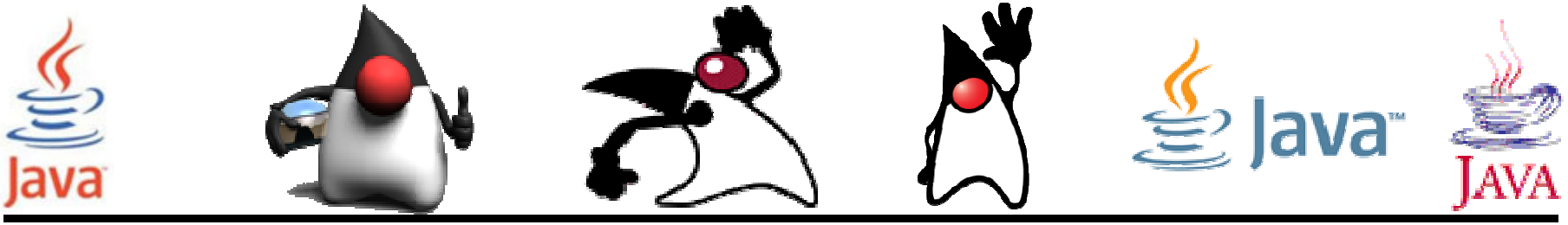


Introducción a la Tecnología Java

Pedro Corcuera

Dpto. Matemática Aplicada y Ciencias de la Computación

Universidad de Cantabria

corcuerp@unican.es

Objetivos

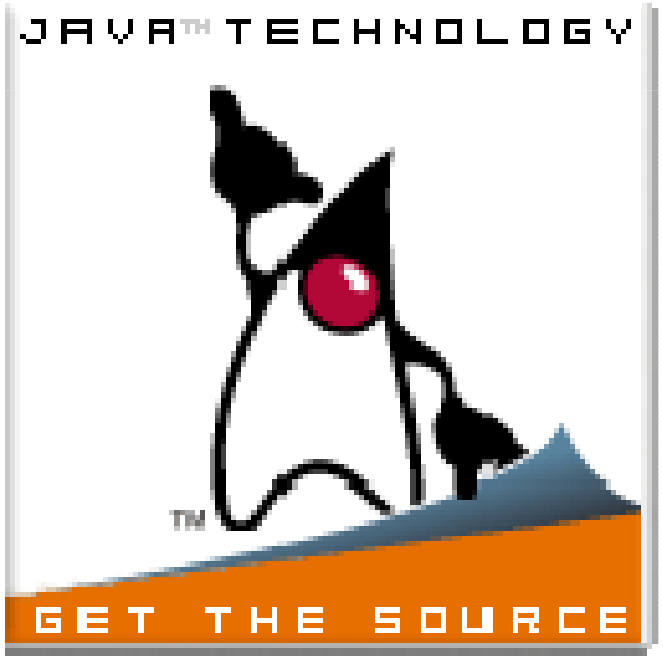
- Describir la tecnología Java
- Describir algunos entornos de programación Java y las fases de desarrollo de un programa Java

Índice

- Tecnología Java
- Entornos de desarrollo de Java
  - Java Development Kit (JDK)
  - Entornos IDE (Integrated Development Environment)

¿ Qué es Java ?

- En la industria del software el término "Java" se refiere a la plataforma Java, así como al lenguaje de programación Java.



Java como lenguaje de programación

- El lenguaje Java fue ideado por *James Gosling*, quien comenzó a trabajar en el proyecto en 1991 para la compañía *Sun Microsystems Inc.*
- El propósito era crear un lenguaje que fuera independiente de la plataforma en la que se vaya a ejecutar.
- En 2010 Sun fue adquirido por Oracle.

Java como lenguaje de programación

- El lenguaje fue creado con los siguientes principios de diseño en mente:
  - **Simple**: Java contiene un pequeño núcleo coherente de conceptos fundamentales que puede ser captado con rapidez.
  - **Familiar**: Sigue el modelo del entonces popular lenguaje C y C++, por lo que los programadores podían fácilmente emigrar a Java.
  - **Orientado a objetos**: se adhiere al paradigma orientado a objetos, sistemas se componen de objetos encapsulados que se comunican pasando mensajes entre sí.

Java como lenguaje de programación

- **Robusto y seguro**: El lenguaje incluye comprobación en tiempo de compilación y de ejecución para garantizar que los errores son identificados rápidamente, así como características de seguridad de red y de acceso a archivos de forma que las aplicaciones distribuidas no se vean comprometidas por intrusión o corrupción.
- **Arquitectura neutral y portable**: Una de las ventajas principales de Java es su *portabilidad*. Las aplicaciones se pueden transferir de una plataforma a otra con un mínimo o sin modificaciones. El lema "Write Once, Run Anywhere", en la primera versión de Java en 1995, se refiere a estos beneficios del lenguaje.

Java como lenguaje de programación

- **Alto rendimiento**: Las aplicaciones se ejecutan de forma rápida y eficiente debido a varias características de bajo nivel, tales como permitir que el intérprete Java se ejecute de forma independiente del entorno de tiempo de ejecución, y la aplicación de un recolector de basura automático para liberar la memoria sin utilizar.

Java como lenguaje de programación

- **Interpretado**: El código fuente desarrollado en Java se traduce a un formato intermedio interpretado, conocida como *bytecode*. El conjunto de instrucciones bytecode hace referencia al lenguaje máquina utilizado por la máquina virtual Java (JVM). Con un intérprete adecuado, este lenguaje puede ser traducido a código nativo para la plataforma en la que se ejecuta.

Java como lenguaje de programación

- **Multitarea (Multithreaded)**: La capacidad de subprocesos múltiples o hilos de ejecución se realiza gracias a la clase *Thread*, que permite que numerosas tareas se realicen de forma simultánea.
- **Dinámico**: El idioma y el sistema en tiempo de ejecución son dinámicos en tanto que las aplicaciones pueden adaptarse a los cambios del entorno durante la ejecución. Otro aspecto es que no es necesario cargar completamente el programa en memoria sino que las clases compiladas pueden ser cargadas bajo demanda en tiempo de ejecución (dynamic binding).

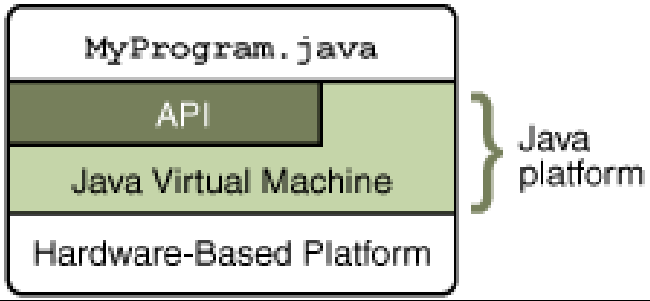
Java como plataforma

- La Plataforma Java para el desarrollo de software se compone de dos partes:
  - La **Java Virtual Machine** (JVM): es un motor que ejecuta las instrucciones generadas por el compilador Java. La JVM se puede considerar como una instancia del JRE (Java Runtime Environment) y está integrada en diversos productos, tales como navegadores web, servidores y sistemas operativos.

Java como plataforma

- La **Java Application Programming Interface** (API): código preescrito, organizados en paquetes de temas similares. Por ejemplo, los paquetes applet y AWT incluyen clases para crear fuentes, menús y botones.

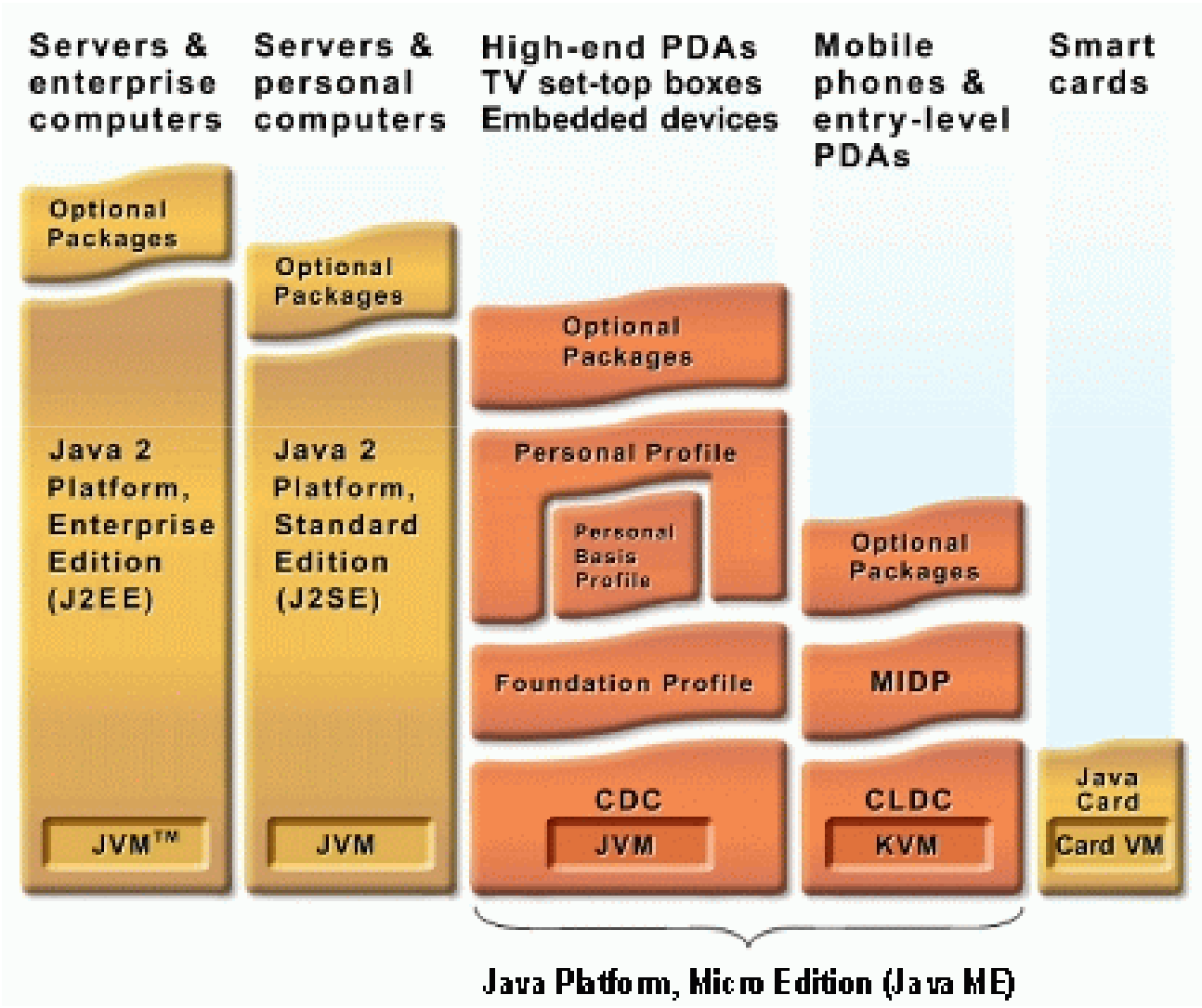
- El Java Development Kit o JDK, se refiere a la edición de Java SE, mientras que otros kits se conocen como "SDK", un término genérico para "kit de desarrollo de software." Por ejemplo, el Java EE SDK.



Plataformas Java

- Hay cuatro plataformas del lenguaje de programación Java:
  - Java Platform, Standard Edition (Java SE)
  - Java Platform, Enterprise Edition (Java EE)
  - Plataforma Java, Edición Micro (Java ME)
  - Java FX

La Plataforma Java



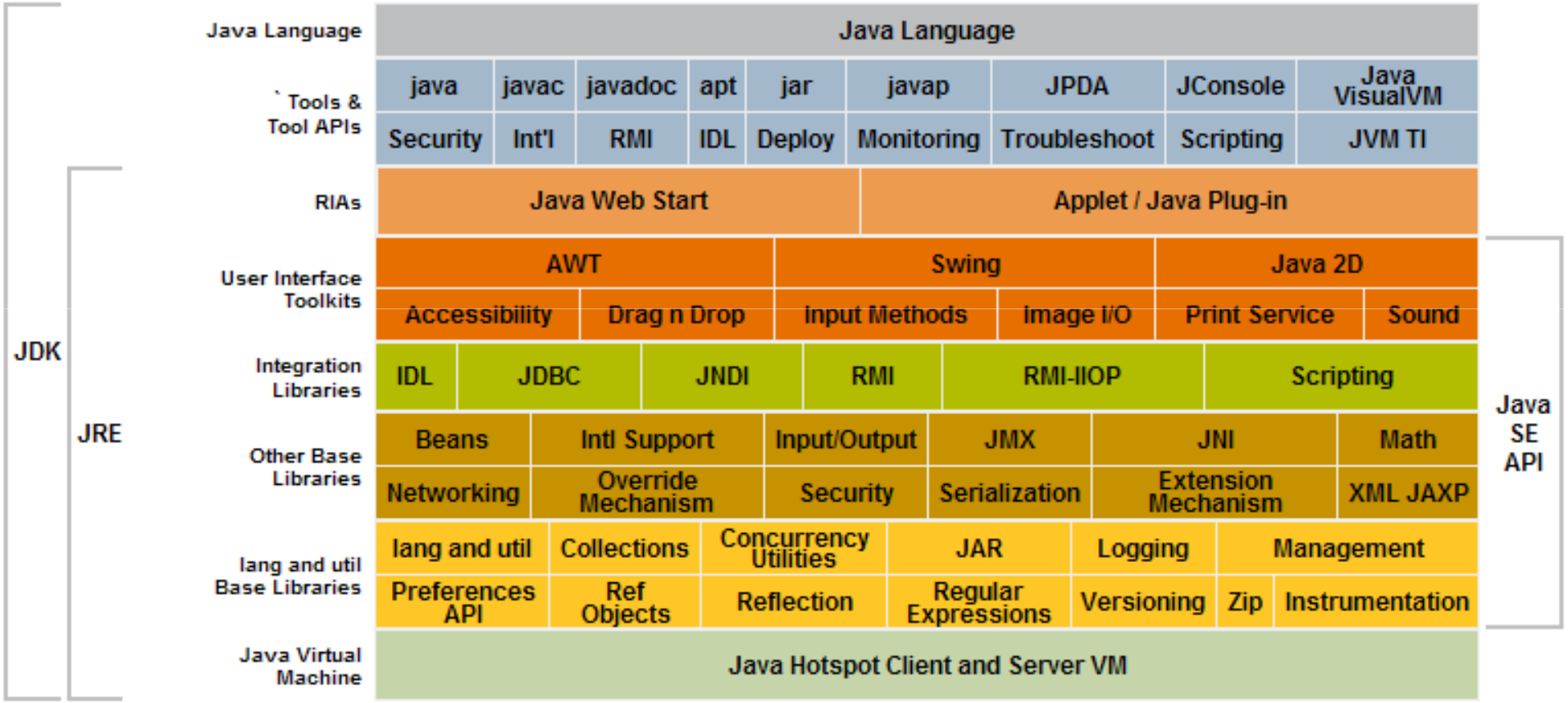
Plataformas Java – Característica común

- Todas las plataformas Java consisten de una máquina virtual Java (VM) y una interfaz de programación de aplicaciones (API).
  - La máquina virtual de Java es un programa, para un hardware en particular y plataforma de software, que ejecuta las aplicaciones de la tecnología Java.
  - Una API es una colección de componentes de software que se puede utilizar para crear otros componentes de software o aplicaciones. Cada plataforma Java proporciona una máquina virtual y una API,

Plataforma Java SE

- Incluye una API que proporciona toda la funcionalidad del lenguaje de programación Java: desde los tipos básicos y los objetos del lenguaje de programación Java, hasta las clases de alto nivel usadas en redes, bases de datos, interfaces de usuario, seguridad, etc.
- Además se compone de una máquina virtual, herramientas de desarrollo, tecnologías de despliegue, y otras librerías de clases y juegos de herramientas (toolkit) de uso común en aplicaciones de la tecnología Java.

Componentes de la Plataforma Java SE



http://download.oracle.com/javase/6/docs/index.html

Plataforma Java EE

- Está construida sobre la plataforma Java SE. Proporciona una API y entorno de ejecución para el desarrollo y ejecución de aplicaciones de tipo servidor para organizaciones de todo tamaño y que pueden ser de varios niveles, escalables, confiables y seguras en red.
- Incluye los siguientes componentes:
  - Java servlets y páginas servidor Java (JSP)
  - Enterprise Java Beans (EJB)
  - E-mail, servicios de mensajes y gestión de transacciones

Plataforma Java ME

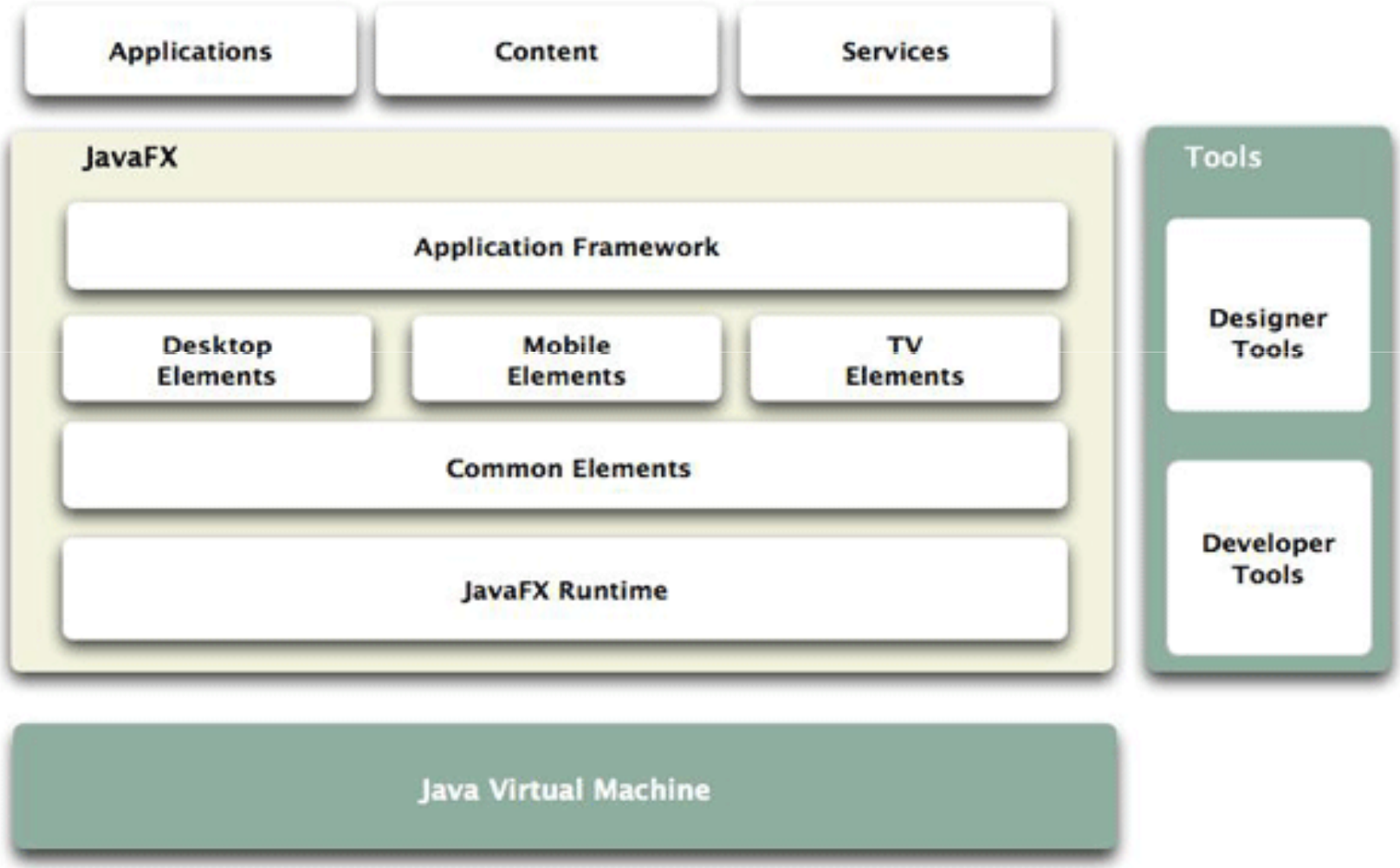
- La plataforma Java ME ofrece un API y una máquina virtual que ocupa poco espacio para ejecutar aplicaciones Java en dispositivos pequeños, como los teléfonos móviles y PDA.
- El API es un subconjunto de la API de Java SE, junto con librerías de clases especiales útiles para el desarrollo de aplicaciones para dispositivos pequeños.
- Las aplicaciones Java ME son a menudo los clientes de los servicios de la plataforma Java EE.

Plataforma Java FX

- La tecnología Java FX es una plataforma para crear aplicaciones ricas de Internet (RIA) escritas en Java FX Script y que se pueden ejecutar en dispositivos conectados.
- Java FX Script es un lenguaje declarativo de tipos estáticos que se compila a bytecode, y que se puede ejecutar en una máquina virtual de Java.
- Las aplicaciones Java FX pueden incluir y vincular a las clases de lenguaje de programación Java, y ser clientes de los servicios de la plataforma Java EE.


Java

Java FX


Applications
Content
Services
JavaFX
Application Framework
Desktop Elements
Mobile Elements
TV Elements
Common Elements
JavaFX Runtime
Tools
Designer Tools
Developer Tools
Java Virtual Machine

Historia de Java

- Creado en 1991 por James Gosling en Sun Microsystems para el desarrollo de aplicaciones domóticas. Inicialmente se llamó Oak
- Se reorientó al desarrollo de aplicaciones en Internet.
- En 1995 se libera el primer Kit de Desarrollo de Java (JDK).
- En 1997 se libera la primera revisión (versión 1.1)
- En 1998 se distribuye la versión 1.2 (Java 2) que introdujo modificaciones bastante significativos.

Evolución de Java

Tabla de versiones Java		
Versión	Año	Características novedosas importantes
JDK 1.0	1996	Liberado el JDK
JDK 1.1	1997	Clases internas
J2SE 1.2	1998	Swing, Framework collections
J2SE 1.3	2000	Mejoras en eficiencia
J2SE 1.4	2002	Aserciones, soporte XML
J2SE 5.0	2004	Clases genéricas, mejoras en ciclo for, autoboxing, enumerados, anotaciones
Java SE 6	2006	Mejoras en las librerías
Java SE 7	2010	Pequeños cambios en el lenguaje y mejoras en librerías

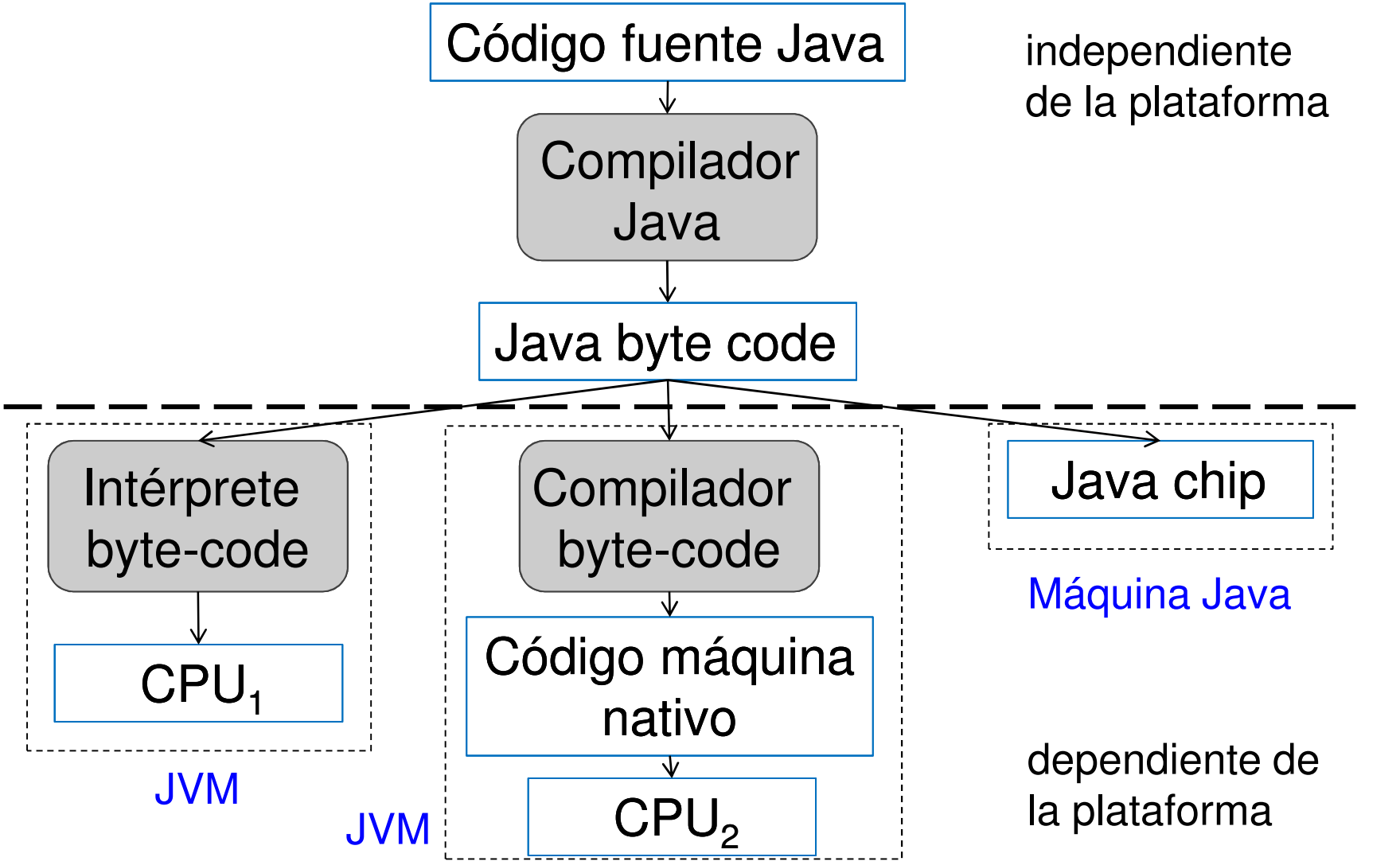
Modelos de ejecución de programas

- Hay dos modelos convencionales para la ejecución de programas:
  - *Compilación*: un compilador (programa) convierte el código fuente de un programa en código máquina (ejecutable) lo que permite ser ejecutado directamente por el sistema operativo y el hardware. La desventaja es que los ejecutables son dependientes de la plataforma.
  - *Interpretación*: un intérprete analiza y ejecuta el código fuente de un programa sin generar código máquina. La gran ventaja es la rapidez de editar y ejecutar y la desventaja es que es dependiente de la plataforma.

Java Virtual Machine

- Los enfoques convencionales no cumplen la característica de independencia de la plataforma (ip).
- El enfoque de Java consiste en un compromiso entre los enfoques de compilación e interpretación.
- Los programas Java se ejecutan en dos fases:
  - Compilación del código fuente a Byte-Code (ip)
  - Ejecución del Byte-Code. Formas:
    - Interpretación. Mediante la JVM.
    - Compilación Just-in-Time (JIT). Produce código nativo.
    - Ejecución directa. Mediante Java chips como PDA, móviles, TV.

Ejecución de Programas Java

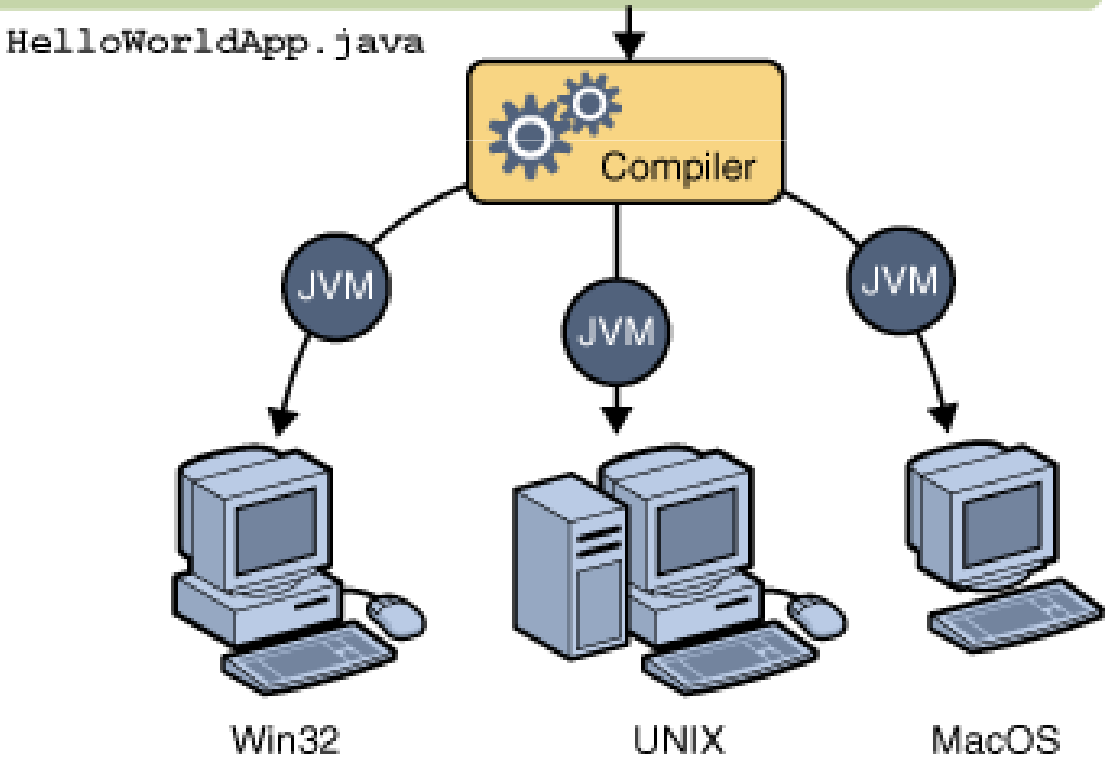


Ejecución de Programas Java

Java Program

```
class HelloWorldApp {
     public static void main(String[] args) {
          System.out.println("Hello World!");
     }
}
```

HelloWorldApp.java



Java Byte-Code

- Son instrucciones de la máquina virtual Java (JVM) que consiste de un código de operación (opcode) de 1 byte y ninguno o más operandos, que son los parámetros de la instrucción.
- Los operandos varían en longitud y el número y longitud de los operandos dependen del opcode.
- La JVM ejecuta el byte-code de la misma forma que una CPU tipo RISC usando registros de 32 bits.
- Las instrucciones JVM utilizan una pila, de la cual obtienen los operandos, los opera y los devuelve.

Tecnología Java

- La tecnología Java es:
  - Un lenguaje de programación
  - Un entorno de desarrollo
  - Un entorno de aplicación
  - Un entorno de despliegue

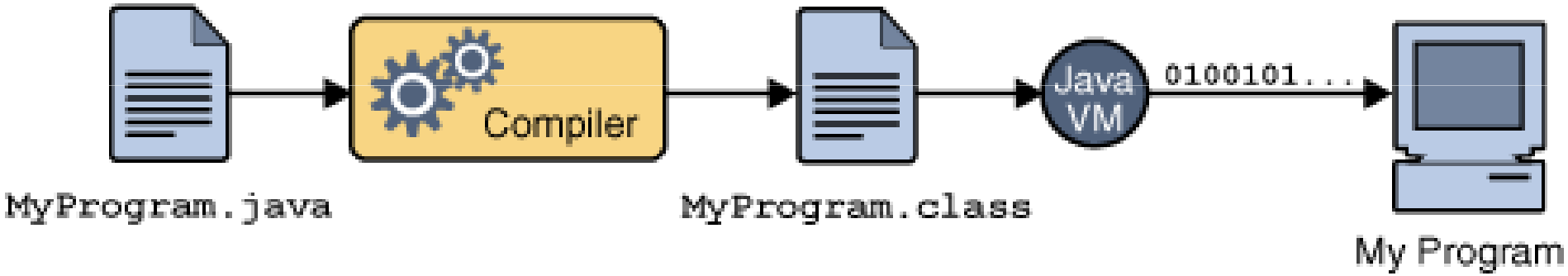
Tecnología Java

- Como lenguaje de programación:
  - Con Java se puede crear todo tipo de aplicaciones que se pueden crear utilizando cualquier lenguaje de programación convencional.
- Como entorno de desarrollo:
  - La tecnología Java proporciona una gran suite de herramientas:
    - Un compilador (javac)
    - Un intérprete (java)
    - Un generador de documentación (javadoc)
    - Una herramienta para el empaquetado de clases y otros toolkits

Tecnología Java

- Como entorno de aplicaciones y ejecución:
  - Las aplicaciones Java generalmente son programas de uso general que se ejecutan en cualquier máquina donde está instalado el entorno de ejecución Java (JRE).
  - Hay dos entornos de despliegue principales:
    - El JRE suministrado por el Java 2 Software Development Kit (SDK) contiene el conjunto completo de ficheros de clases para todos los paquetes de la tecnología Java.
    - Los navegadores web incorporan un intérprete de la tecnología Java y el entorno en tiempo de ejecución.

Proceso de desarrollo con Java



Fases de una aplicación (programa) Java

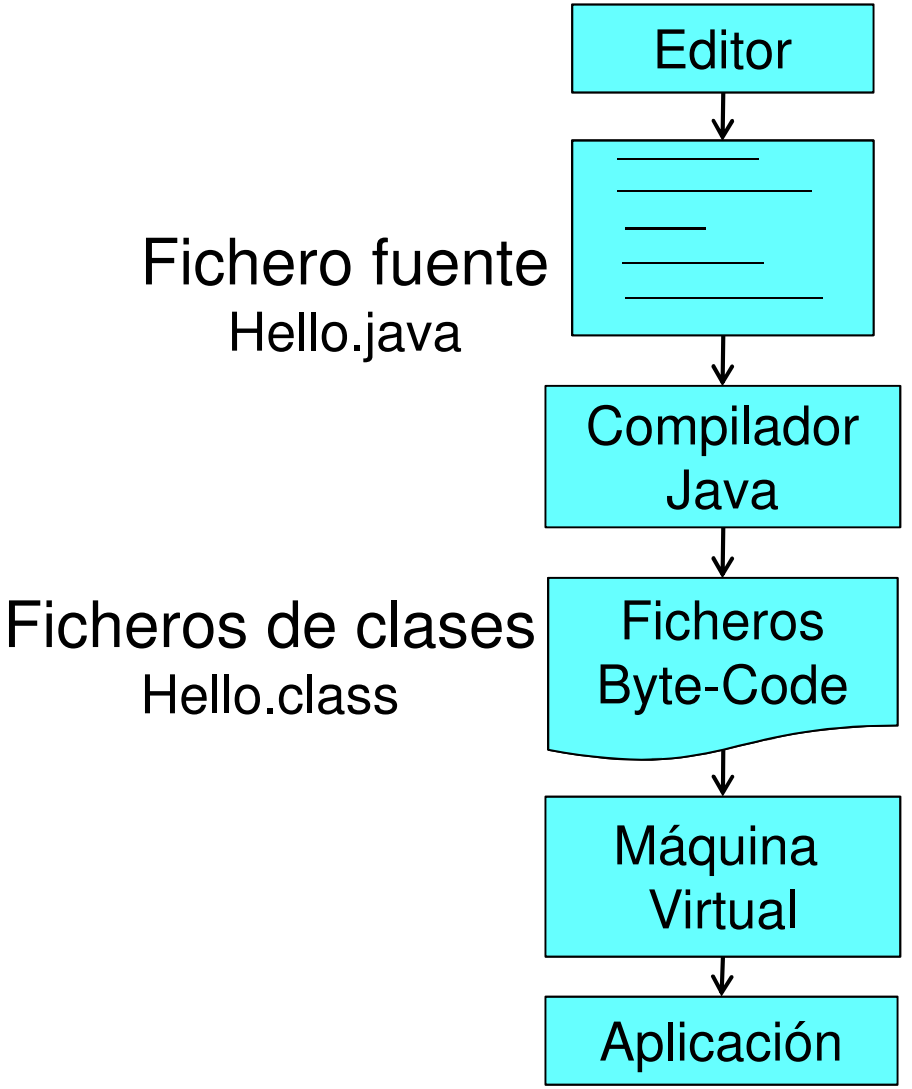
- Edición de código fuente

  Hello.java

- Compilación (javac)

  Hello.class (byte-code)

- Ejecución (JVM) (java)



Índice

- Tecnología Java
- Entornos de desarrollo de Java
  - Java Development Kit (JDK)
  - Entornos IDE (Integrated Development Environment)

Instalación del JDK

- Para crear programas en Java se requiere instalar el Java Development Kit (JDK). Link de descarga:

http://www.oracle.com/technetwork/java/javase/downloads/index.html

- La ubicación común después de instalado es:

  C:\Archivos de programa\Java\jdk1.6.0_20

  es necesario añadir a la variable de entorno PATH el directorio bin

- El JDK incluye programas tales como:
  - java.exe (Ejecuta aplicaciones Java)
  - javac.exe (compilador Java)
  - javadoc.exe (generador de documentación de programas)

Editores y Entornos de Programación

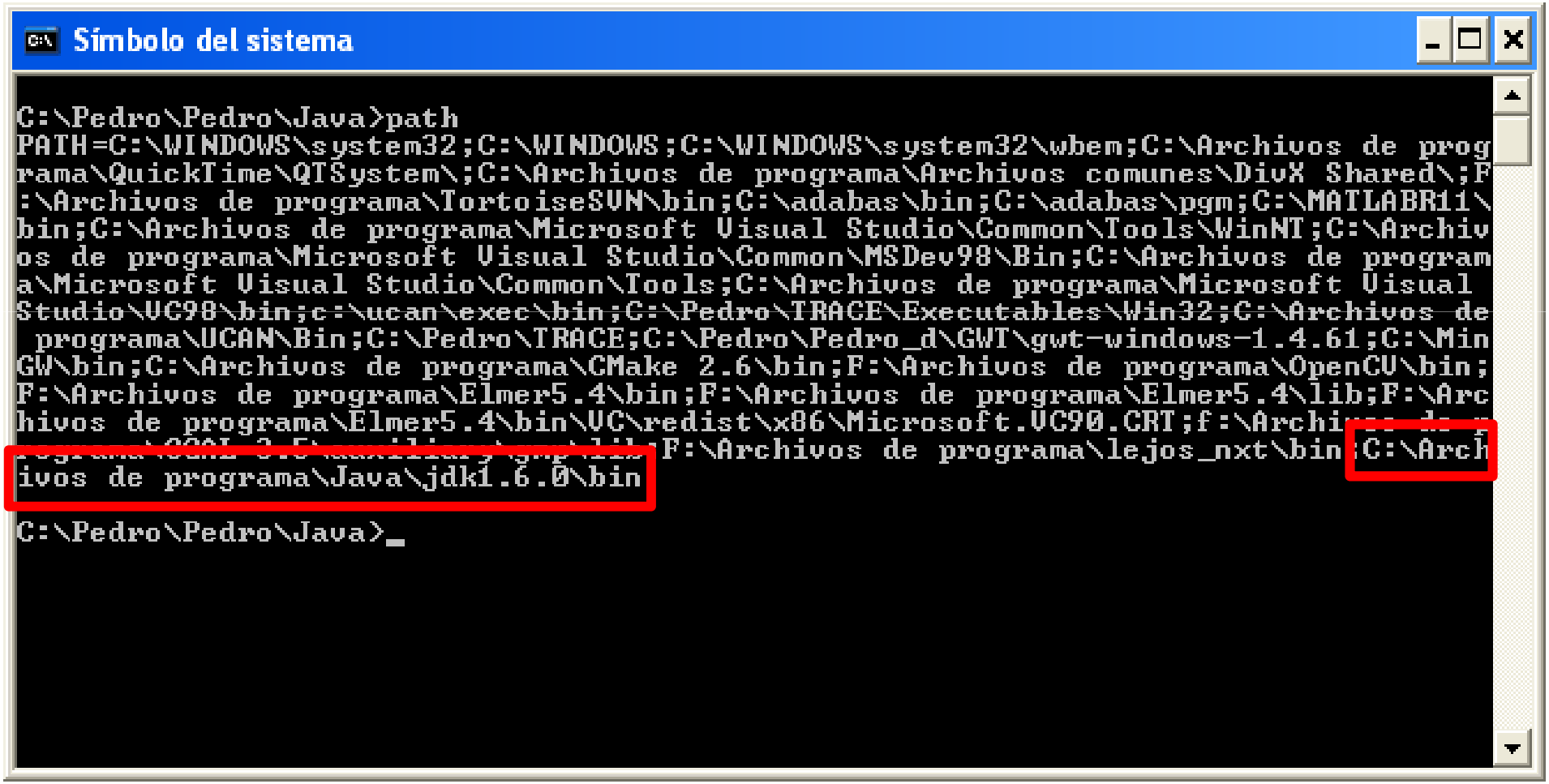
- Hay muchos editores y entornos de desarrollo integrado (IDE) gratuitos y comerciales disponibles para la programación en Java.
  - Ejemplos editores: Notepad, JEdit, emacs, PFE, vi.
  - Ejemplos IDE: **NetBeans**, Eclipse, JCreator, Jbuilder, DrJava.
- Componentes de un IDE (mínimo):
  - Editor sintáctico de código fuente
  - Ventana de resultados (Output)
  - Depurador

Primer programa – Uso consola MSDOS

- La compilación y ejecución de programas Java usando el JDK se puede hacer desde una consola o ventana de comandos MSDOS:
  - Inicio → Todos los programas → Accesorios → Símbolo del sistema
  - Inicio → Ejecutar → en Abrir escribir *cmd*

  Nota: Hay que comprobar que la ruta de comandos incluye los comandos del compilador y el intérprete Java.

Primer programa – Uso consola MSDOS

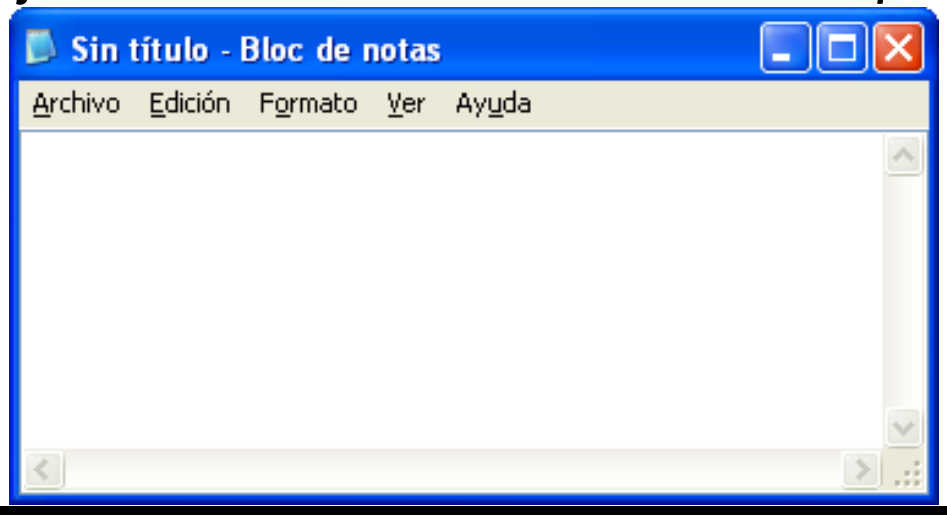


Primer programa – Pasos y comandos

- Los pasos para crear la primera aplicación son:
  - Crear un fichero fuente
    - Un fichero fuente contiene el código o programa en Java. Se puede utilizar cualquier **editor de texto** para crear y editar archivos de código fuente.
  - Compilar el fichero fuente en una fichero .class.
    - El compilador del lenguaje de programación Java (**javac**) convierte el fichero fuente en instrucciones que la máquina virtual Java pueda entender (bytecodes).
  - Ejecutar el programa
    - La herramienta de ejecución de aplicaciones Java (**java**) usa la máquina virtual Java para ejecutar la aplicación.

Primer programa usando editor de textos

- Se puede programar en Java utilizando un editor de textos simple. En Windows existe el editor **Notepad**.
- Se puede obtener de varias formas (XP y Vista):
  - Inicio → Todos los programas → Accesorios → Bloc de Notas
  - Inicio → Ejecutar → en Abrir escribir *notepad*



Primer programa – Código fuente

- Escribir en la ventana del editor de textos:

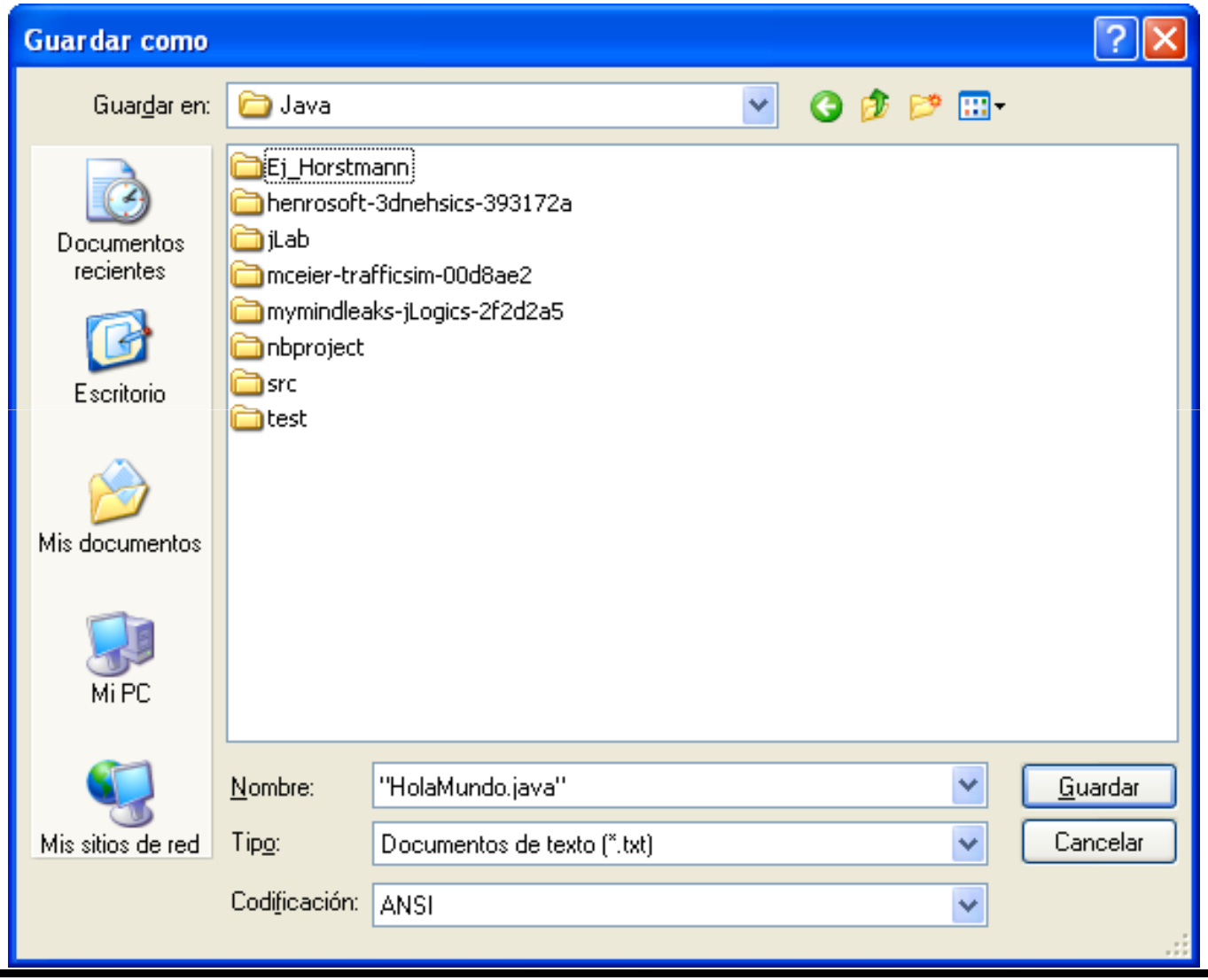
```
/**
 * Ejemplo HolaMundo
 * Imprime el mensaje "Hola, Mundo!"
 */
public class HolaMundo {
  public static void main(String[] args) {
    System.out.println("Hola, Mundo!");
  }
}
```

- Observaciones:
  - Java distingue mayúsculas y minúsculas
  - Java usa caracteres especiales, p.e. { } ( ) ;

Primer programa – Código fuente

- Guardar el código en un fichero de nombre HolaMundo.java. En el bloc de notas:
  - Seleccionar Archivo > Guardar como….
  - En la ventana de diálogo que aparece, en el campo desplegable Guardar en, especificar la carpeta (directorio) donde se desea guardar el archivo.
  - En el campo de texto Nombre de archivo, escribir "HolaMundo.java", incluyendo las comillas.
  - En el campo Tipo combo, seleccionar Documentos de texto (*. txt).
  - En el campo Codificación dejar la codificación como ANSI.

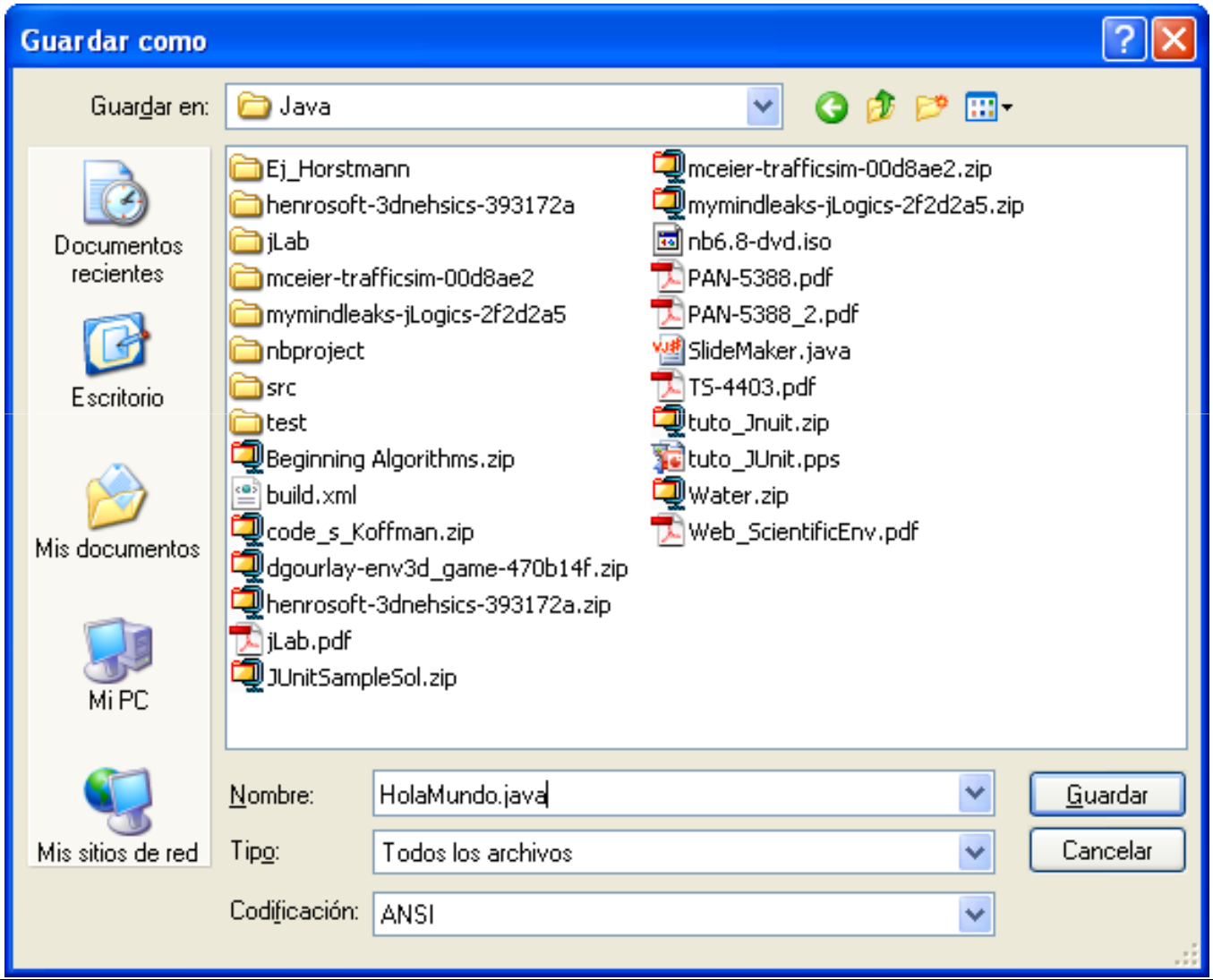
Primer programa – Código fuente



Primer programa – Código fuente

- Otra posibilidad es:
  - En el bloc de notas seleccionar Archivo > Guardar como….
  - En la ventana de diálogo que aparece, en el campo desplegable Guardar en, especificar la carpeta (directorio) donde se desea guardar el archivo.
  - En el campo de texto Nombre de archivo, escribir `HolaMundo.java`.
  - En el campo Tipo combo, seleccionar Todos los archivos.
  - En el campo Codificación dejar la codificación como ANSI.
  - Hacer clic en Guardar y salir del editor.

Primer programa – Código fuente



Primer programa – Compilación

- Obtener una consola MSDOS. El prompt o indicador (>) muestra el directorio actual. Por lo general el directorio actual la primera vez es el directorio de usuario (Windows XP).
- Para compilar el fichero fuente, cambiar al directorio donde se encuentra el archivo (comando cd). Se puede usar el comando dir para ver el fichero fuente.
- Para compilar, escribir el siguiente comando y pulsar Enter.

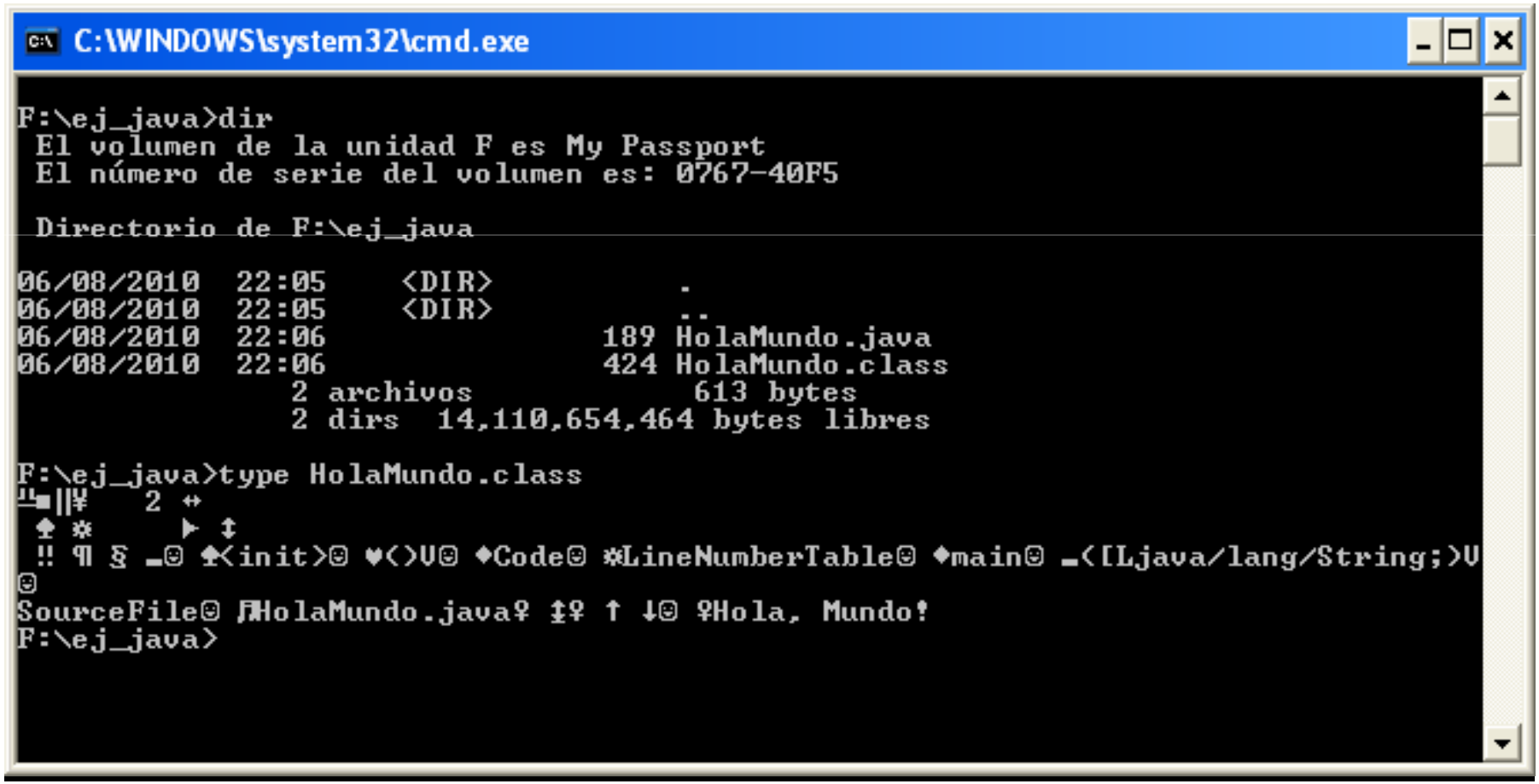
```
> javac HolaMundo.java
```

Primer programa – Compilación

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Pedro>f:

F:\>cd ej_java

F:\ej_java>dir
 El volumen de la unidad F es My Passport
 El número de serie del volumen es: 0767-40F5

 Directorio de F:\ej_java

06/08/2010  22:05    <DIR>          .
06/08/2010  22:05    <DIR>          ..
06/08/2010  22:06               189 HolaMundo.java
               1 archivos            189 bytes
               2 dirs  14,110,818,304 bytes libres

F:\ej_java>javac HolaMundo.java

F:\ej_java>_
```

Primer programa – Compilación

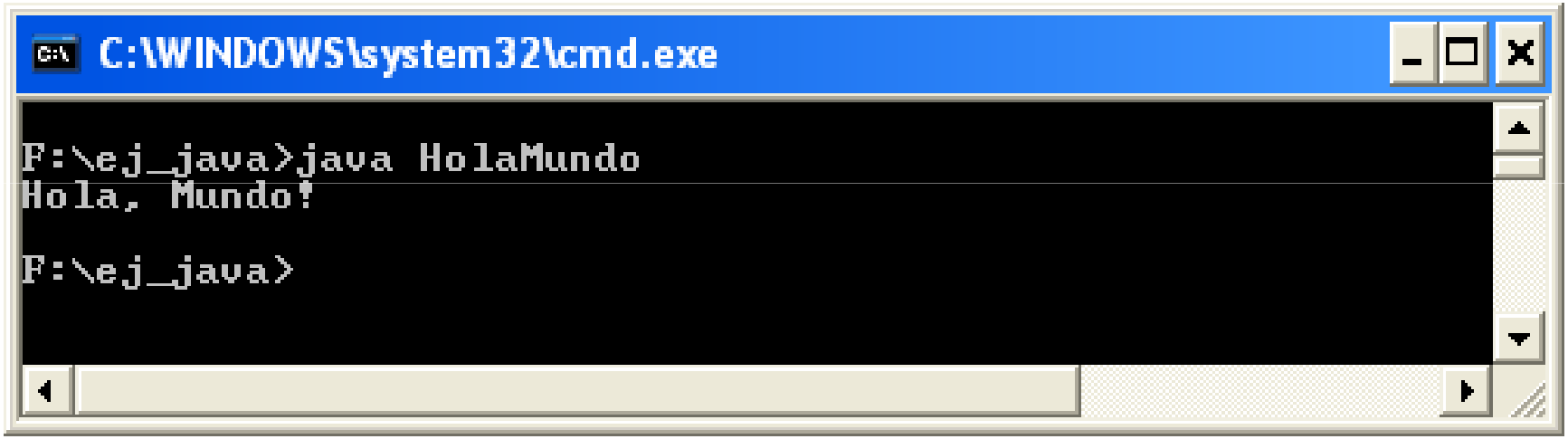
- El compilador genera un fichero binario de bytecodes, HolaMundo.class.



Primer programa – Ejecución

- En el mismo directorio escribir el siguiente comando:

  > java HolaMundo



- Si se obtienen errores en la fase de compilación o ejecución se puede deber a errores en la escritura del programa.

Resumen ejecución MSDOS

- En un *editor de texto* escribir el código fuente del programa Java y guardarlo en un fichero con el **mismo nombre** que la clase que contiene el método *main* y extensión **.java**.
- En una Consola MSDOS escribir el comando:

  ```
  > javac NombreClase.java
  ```

- Para ejecutar el programa escribir el comando:

  ```
  > java NombreClase
  ```

Errores

- Errores de compilación
  - Debido a errores sintácticos: errores de escritura en el código fuente. Los más comunes son:
    - Errores de ortografía, capitalización, orden de las declaraciones, mal emparejamiento de llaves / paréntesis, olvidarse de escribir un punto y coma al final de una instrucción.
  - El compilador no genera ninguna clase.
  - Debe corregirse el primero (lista) y compilar de nuevo.
- Errores de ejecución
  - Debido a errores lógicos.
  - El programa corre pero produce resultados inesperados.
  - El programa puede abortar.

Primer programa – IDE NetBeans

- NetBeans es un entorno de programación integrado de código abierto y gratuito desarrollado en Java y que soporta la tecnología Java otros lenguajes.
- Se puede descargar en:

  http://netbeans.org/downloads/index.html
- El IDE NetBeans facilita la programación en Java y soporta el desarrollo de todos los tipos de aplicación Java (J2SE, J2EE, Web, EJB y aplicaciones móviles).

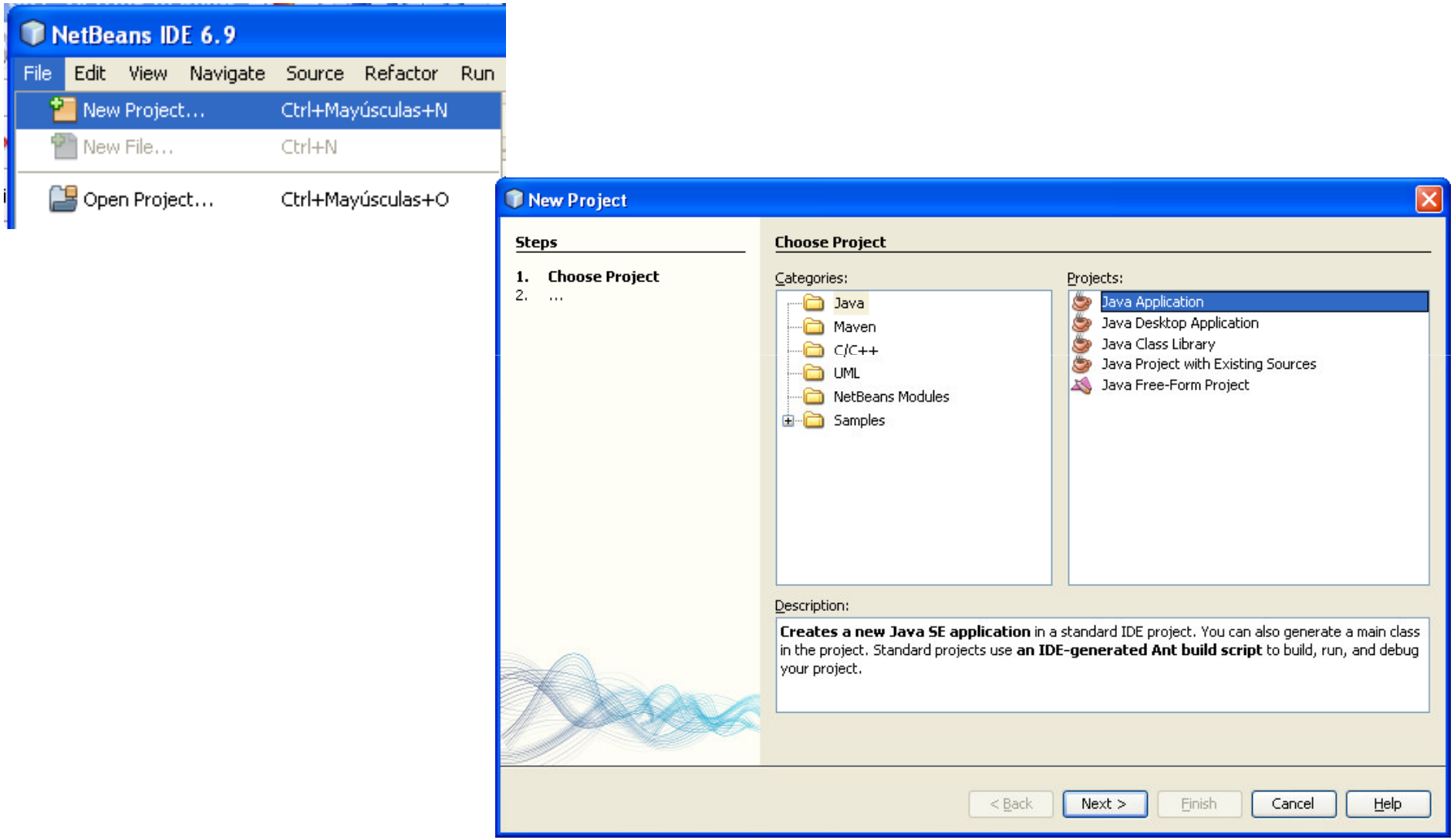
Primer programa – Pasos con NetBeans

- Los pasos para crear la primera aplicación son:
  - Crear un proyecto IDE
    - Cuando se crea un proyecto de IDE, se crea un entorno para generar y ejecutar las aplicaciones, eliminando los problemas de configuración asociados al desarrollo en modo Consola.
  - Añadir código al fichero fuente generado
    - Cuando se crea un proyecto se genera automáticamante una plantilla que se puede modificar con el código fuente.
  - Compilar el código en un fichero .class
  - Ejecutar el programa
    - En ambas acciones NB invoca los comandos javac y java .

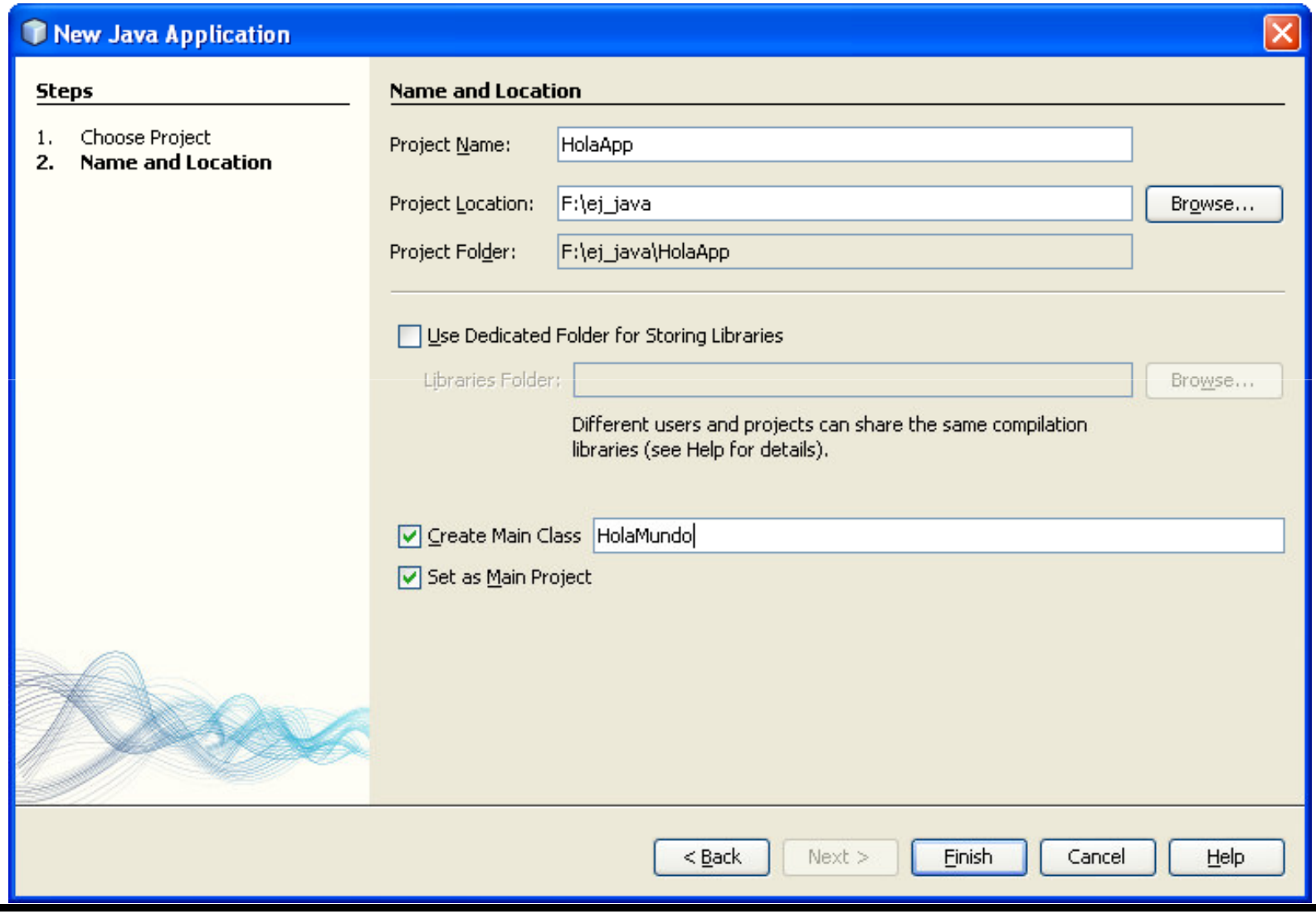
Primer programa – Proyecto en NetBeans

- Ejecutar NetBeans: Inicio → Todos los programas → NetBeans → NetBeans IDE 6.9
- En NB seleccionar File > New Project
- En la ventana New Project seleccionar en Categories: Java y en Projects: Java Application y pulsar Next
- En Name and Location
  - En el campo Project Name, escribir HolaApp
  - En el campo Create Main Class, escribir HolaMundo
  - Dejar Set as Main Project checkbox seleccionado.
  - Seleccionar el directorio para el proyecto. Pulsar Finish.

Primer programa – Proyecto en NetBeans



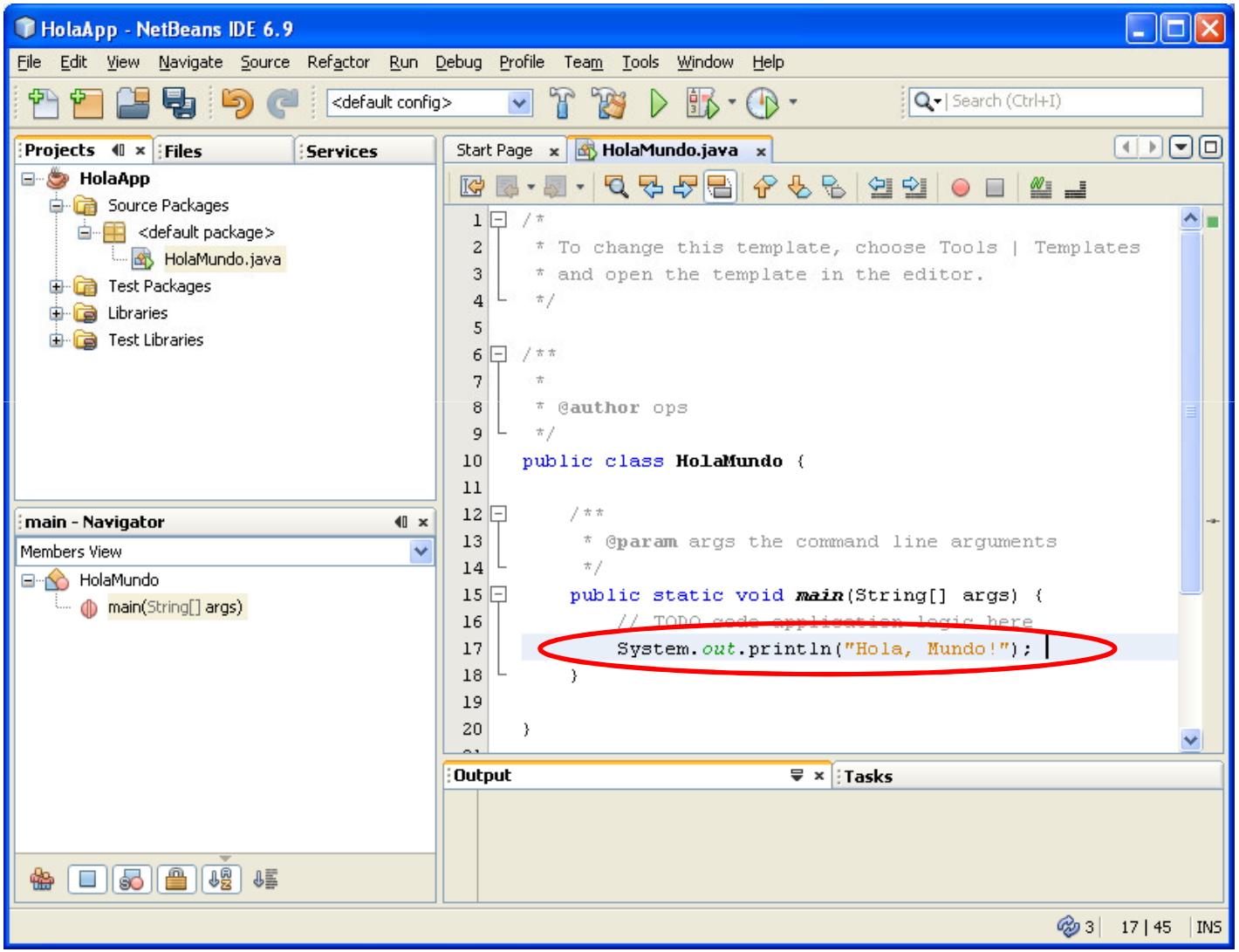
Primer programa – Proyecto en NetBeans



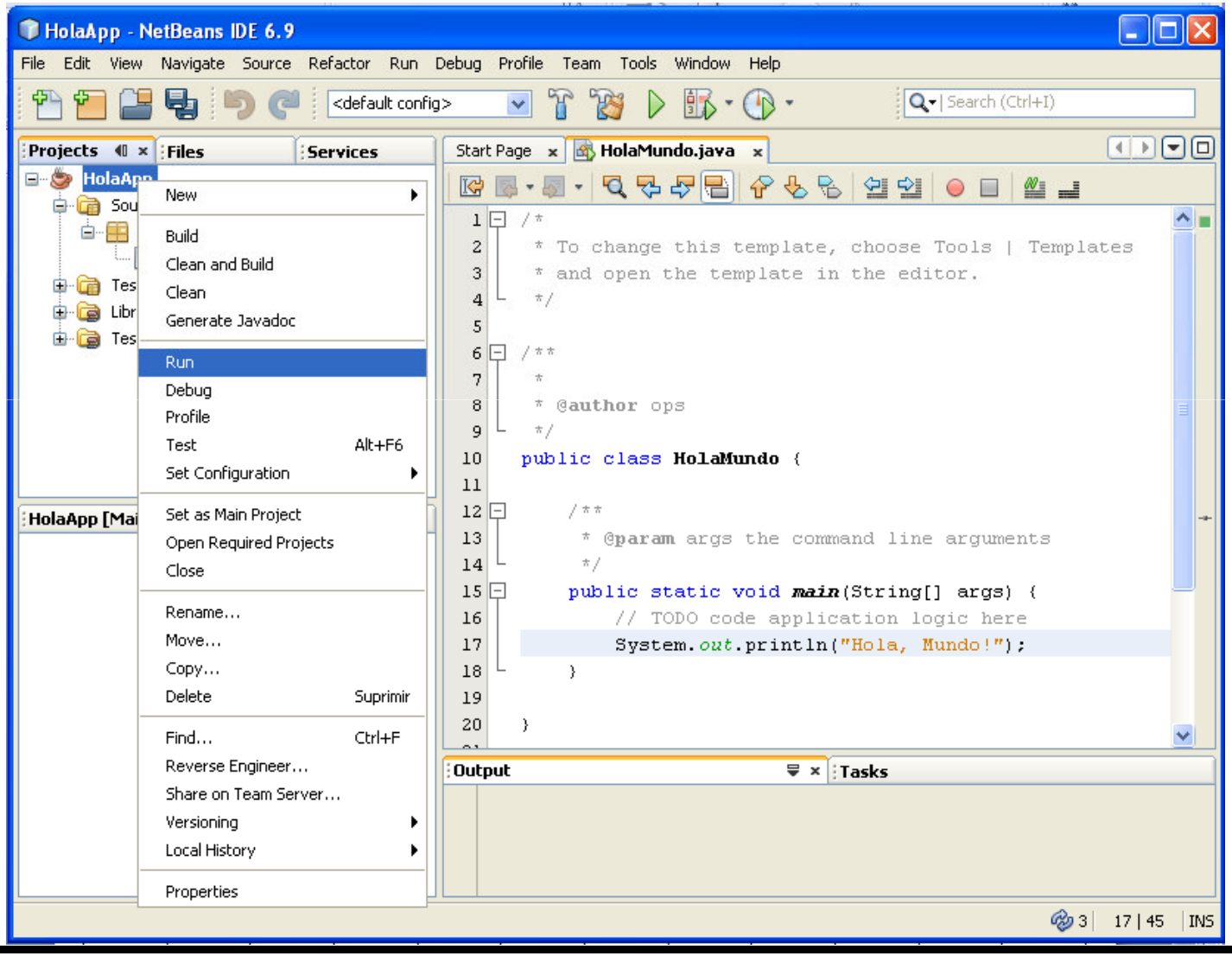
Primer programa – Proyecto en NetBeans

- Aparece el nodo HolaApp en la ventana Projects.
- Modificar HolaMundo.java según el código del programa.
- Para compilar y ejecutar, seleccionar con el botón derecho HolaApp y seleccionar Run.
- Para compilar solamente, seleccionar Run > Build Main Project o el icono 
- Para ejecutar seleccionar Run > Run Main Project o 
- El resultado aparece en la ventana Output.

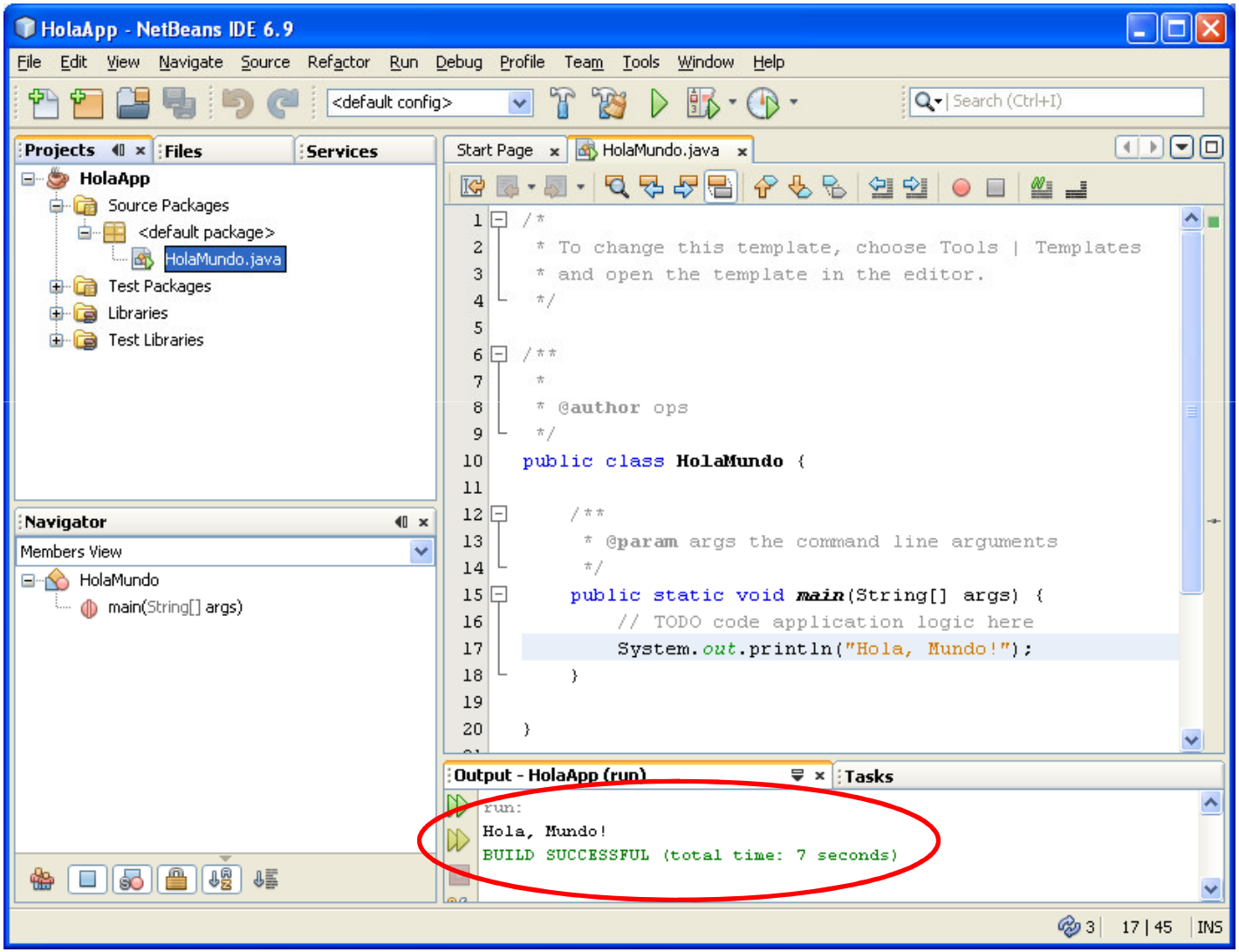
Primer programa – Proyecto en NetBeans



Primer programa – Proyecto en NetBeans

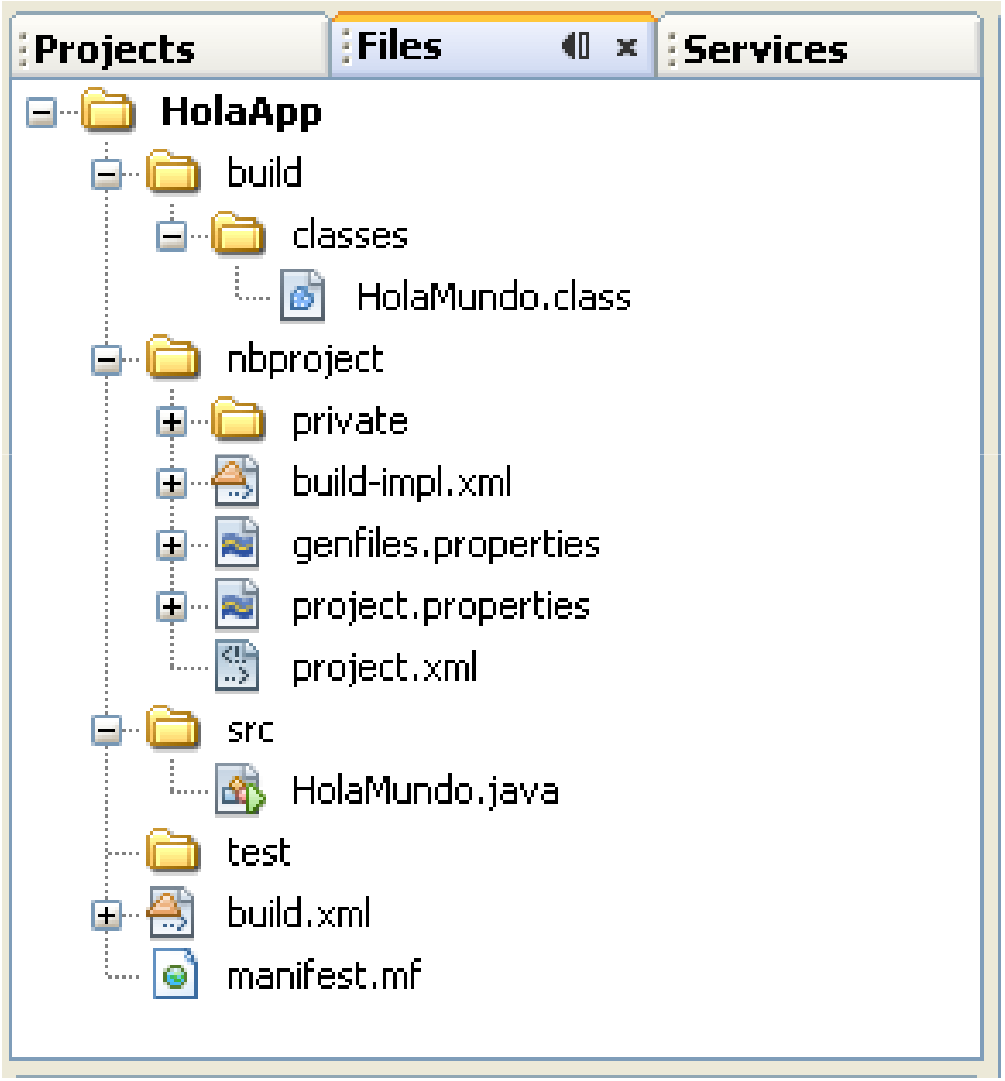


Primer programa – Proyecto en NetBeans



Primer programa – Proyecto en NetBeans

- Ficheros generados por NetBeans.
- El directorio de trabajo contiene los mismos ficheros.



Primer programa – Análisis

```
/*
 * Ejemplo HolaMundo
 * Imprime el mensaje "Hola, Mundo!"
 */
public class HolaMundo {
  public static void main(String[] args){
    System.out.println("Hola, Mundo!");
  }
}
```

- Un programa Java tiene una o más clases.
- Un programa Java tiene un método main de inicio.
- Las instrucciones deben terminar con ; punto y coma.

Primer programa – Análisis

```
/*
 * Ejemplo HolaMundo
 * Imprime el mensaje "Hola, Mundo!"
 */
```

- Comentarios tipo C. Toda cadena encerrada entre

```
/*. . . */
```

```
System.out.println("Hola, Mundo!");
```

- Método para imprimir: println.
- Los paréntesis encierran el argumento que se pasa al método.
- Imprime valores o cadenas de caracteres.