
Tema 2: La Tecnología Java

La Tecnología de Java

Es :

- Un lenguaje de programación

Similar a C/C++ sin los problemas de manejo de memoria.

- Un ambiente de desarrollo

Compilador, intérprete, generados de documento, herramientas de empaquetamiento.

- Un ambiente de aplicación

Código portable

- Un ambiente de funcionalidad (deployment)

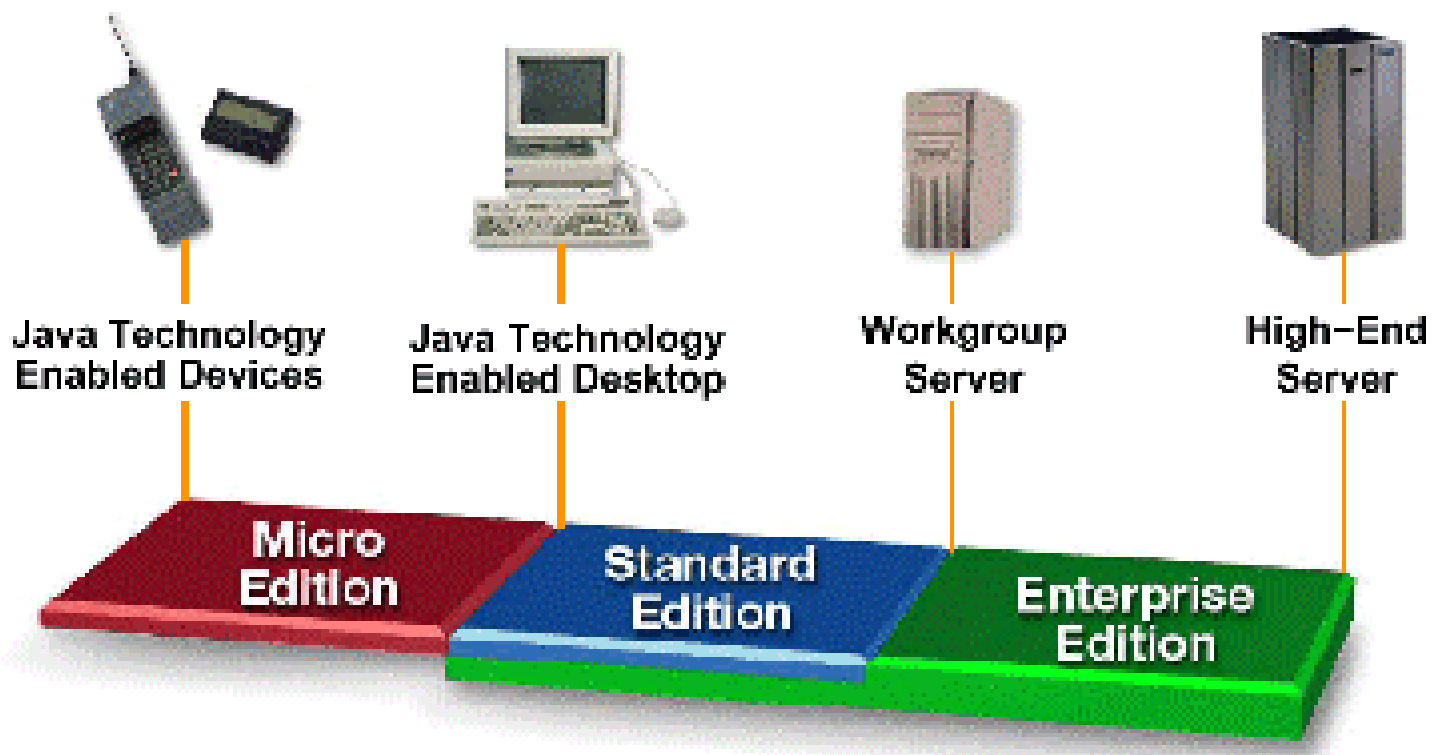
EL SDK (Software Developer's Kit) o JDK (Java Development Kit), JRE (Java Runtime Environment), el soporte en web.



Grupos de la Tecnología de Java

- J2SE™, Java™ 2 Platform Standard Edition.
 - Creado para applets y aplicaciones que corren sin navegadores web y en una computadora respectivamente.
- J2EE™, Java™ 2 Platform Enterprise Edition.
 - Para aplicaciones distribuidas del lado del cliente y del servidor; aplicaciones web en un site (entornos colaborativos) respectivamente.
- J2ME™, Java™ 2 Platform Micro Edition.
 - Para crear aplicaciones en dispositivos con recursos limitados como los PDA's o los teléfonos móviles.

Grupos de la Tecnología de Java



J2SE

- El SDK está incluido en cada edición y permite a los programadores. Crear, compilar y ejecutar programas en Java.
- El J2SE SDK components incluye:
 - ✓ JRE (1.5) :
 - ✓ JVM
 - ✓ Java class libraries
 - ✓ Compilador Java
 - ✓ Documentación para API
 - ✓ Utilerías adicionales (ejem: crear archivos *.jar*).
 - ✓ Ejemplos.



Versiones de J2SE

- Java 1
 - Java 1.0 (Enero 1996) :- 8 paquetes, 212 clases.
 - Java 1.1 (Marzo 1997) - 23 paquetes, 504 clases.
- Java 2
 - Java 1.2 (Diciembre 1998) - 59 paquetes, 1520 clases
 - Java 1.3 (Abril 2000) - 77 paquetes, 1595 clases
 - Java 1.4 (2002) - 103 paquetes, 2175 clases - También conocido como Merlin
 - Java 1.5 (Octubre 2004) - 131 paquetes, 2656 clases - También conocido como Tiger
 - Actualmente se esta trabajando sobre Java 1.6 llamado Mustang.

Versión 1.5

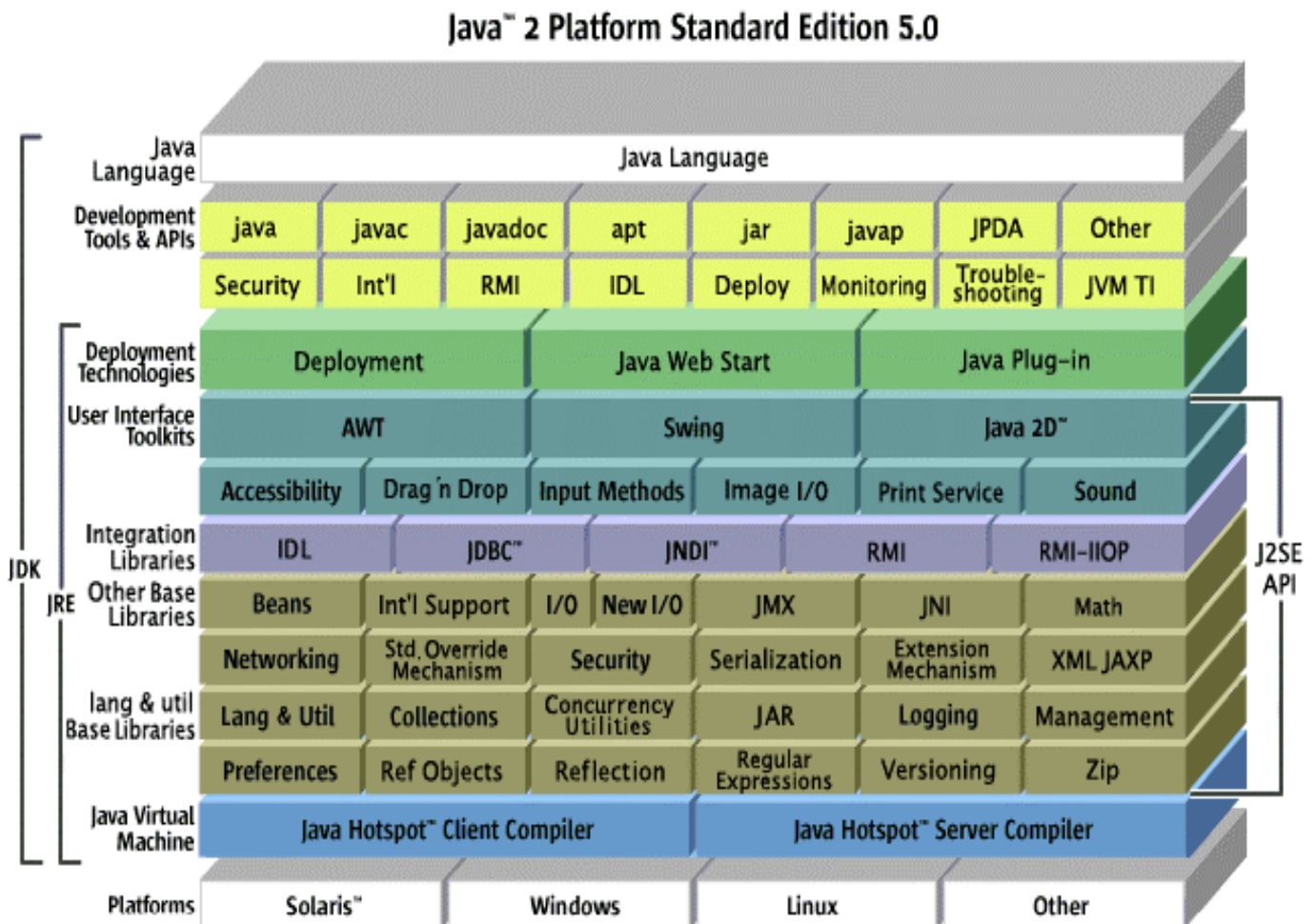
- Añade las siguientes características:
 - ✓ Mecanismos de autoboxing y unboxing de tipos primitivos.
 - ✓ Mejora a la sentencia de control for.
 - ✓ Métodos con número de argumentos variable (*varargs*).
 - ✓ Enumeraciones.
 - ✓ Generics (templates de C++) proporcionan seguridad de tipos en tiempo de compilación.
 - ✓ Imports estáticos.
 - ✓ Metadatos.
 - ✓ Nuevo método (printf) para generar salida por consola.

La arquitectura Java

- Una arquitectura o plataforma es un ambiente de SW o HW sobre el cuál se ejecuta un programa.
- Esta plataforma es de SW ya que se ejecuta sobre otra plataforma que es el sistema operativo.
- La plataforma java consta de:
 - ✓ La JVM, Java Virtual Machine.
 - ✓ El Garbage Collector
 - ✓ EL JRE y
 - ✓ La interfaz de herramientas de JVM.

J2SE 5.0

- J2SE™, Java™ 2 Platform Standard Edition 5.0.

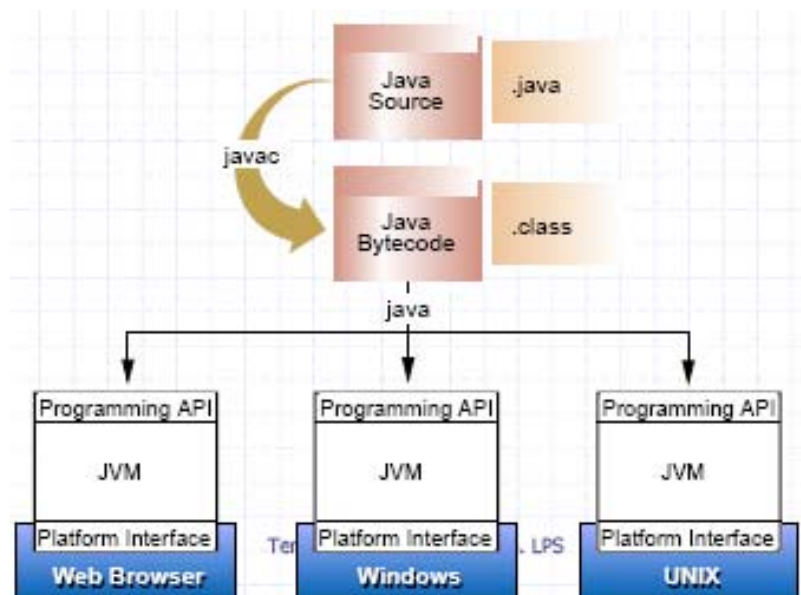


JRE

- Java™ Runtime Environment es el entorno de ejecución de Java.
- Este paquete de SW contiene herramientas para ejecutar programas java, previamente compilados:
 - ✓ The Java platform core classes.
 - ✓ The Java™ virtual machine.
 - ✓ The Java™ technology plug-in.

La JVM

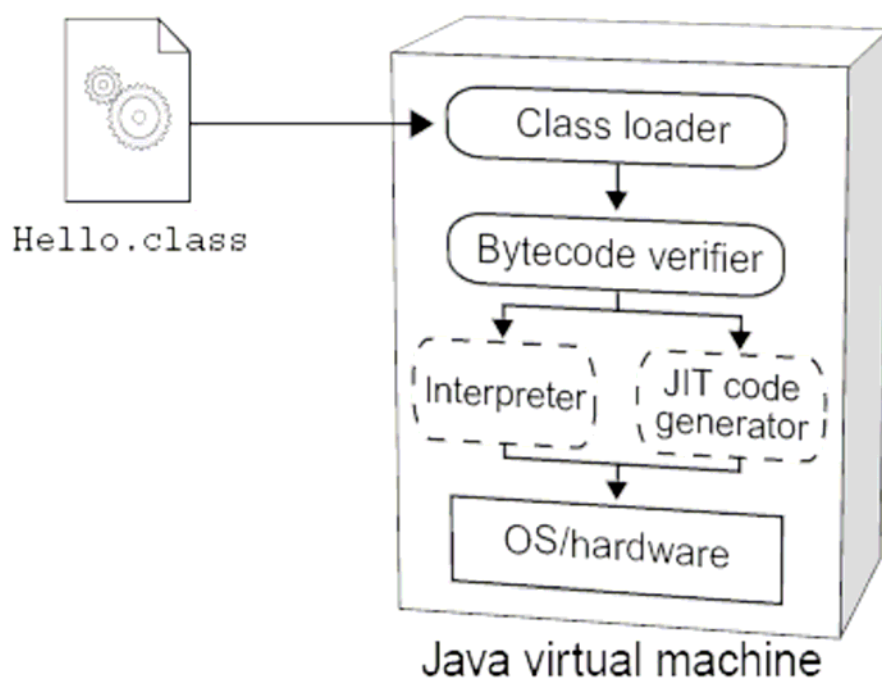
- La especificación de la JVM la define como: “Una maquina imaginaria que es implementada para emularla en su software dentro de una máquina real. El código para la JVM es almacenado en los archivos *.class*, cada uno de los cuales contiene código para al menos una clase pública”.
- Contiene el interprete de Java.



La JVM

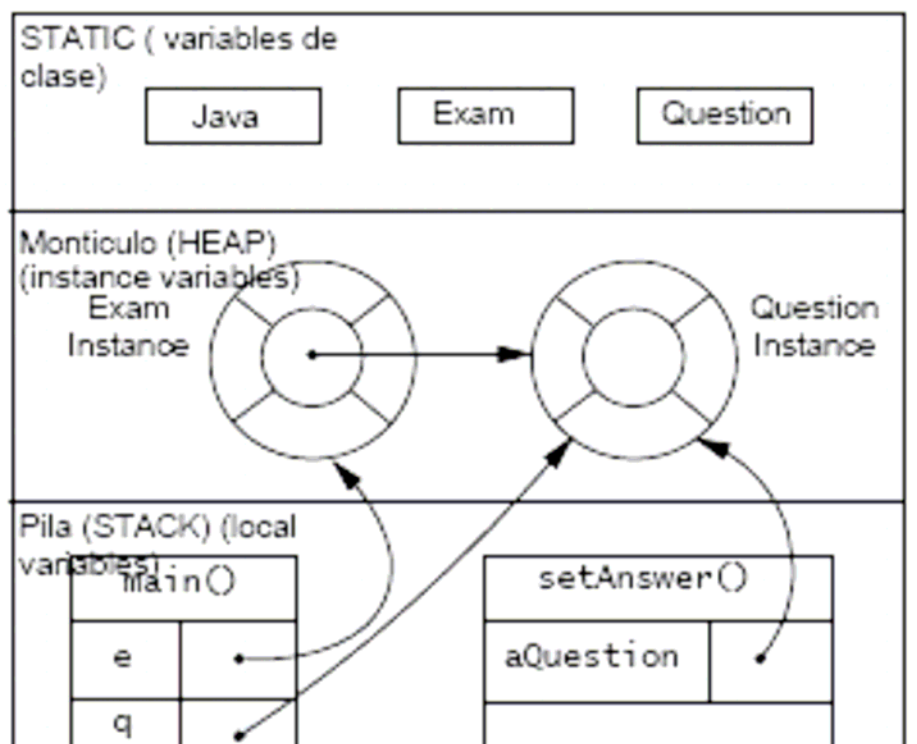
Las 3 principales tareas de la JVM son:

- ✓ Cargar del código.
- ✓ Verificar el Código.
- ✓ Executar el código.



JVM

- La especificación de JVM también proporciona definiciones concretas para la implementación, como:
 - ✓ Registros.
 - ✓ La pila o stack.
 - ✓ El montículo del Garbage Collector.
 - ✓ El área de métodos.
 - ✓ El conjunto de instrucciones.



El Garbage Collector

- La recolección y liberación de memoria es responsabilidad de un thread llamado garbage collector (recolector automático de basura).
- El trabajo del GC es identificar objetos que no son usados y no requieren memoria.
- Se ejecuta automáticamente.



El Garbage Collector

- Un objeto es elegible para el GC cuando ya han perdido la referencia, es decir, objetos que ya no tienen ningún nombre que permita acceder a ellos, por ejemplo:
 - ✓ Por haber llegado al final del bloque en el que habían sido definidos.
 - ✓ Porque a la referencia se la ha asignado el valor null.
 - ✓ Porque a la referencia se la ha asignado la dirección de otro objeto.

El Garbage Collector

- Nota importante:
- La JVM se hace cargo de iniciar el proceso de "Garbage Collection" *únicamente* cuando éste (según sus algoritmos internos) determine que su memoria esta en proceso de agotarse, es hasta entonces que es liberada la memoria de las instancias/referencias que se han declarado como elegibles para el GC.

System.gc()

- Existe un método llamado gc() dentro de las Clases java.lang.System y java.lang.Runtime cuyas iniciales significan "Garbage Collection" y puede ser llamado por el programador directamente dentro de un programa Java.
- Pero el hecho de invocar este método explícitamente no implica que se realice el proceso de "Garbage Collection" *inmediatamente*, lo único que logra es *expeditar* (acelerar) los mecanismos para iniciar la recolección(depende de la JVM).

finalize()

- Este método es miembro de la clase **Object** (por lo que todas las clase lo contienen).
- El objetivo de este método es liberar los recursos del sistema, como archivos o conexiones abiertas antes de empezar la recolección.
- Declaración por default:
`protected void finalize() { }`
Otra declaración:
`protected void finalize () throws throwable`
- Para utilizar este método, hay que sobrecargarlo, proporcionando el código que contenga las acciones que se desee ejecutar antes de liberar la memoria consumida por el objeto.

La API

- API, Application Programming Interface es una colección de componentes de SW que proporcionan una amplia variedad de funcionalidades, como GUI, I/O, etc. Está agrupada en paquetes y librerías de componentes relacionados.
- Página web de la API
[//java.sun.com/j2se/5.0/docs/api/index.html](http://java.sun.com/j2se/5.0/docs/api/index.html)

API en línea

Java™ 2 Platform
Standard Ed. 5.0

[All Classes](#)

Packages

[java.applet](#)

[java.awt](#)

[java.awt.color](#)

[java.awt.datatransfer](#)

[java.awt.dnd](#)

[java.awt.event](#)

[java.awt.font](#)

All Classes

[AbstractAction](#)

[AbstractBorder](#)

[AbstractButton](#)

[AbstractCellEditor](#)

[AbstractCollection](#)

[AbstractColorChooserPane](#)

[AbstractDocument](#)

[AbstractDocument.Attribute](#)

[AbstractDocument.Content](#)

[AbstractDocument.Element](#)

[AbstractExecutorService](#)

[AbstractInterruptibleChannel](#)

[AbstractLayoutCache](#)

[AbstractLayoutCache.Node](#)

[AbstractList](#)

Overview Package Class Use Tree Deprecated Index Help

PREV NEXT

FRAMES NO FRAMES

Java™ 2 Platform
Standard

Java™ 2 Platform Standard Edition 5.0
API Specification

This document is the API specification for the Java 2 Platform Standard Edition 5.0.

See:
[Description](#)

Java 2 Platform Packages

java.applet	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
java.awt	Contains all of the classes for creating user interfaces and for graphics and images.
java.awt.color	Provides classes for color spaces.
java.awt.datatransfer	Provides interfaces and classes for transferring data between applications within applications.
java.awt.dnd	Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.
java.awt.event	Provides interfaces and classes for dealing with different types

Paquetes principales

java.lang - clases esenciales, números, strings, objetos, compilador, runtime, seguridad y threads (es el único paquete que se incluye automáticamente en todo programa Java)

java.io - clases que manejan entradas y salidas

java.util - clases útiles, como estructuras genéricas, manejo de fecha, hora y strings, número aleatorios, etc.

java.net - clases para soportar redes: URL, TCP, UDP, IP, etc.

java.awt - clases para manejo de interface gráfica, ventanas, etc.

java.awt.image - clases para manejo de imágenes

java.sql- clases para el manejo de bases de datos (JDBC,JDBC-ODBC).

java.security-clases e interfaces para el manejo de seguridad (criptografía, firmas digitales, encriptación y autenticación).

java.applet - clases para la creación de applets y recursos para reproducción de audio.

IDE's

- Los IDE's ("Integrated Development Environment") ofrecen un ambiente gráfico en los que se tiene acceso a mayor número de herramientas no ofrecidas en los JDK's:
 - ✓ Debuggers más elaborados.
 - ✓ Check-points dentro de la compilación.
 - ✓ Creación de WAR's (Web-Archives).
 - ✓ "Wizards" para acelerar desarrollo, etc

IDE's

- Algunos IDE's :
 - ✓ **NetBeans**, www.netbeans.org, Open-Source.
 - ✓ **Eclipse**, www.eclipse.org, Open-Source.
 - ✓ **Sun Java Studio Enterprise**,
www.sun.com/software/sundev/jde/index.xml
 - ✓ **JBuilder**, www.borland.com/jbuilder, de Borland.
 - ✓ **WebSphere Studio**, <http://www-306.ibm.com/software/info1/websphere/index.jsp?tab=products/studio>, de IBM.
 - ✓ **JDeveloper**,
[//otn.oracle.com/products/jdev/content.html](http://otn.oracle.com/products/jdev/content.html), de Oracle