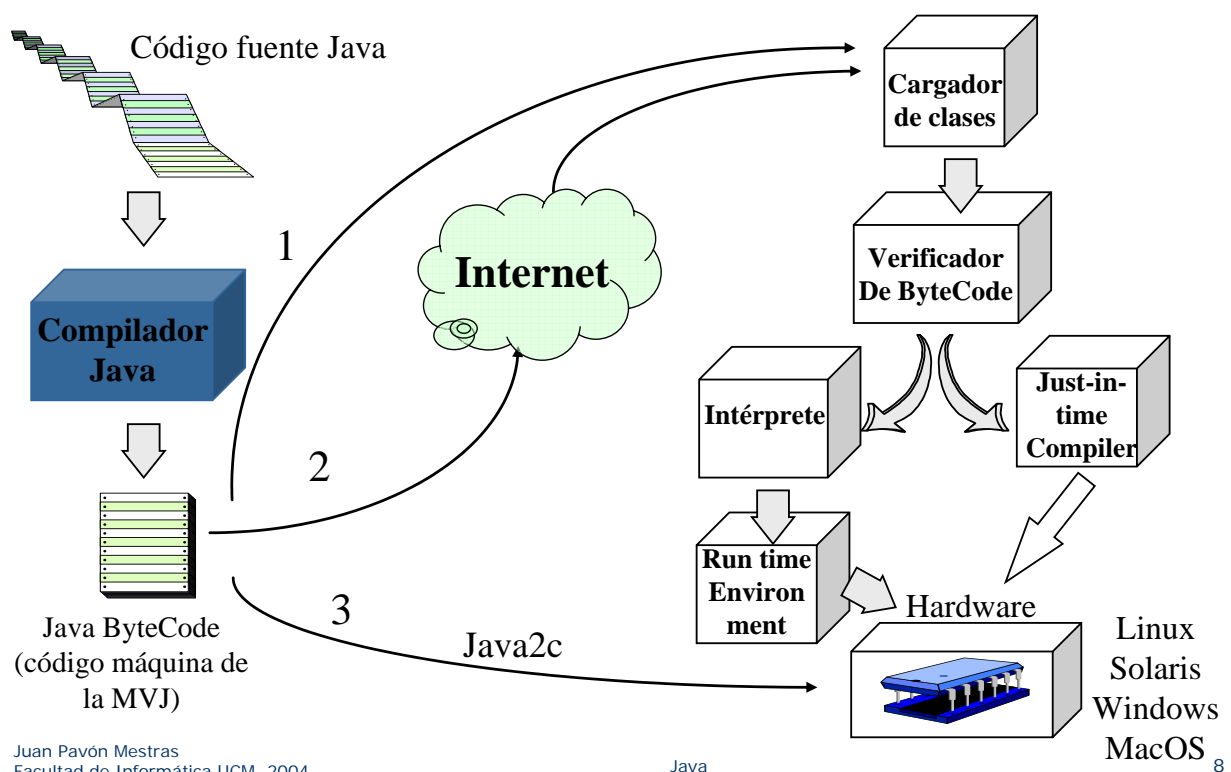
¿Qué es Java?

- Apropriado para Internet
 - Ejecutable en múltiples plataformas
 - Fácil de distribuir
 - Seguridad
 - Soporte para interacción con el usuario:
 - Gráficos
 - Tipos de datos arbitrarios
 - Descarga el servidor

¿Qué es Java?

- Según Sun:
 - Sencillo, Orientado a objetos, y Familiar
 - Robusto y Seguro
 - Independiente de la arquitectura y portable
 - Alto rendimiento
 - Interpretado, Concurrente, y Dinámico

Arquitectura de Máquina Virtual Java



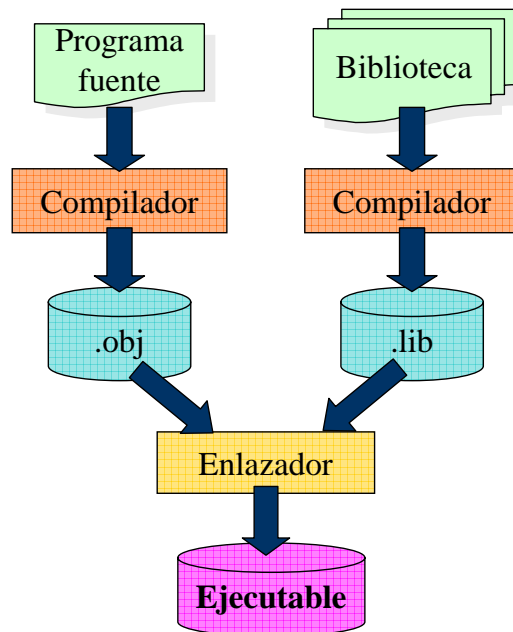
- Un nuevo paradigma de software
 - Software bajo demanda:
 - El software se carga de la red
 - cuando es necesario
 - lo que es estrictamente necesario
 - sin necesidad de instalación
 - Se puede controlar el uso del software
 - No hacen falta actualizaciones software individuales
 - Las versiones nuevas reemplazan a las anteriores de forma invisible
 - No hay problemas de configuración específicos del HW
 - Codifica una vez, ejecuta en varias plataformas

La máquina virtual de Java (MVJ)

- Java se ejecuta en un computador simulado
 - Portabilidad
 - Protección
 - Para portar todo el código Java, portar la MVJ
- Otras arquitecturas de MV:
 - UCSD P-system, Apple Newton, Smalltalk, Emacs, ...
- La máquina virtual Java es sofisticada:
 - Multihilo (Threads)
 - Recogida de basura

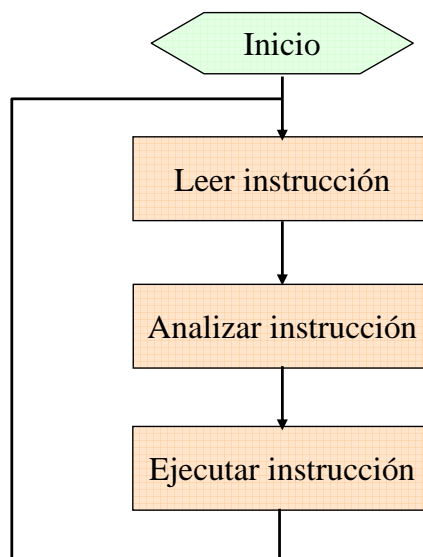
La máquina virtual de Java (MVJ)

■ Entorno de compilación y ejecución tradicional



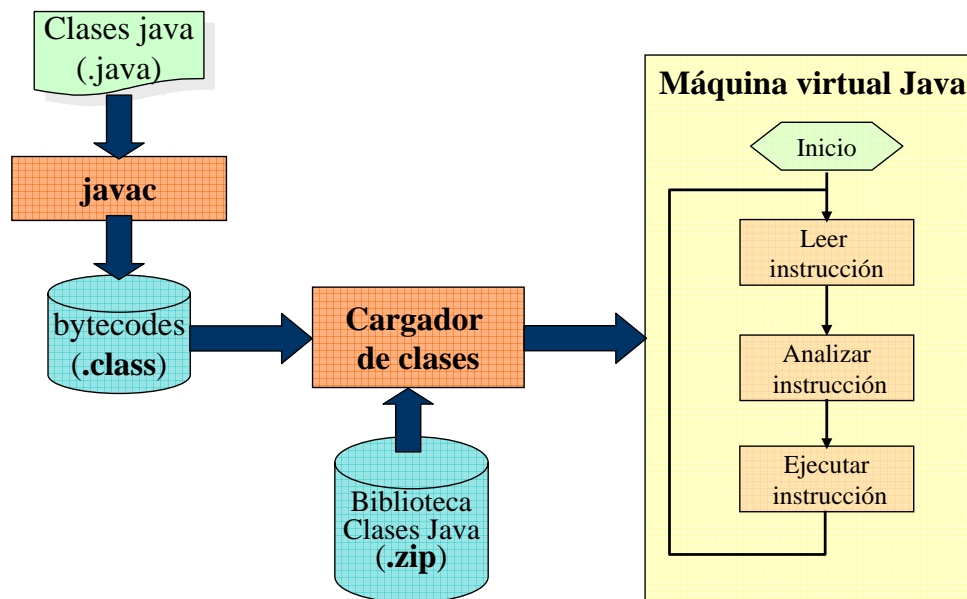
La máquina virtual de Java (MVJ)

■ Intérpretes



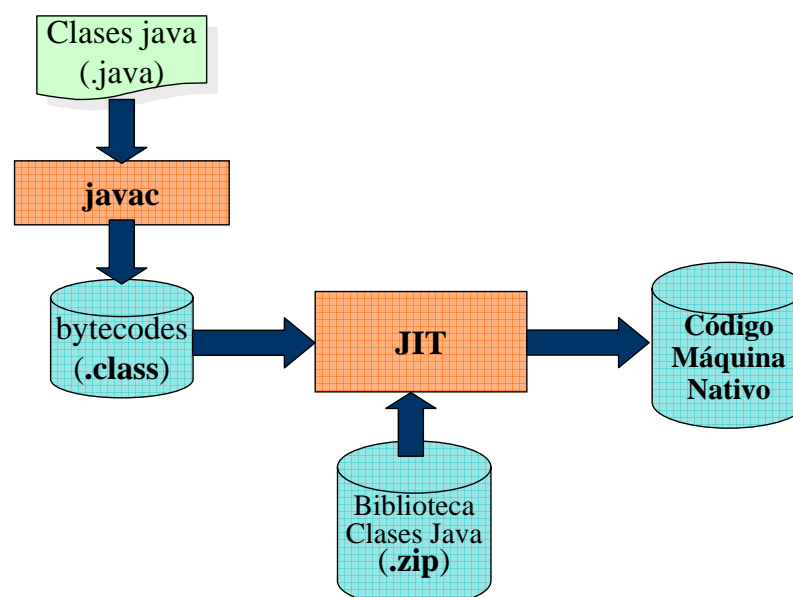
La máquina virtual de Java (MVJ)

- El intérprete de bytecodes de Java



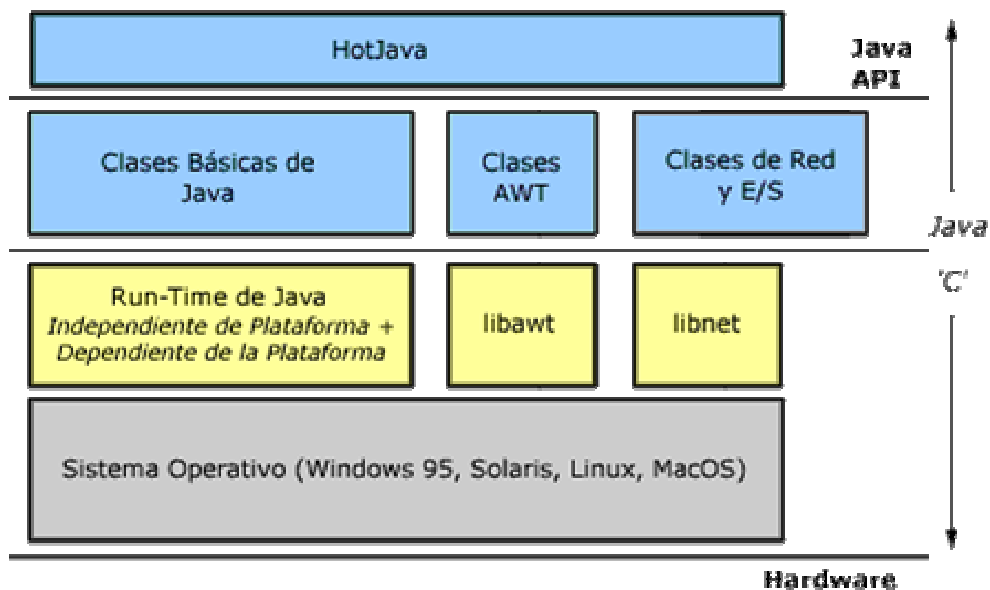
La máquina virtual de Java (MVJ)

- Compilador Just-In-Time (JIT)



La máquina virtual de Java (MVJ)

■ Arquitectura Java sobre un SO



Sistema de ejecución de Java

- Facilidades básicas
 - Class Object
 - Class Thread, Interface Runnable
 - Class Throwable, Class Exception
- Abstract Window Toolkit
 - Class Component
 - Class Graphics
 - Class Applet

Sistema de ejecución de Java

- E/S y comunicaciones
 - Class InputStream y OutputStream
 - Encadenamiento y herencia de flujos (Stream)
 - Class URL
 - Class Socket
- Otras utilidades
 - Class String
 - Colecciones: Vector, Stack, HashTable
- Recogida de basura
- Comprobación de rangos

Java Development Kit (JDK)

- Bibliotecas de clases Java
- Herramientas básicas
 - javac -- compilador Java -> bytecodes
 - java -- intérprete Java
 - jdb -- depurador
 - javah -- crea ficheros de cabecera y stubs para conectividad con C
 - javap -- desensamblador bytecodes -> Java
 - javadoc -- genera documentación de API en formato HTML a partir de código fuente Java
 - appletviewer -- permite ejecutar applets sin un navegador

- Herramientas RMI
 - `rmic` -- genera clases stub y skeleton para objetos Java que implementan la interfaz `java.rmi.Remote`
 - `rmiregistry` -- arranca un objeto registrador en un puerto dado
 - `serialver` -- devuelve el `serialVersionUID` para una o más clases
- Herramientas de internacionalización
 - `native2ascii` -- convierte ficheros no-Unicode Latin-1 a éste
- Herramienta JAR
 - `jar` -- combina varios ficheros en un fichero Java Archive (JAR)
- Herramienta de firma digital
 - `javakey` -- genera firmas digitales

Versiones de Sun JDK/J2SE

- 23 de mayo de 1995: James Gosling presenta Java
- Java Development Kit (JDK) 1.0.2 [Mayo 1996]
 - Core API: `java.applet`, `java.awt`, `java.io`, `java.lang`, `java.net`, `java.util`
 - Herramientas: `javac`, `java`, `appletviewer`, `jdb`
 - Applets ejecutables en Netscape 3.0
- JDK 1.1.x [1997/98]
 - Añade: internacionalización, seguridad, JavaBeans, JAR, RMI, serialización de objetos, JDBC, JNI, y mejoras a las bibliotecas anteriores
 - Herramientas: `javac`, `java`, `appletviewer`, `jdb`
 - Applets ejecutables en HotJava 1.x y Netscape 4.0
- Java 2 SDK, Standard Edition (J2SE) 1.2 [Dic. 1998], 1.3 [Mayo 2000], 1.4 [Marzo 2002]
 - Swing, 3D, ...
 - Incorpora todas las facilidades para la programación con componentes

Versiones de Sun JDK

- J2SE 5.0 ("Tiger") [30 sep. 2004]

<http://java.sun.com/developer/technicalArticles/releases/j2se15/>

- Añadidos al lenguaje: metadatos, tipos enumerados, tipos genéricos, argumentos variables, utilidades de concurrencia
- Más control y gestión de la MVJ (también gestión remota)
- Mayor eficiencia y estabilidad
 - Menor tiempo de arranque
- Un nuevo look & feel del GUI de Java

Entornos de desarrollo integrados (IDE)

- En casi todos los sistemas operativos: Unix (Solaris, Linux), Windows, Mac
 - Comerciales
 - Sun Java Workshop, Visual Café (Symantec), JBuilder (Borland), Metrowerks CodeWarrior, etc.
 - Libre distribución
 - GNU Java, Kawa
- *Y mi recomendación:* www.eclipse.org

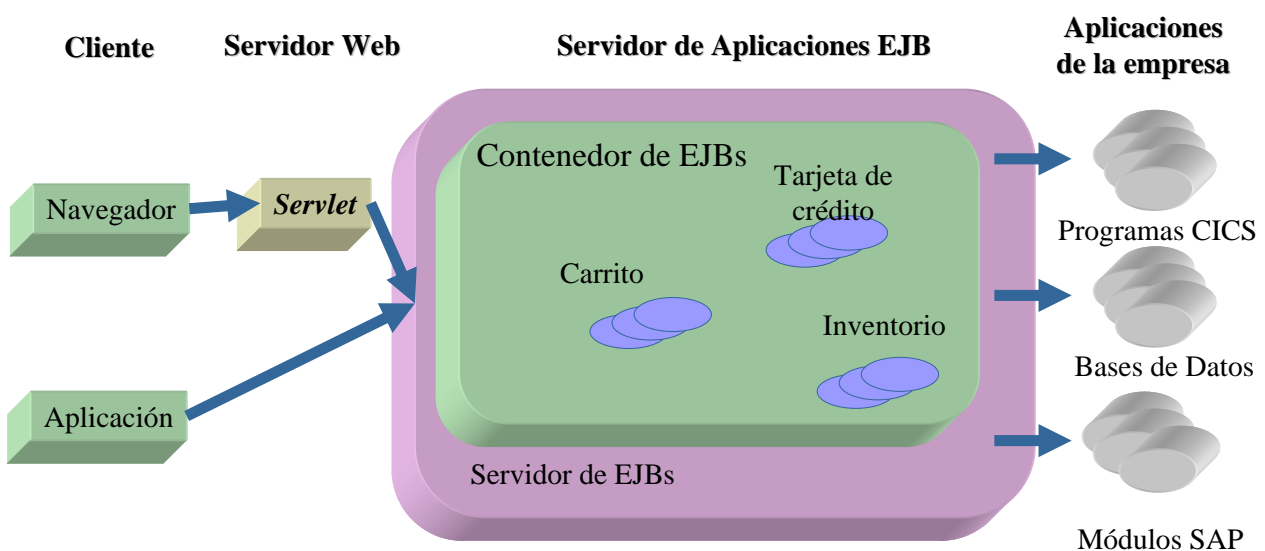


Tecnologías Java

- El 15 de junio 1999 Sun anuncia 3 ediciones de la plataforma Java:
 - **J2SE:** Edición estándar
 - Para desarrollar aplicaciones y applets
 - **J2ME:** Java Micro Edition
 - Aplicaciones para dispositivos móviles
 - Midlets
 - **J2EE:** Java Enterprise Edition
 - Aplicaciones para la web
 - Componentes para servidores de aplicaciones
 - Servlets, JSPs, EJBs, ...

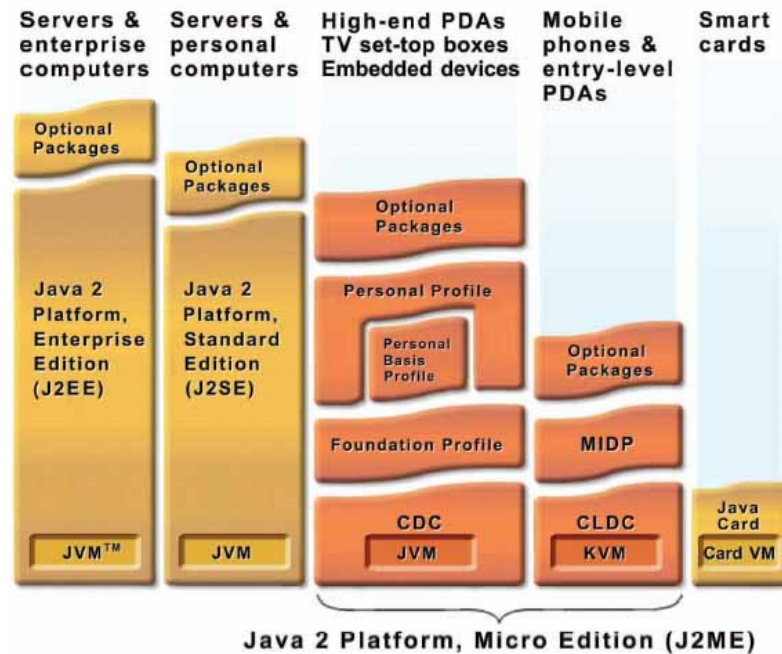
Ingeniería de componentes

- J2EE: Java Enterprise Edition
 - Servidores web con tecnología de componentes soportada por el lenguaje Java



Dispositivos móviles

- J2ME: Java 2 Micro Edition
 - Orientada a pequeños dispositivos y sistemas embebidos (teléfonos móviles, PDAs, Set-Top Boxes, ...)



Juan Pavón Mestras
Facultad de Informática UCM, 2004

25



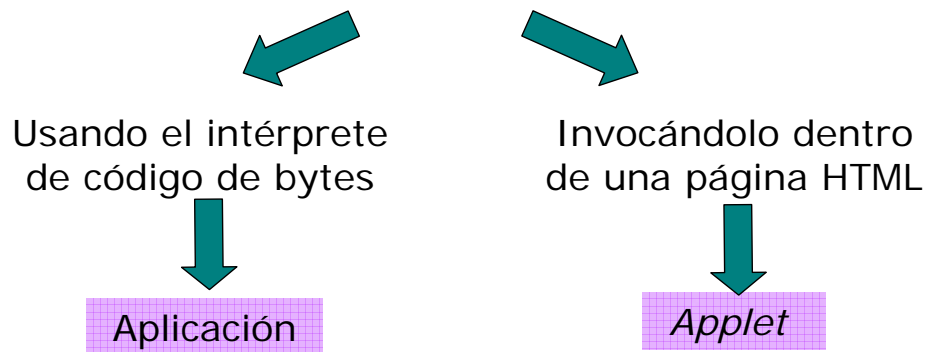
Juan Pavón Mestras
Facultad de Informática UCM, 2004

Java

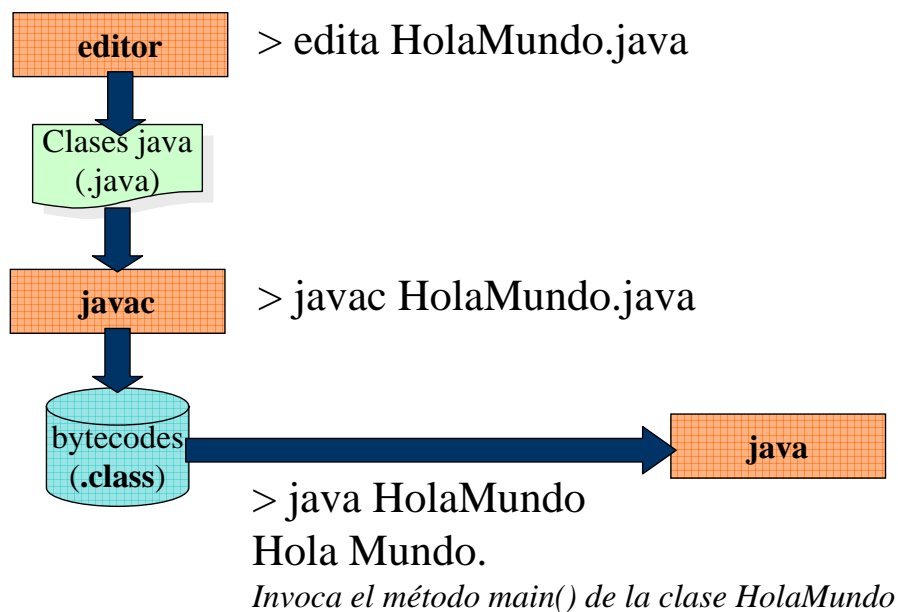
26

Cómo realizar un programa Java

- 1) Editar el fuente
- 2) Compilar
- 3) Ejecutarlo



Cómo realizar una aplicación Java



Cómo realizar un aplicación Java

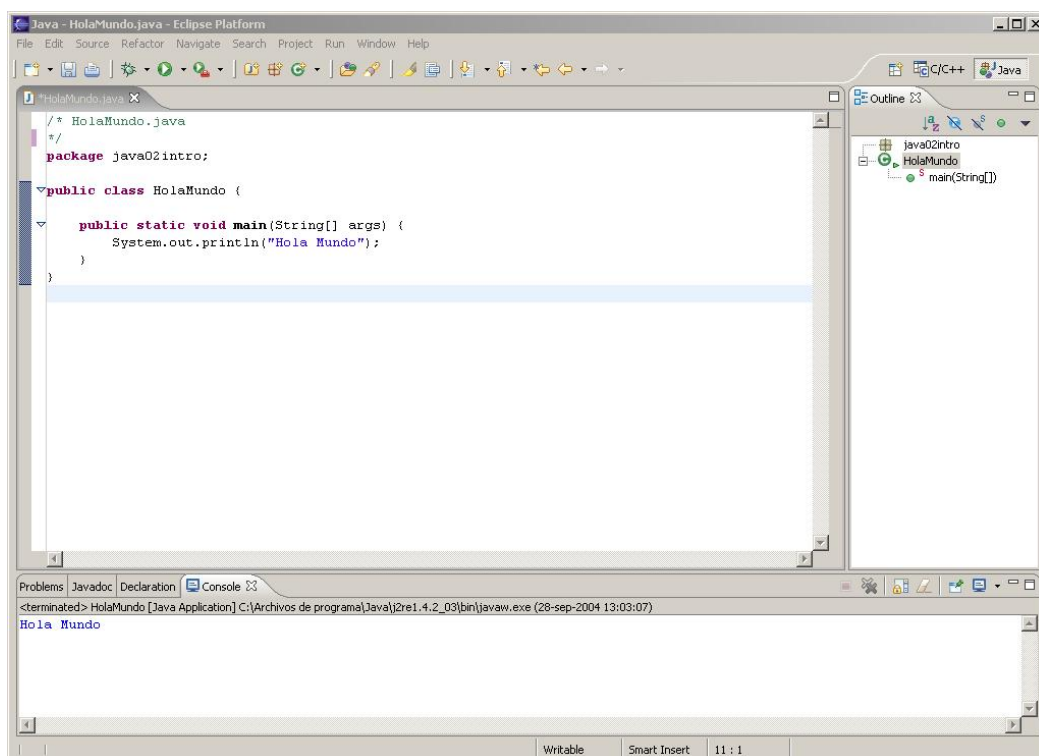
```
public class HolaMundo {  
    public static void main(String[] args) {  
        System.out.println("Hola Mundo");  
    }  
}
```

> edita HolaMundo.java
> javac HolaMundo.java

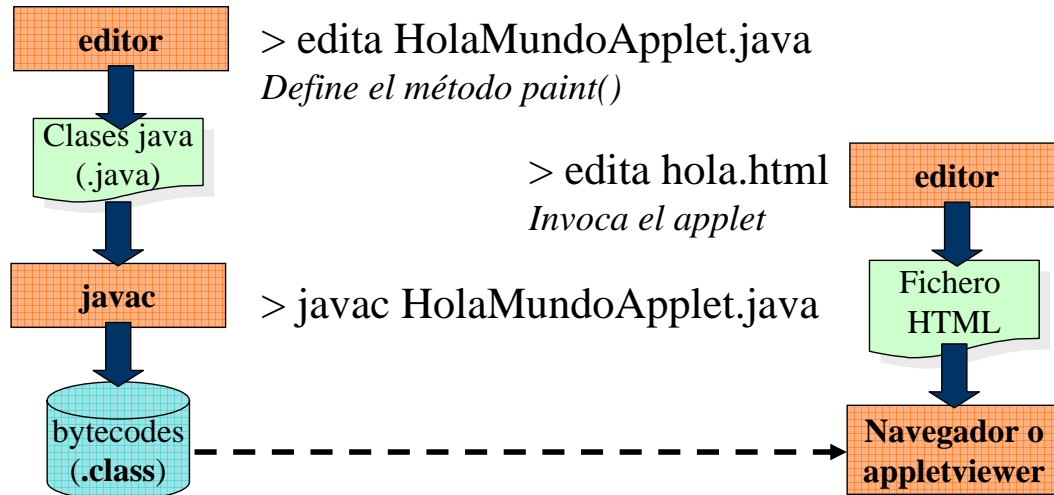
> java HolaMundo
Hola Mundo.

Invoca el método main() de la clase HolaMundo

Más fácil con Eclipse



Cómo realizar un applet Java



Cómo realizar un applet Java

Hola.html

```
<HTML>
<HEAD>
<TITLE> Hola Mundo</TITLE>
</HEAD>
<BODY>
Saluda amigablemente:
<APPLET
CODE="HolaMundoApplet.class"
WIDTH=150 HEIGHT=125>
</BODY>
</HTML>
```

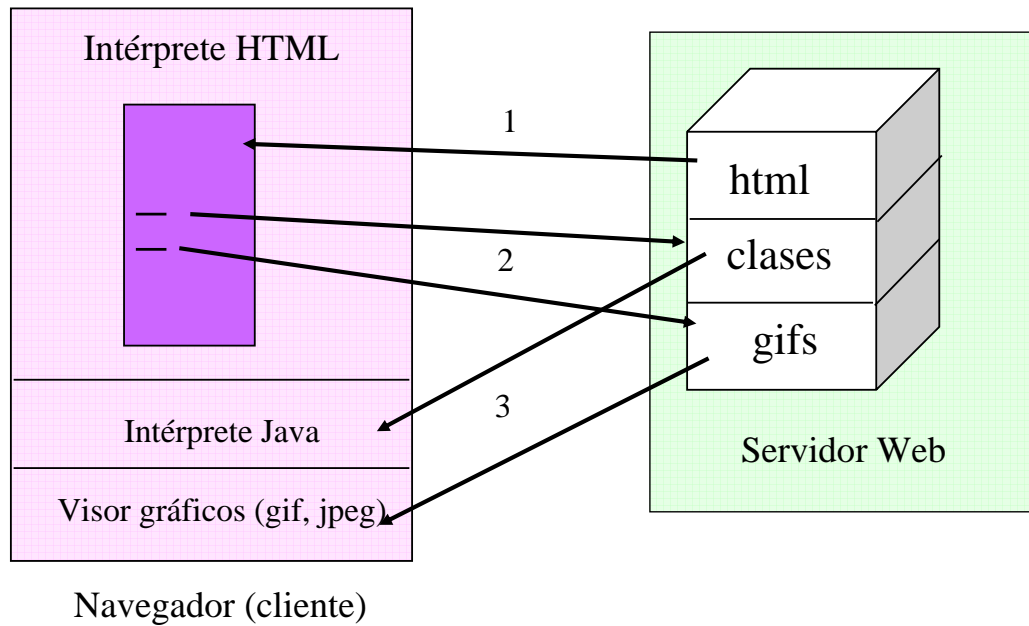
HolaMundoApplet.java

```
import java.applet.Applet;
import java.awt.Graphics;

public class HolaMundoApplet extends Applet
{
    public void init() {
        resize (150,25);
    }

    public void paint (Graphics g) {
        g.drawString ("Hola Mundo", 50,25);
    }
}
```


Cómo se ejecuta un applet Java



Java y C++

- Java se parece a C++
 - Sin algunas cosas
 - Punteros, direcciones, estructuras (struct y union), preprocesado, herencia múltiple, sobrecarga de operadores, tipos enumerados (enum), tipos sin signo (unsigned), goto, variables y funciones globales
 - Más seguro
 - Recogida de basura, interfaces, paquetes, arrays con límites, soporte para concurrencia y sincronización
 - Aspectos recientes de C++
 - Excepciones uniformes
 - Información de tipos en tiempo de ejecución
 - Tipo boolean
 - Casting seguro

- El código Java es más sencillo
 - recogida de basura
 - Las facilidades en tiempo de ejecución más avanzadas
 - Las interfaces simplifican la jerarquía de herencia
- El código Java es más robusto
 - Verificación al cargar y comprobación en tiempo de ejecución
 - Arrays seguros
 - Compilador estricto e inteligente
- Menos oportunidades para los más audaces
 - Imposible realizar aritmética de punteros
 - No es posible hacer mejoras específicas para una plataforma
- No es cierto eso de que las aplicaciones Java tengan que ser más lentas que las C++