

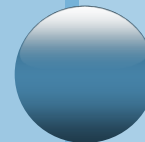


# Curso de Java

## La tecnología Java



**Prof. Ing. Guido Acosta**





# Objetivos del curso

---

- Conocer los conceptos más importantes de la plataforma y el lenguaje de programación Java.
- Entender y aplicar conceptos de Programación Orientada a Objetos.



# Contenido de la clase

---




- **Elementos de la tecnología Java**
- **Compilación y ejecución de un programa Java**
- **Estructura básica de un programa Java**



# ¿Por qué Java?



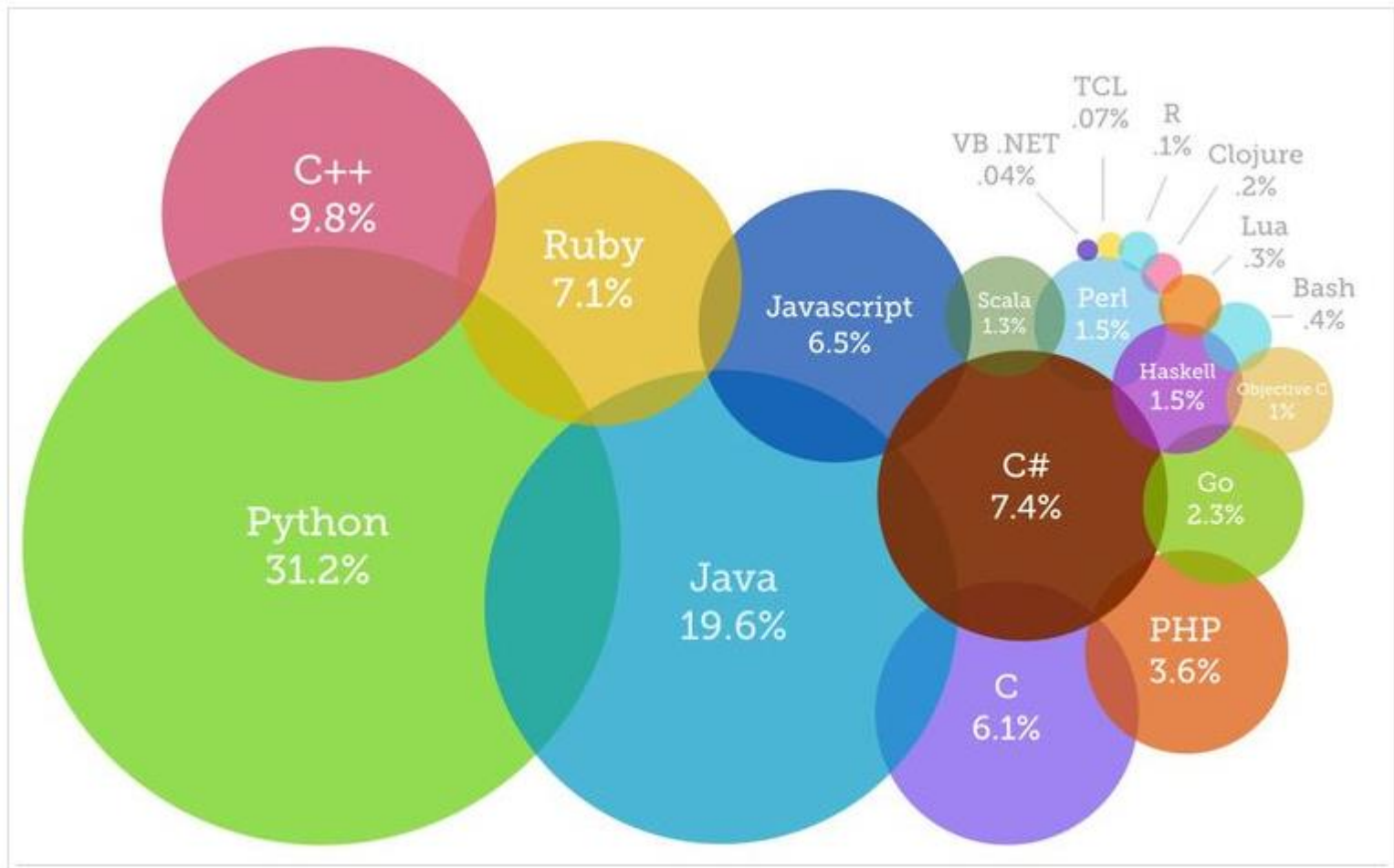
## The Most In-Demand Programming Languages of 2019

1	Java		11	MATLAB	
2	C		12	R	
3	Python		13	Perl	
4	C++		14	Assembly Language	
5	Visual Basic .NET		15	Swift	
6	Javascript		16	Go	
7	C#		17	Delphi/Object Pascal	
8	PHP		18	Ruby	
9	SQL		19	PL/SQL	
10	Objective-C		20	Visual Basic	



# ¿Por qué Java?

## 10 Best Programming Languages of 2019 You Should Know





# Java en Acción



**Java +  
Google Maps**



**Java + Sony  
Pictures**



**Java EE**



**Java + Android**



# La tecnología Java

---

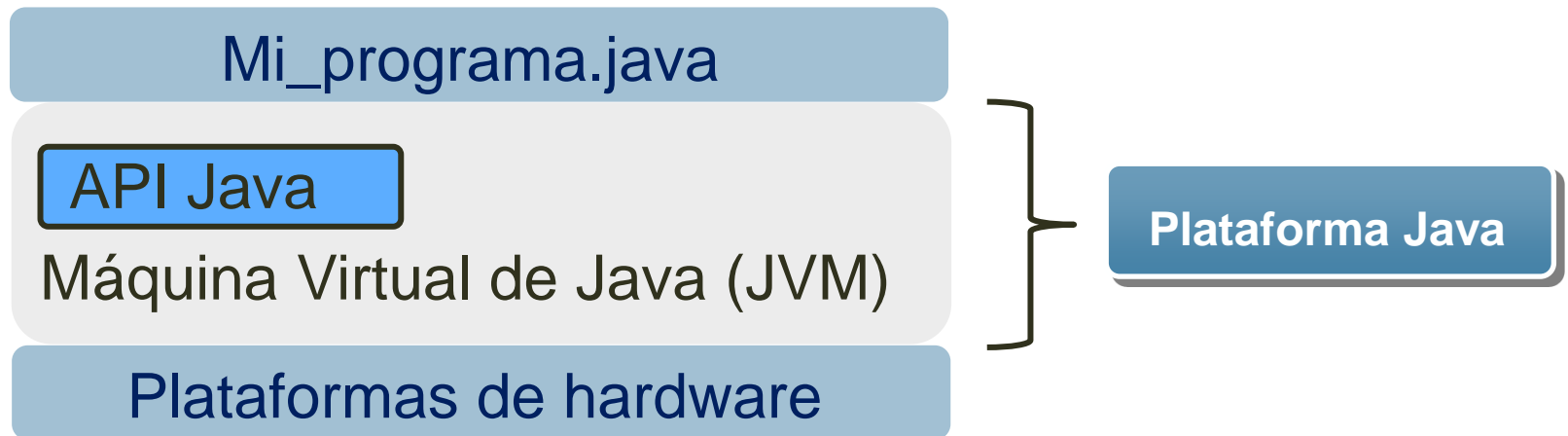
- El lenguaje de programación
- La plataforma



# La plataforma

La plataforma Java tiene dos componentes:

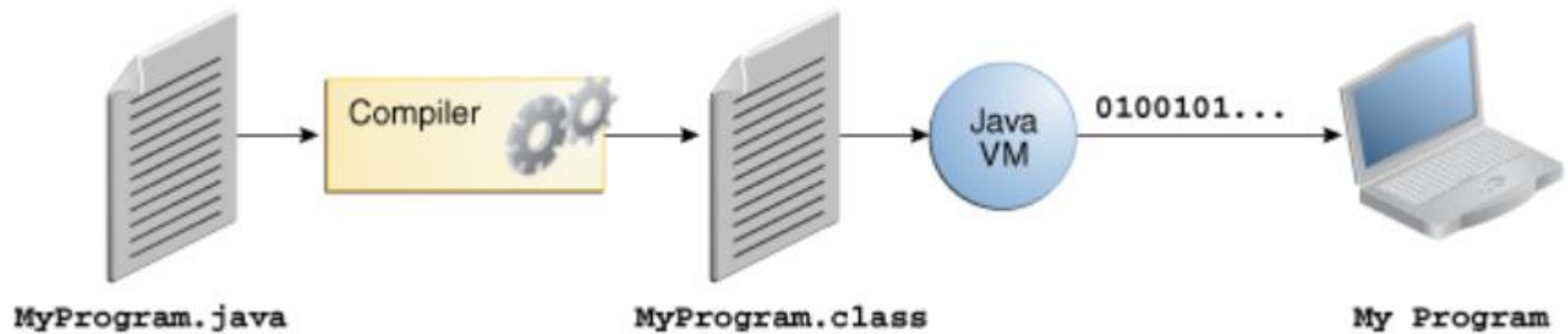
- The Java Virtual Machine (JVM)
- The Java Application Programming Interface (API)







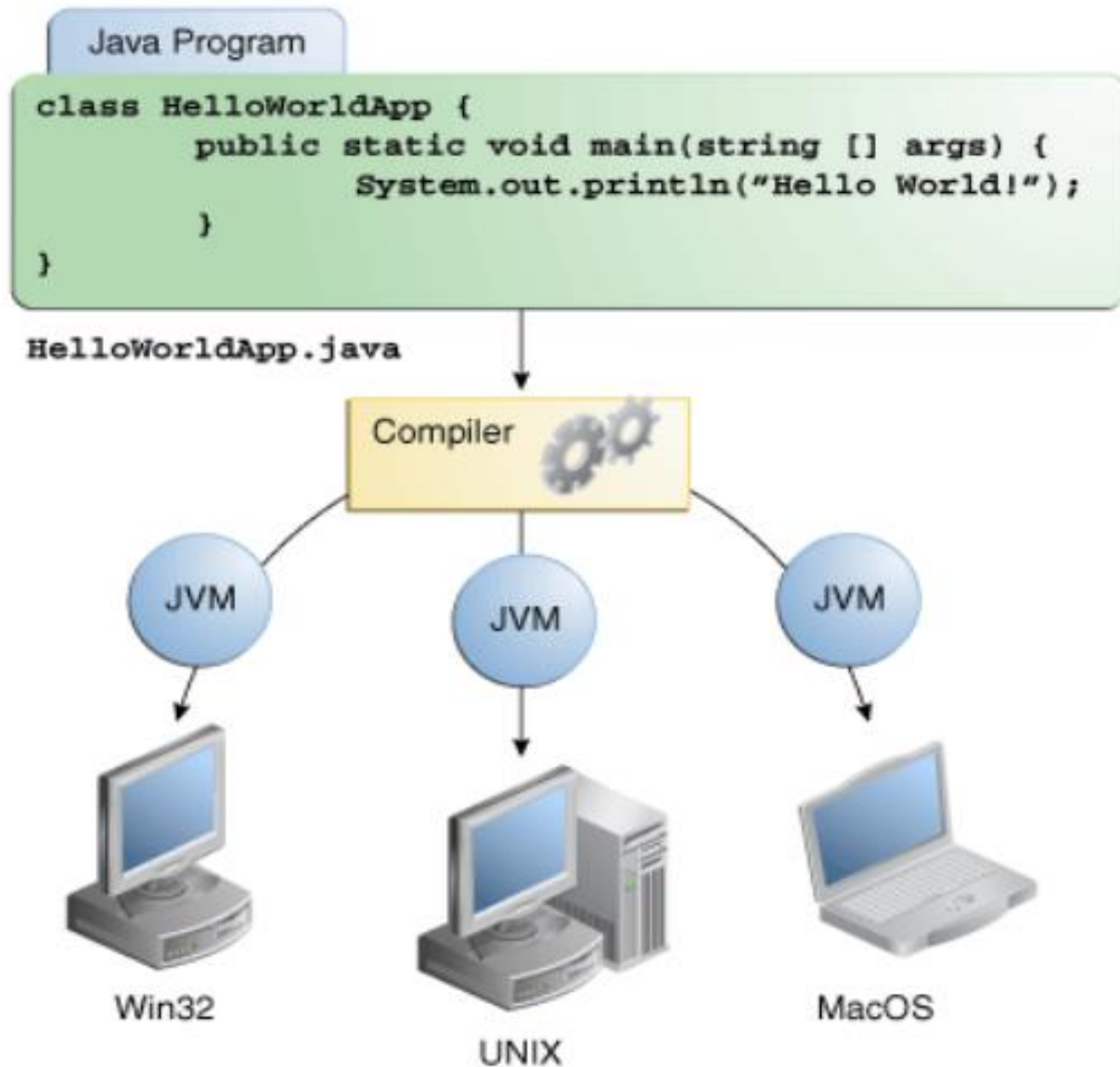
# Proceso de Desarrollo de Software



↓

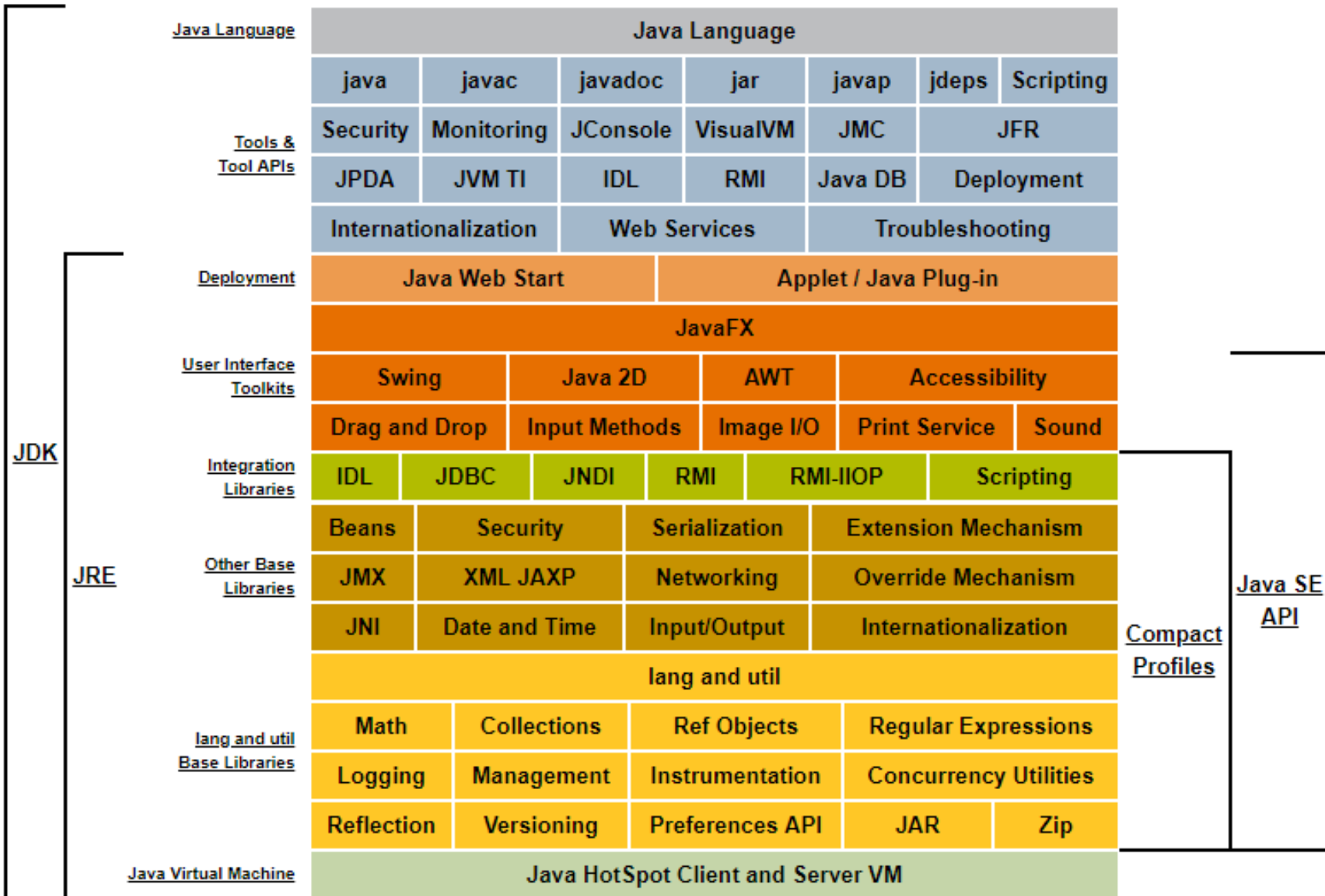
```
private static int[] sort(int[] numbers) {  
    for (int i = 0; i < numbers.length - 1; i++) {  
        for (int j = i + 1; j < numbers.length; j++) {  
            if (numbers[i] > numbers[j]) {  
                int auxiliar = numbers[i];  
                numbers[i] = numbers[j];  
                numbers[j] = auxiliar;  
            }  
        }  
    }  
    return numbers;  
}
```

# Programa portable





# La tecnología Java



<https://docs.oracle.com/javase/8/docs/index.html>



# JRE (Java Runtime Environment)

---

**El JRE es una máquina virtual de Java y su función es hacer de intermediario entre una aplicación programada en Java y el sistema operativo que se este usando. De este modo, cualquier aplicación puede funcionar en cualquier sistema operativo que disponga del JRE.**



# JDK (Java Development Kit)

---

Se trata de un conjunto de herramientas que permiten desarrollar programas en lenguaje Java.

- **Compilar**
- **Ejecutar**
- **Generar documentación, etc.**



# JDK - Componentes

---

**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.

**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.



# Java Virtual Machine

**Entorno de  
Ejecución**

**Archivos .class**

**Cargador de clases**

**Verificador bytecode**

**Intérprete**

**Máquina Virtual  
Java (JVM)**



# JDK - Ubicación

Windows Explorer window showing the location of the JDK bin directory: C:\Archivos de programas\Java\jdk1.6.0\_02\bin

Favorite Links: Documents, Pictures, Music, More >>

Folders: InstallShield Insta, Intel, Internet Explorer, Internet Movil Ti, Internet Personal, InterVideo, Intuit

Name	Date modified	Type	Size	Tags
appletviewer.exe	4/12/2007 7:17 AM	Application	25 KB	
apt.exe	4/12/2007 7:17 AM	Application	25 KB	
beanreg.dll	4/12/2007 9:15 AM	Application Extens...	29 KB	
extcheck.exe	4/12/2007 7:17 AM	Application	25 KB	
HtmlConverter.exe	4/12/2007 7:36 AM	Application	44 KB	
idlj.exe	4/12/2007 7:17 AM	Application	25 KB	
jar.exe	4/12/2007 7:17 AM	Application	25 KB	
jarsigner.exe	4/12/2007 7:17 AM	Application	25 KB	
java.exe	4/12/2007 7:17 AM	Application	132 KB	
javac.exe	4/12/2007 7:17 AM	Application	25 KB	
javadoc.exe	4/12/2007 7:17 AM	Application	25 KB	
javah.exe	4/12/2007 7:17 AM	Application	25 KB	
javap.exe	4/12/2007 7:17 AM	Application	25 KB	





# El lenguaje de programación

---

**Java es un lenguaje de programación de propósito general Orientado a Objetos.**



# Java es multiplataforma

---

**Los programas escritos en java se compilan en un bytecode independiente de la máquina y todos los SO principales tienen entornos de ejecución de aplicaciones Java (máquina virtual).**



# Java es seguro

---

**Pueden forzarse restricciones sobre operaciones permitidas, algunos aplicaciones no acceden directamente el hardware de la máquina.**

**Al cargar un programa en memoria, la máquina virtual Java verifica los bytecodes de la aplicación.**



# Amplio conjunto de bibliotecas estándar

---

**Bibliotecas para trabajar con colecciones y otras estructuras de datos, archivos, acceso a BD, interfaces gráficas de usuarios, interfaces web, thread, compresión de datos, criptografía, etc.**



# Contenido de la clase

---

- Elementos de la tecnología Java
- Compilación y ejecución de un programa Java**
- Estructura básica de un programa Java



# Compilación

---

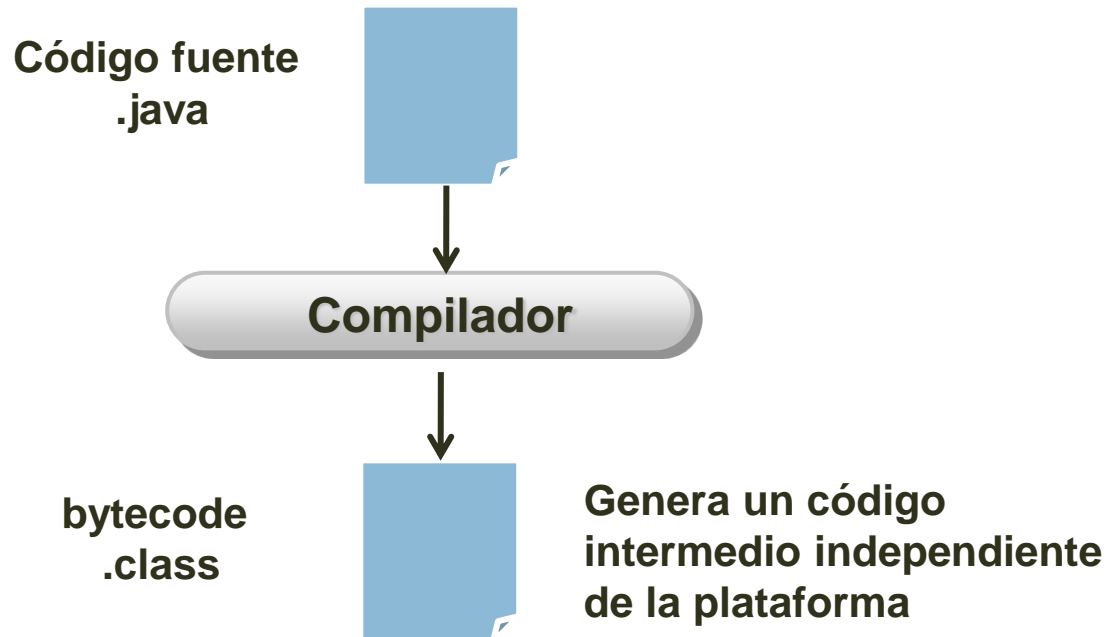
Código fuente  
.java



Compilador



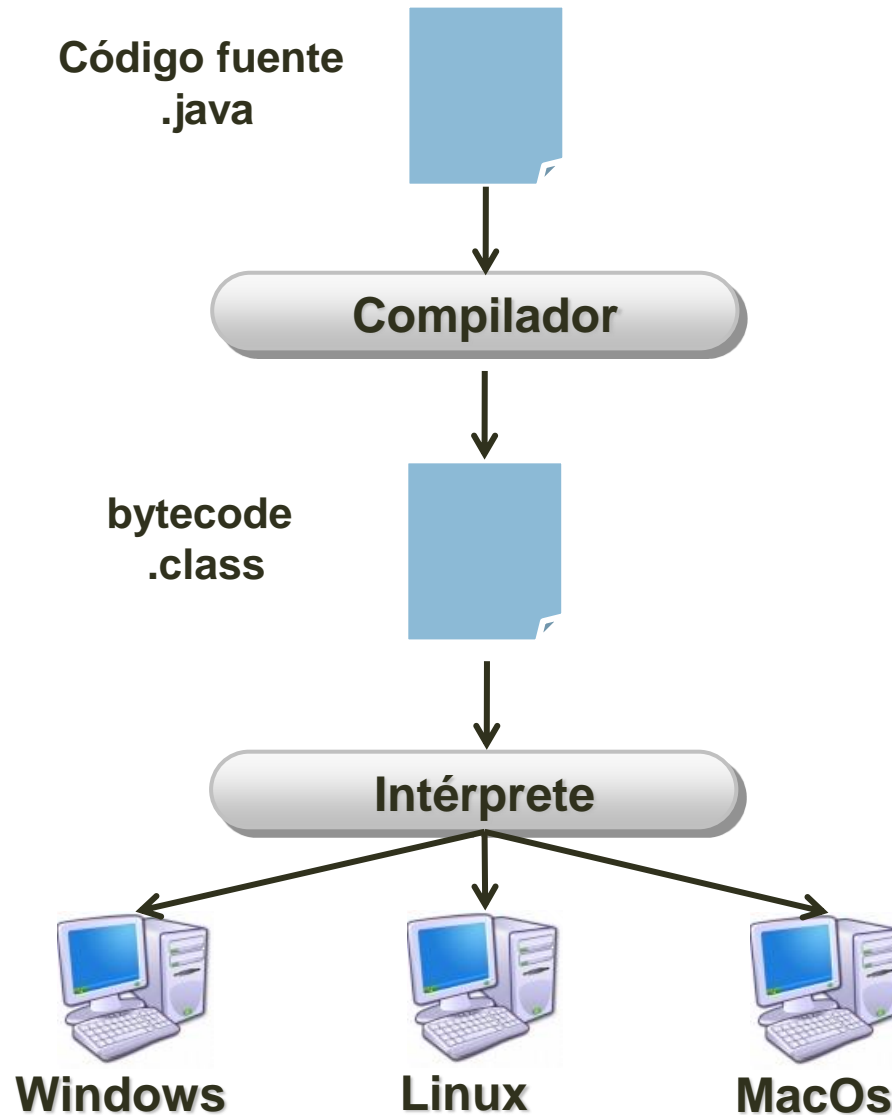
# Compilación



Pueden considerarse como el lenguaje máquina de la JVM. Pero el archivo .class que produce el compilador no está listo todavía para ejecutarse ya que requiere vincularse con las bibliotecas de clases Java proceso que se realizará cuando el programa se ejecute.



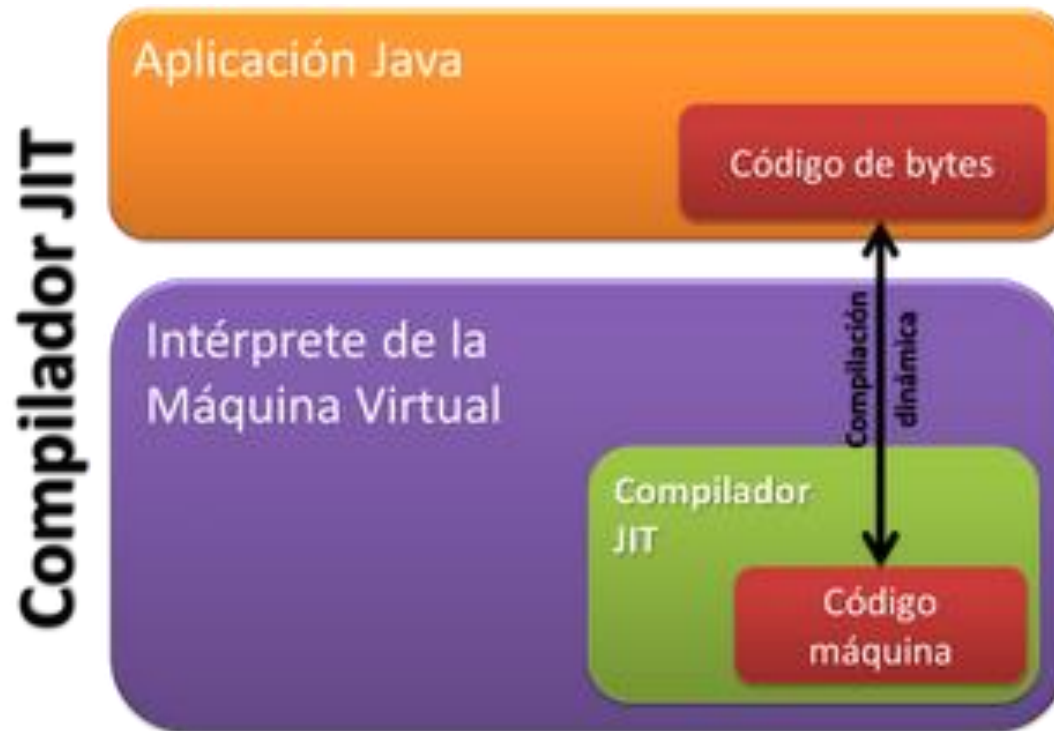
# Ejecución







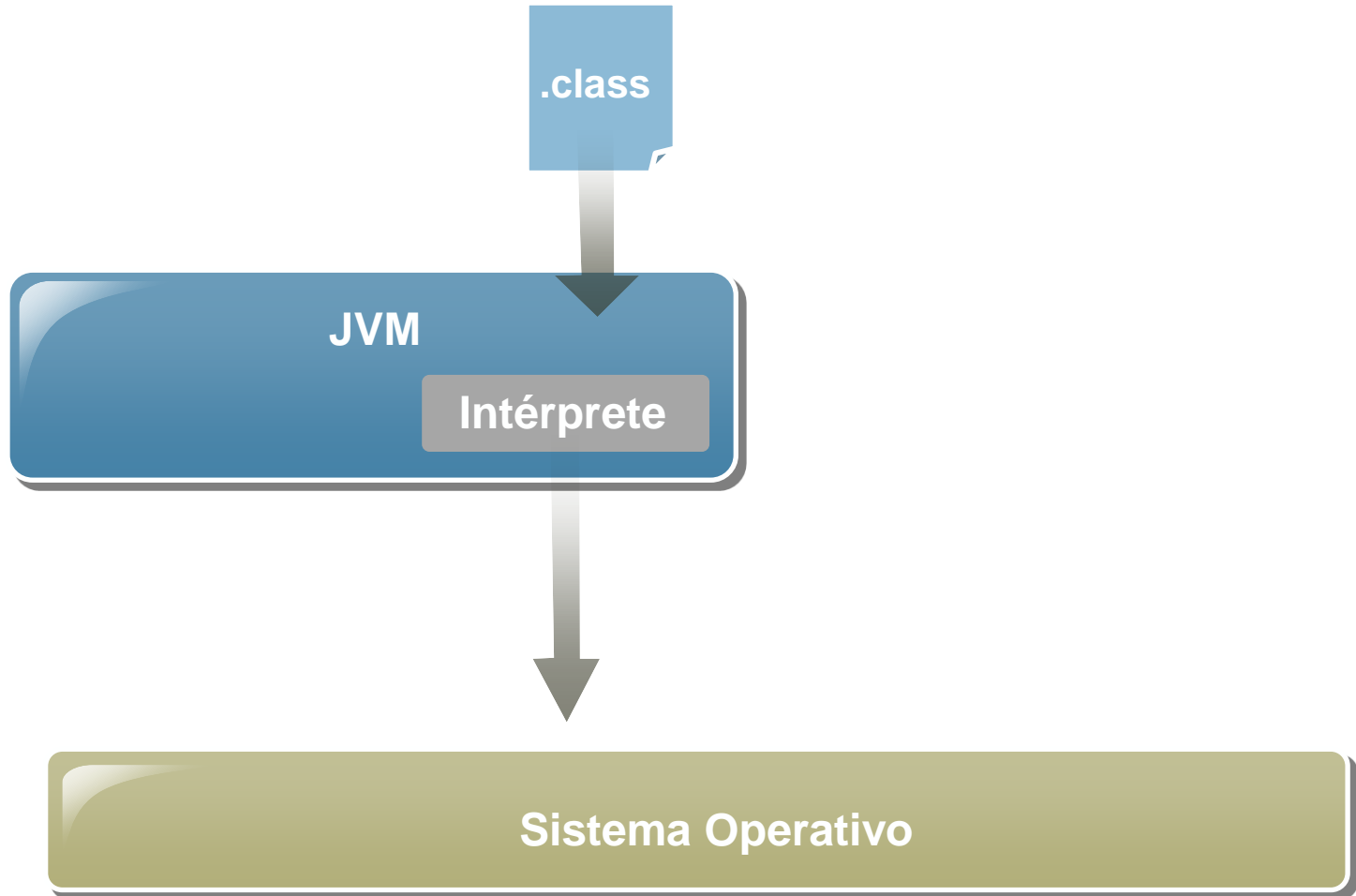
# JIT (Java Just in Time)





# JIT (Java Just in Time)

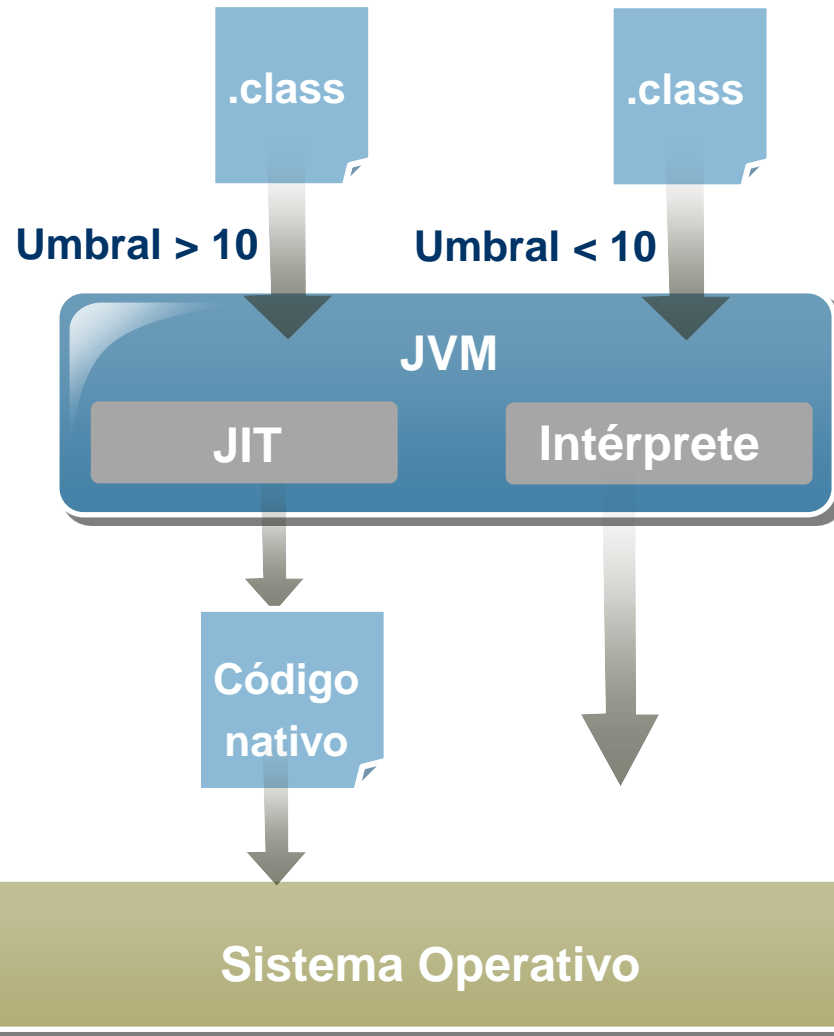
**JIT OFF**





# JIT (Java Just in Time)

**JIT = ON    Umbral = 10**





# Contenido de la clase

---

- Elementos de la tecnología Java
- Compilación y ejecución de un programa Java
- **Estructura básica de un programa Java**



# Ciclo editar, compilar, ejecutar

---

**El aprendizaje de todo lenguaje de programación pasa por la etapa obligatoria de realizar un primer programa que muestre:**

- La estructura sintáctica mínima del lenguaje.**
- La manera de editar, compilar y ejecutar el programa.**
- La configuración correcta del ambiente de desarrollo**



# Práctica 1 – Edición

- Se crea el programa con la ayuda de un editor

```
// HolaMundo.java
public class Hola {
    public static void main(String[ ] args) {
        System.out.println("Hola, mundo!");
    }
}
```

Edición (Case sensitive)

- Se guarda el archivo con el nombre `HolaMundo.java` en la carpeta de trabajo (Mis Documentos\Curso Java\Clase1)



# Práctica 1 - Configuración del ambiente

---



## Setear el path del JDK

**C:\>Archivos de programa\Java\jdk1.6.0\bin**



```
SET JAVA_HOME=C:\jdk1.6.0  
SET PATH=%PATH%;%JAVA_HOME%\bin
```



# Práctica 1 – Compilación

---

- Abrir cmd e ingresar a la carpeta de trabajo
- Compilar el archivo HolaMundo.java

```
C:\>javac HolaMundo.java
```

- Si se detectan errores sintácticos se despliegan mensajes para indicar esto.
- Se generar los bytecodes que se almacenan en un archivo .class





# Práctica 1 – Ejecución

---

- Ejecutar el archivo Hola

```
C:\>java Hola
```

```
C:\Users\Guido\Documents\Programas>java Hola  
Hola, mundo!
```



# Práctica 1 – Detalles

---

```
//Hola.java
public class Hola {
    public static void main(String[ ] args) {
        System.out.println("Hola, mundo!");
    }
}
```

- Todo en Java está dentro de una clase, incluyendo programas autónomos.
- El código fuente se guarda en archivos con el mismo nombre que la clase que contienen y con extensión “.java”.
- El punto de entrada de un programa Java es la función *main*. La máquina virtual ejecuta el programa invocando esta función.
- La palabra reservada “void” indica que el método main no devuelve nada.
- El método main debe aceptar un array de objetos tipo String.



# Práctica 1 – Detalles

---

```
//Hola.java
public class Hola {
    public static void main(String[ ] args) {
        System.out.println("Hola, mundo!");
    }
}
```

- La palabra reservada "static" indica que el método es un método de clase, asociado a la clase en vez de una instancias de la misma. El método main debe ser estático.
- La palabra reservada public significa que un método puede ser llamado desde otras clases, o que la clase puede ser usada por clases fuera de la jerarquía de la propia clase.
- La función System.out.println nos permite mostrar una línea de texto en la pantalla cuando ejecutado el programa.



# Argumentos de main

---

*// Hola.java*

```
public class Hola {  
    public static void main(String[ ] args) {  
        System.out.println("Hola "+args[0]);  
    }  
}
```

## Ejecución

java Hola Guido



# Práctica 2 - Eclipse



*// Hola.java*

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

