Introducción al lenguaje Java

Jorge I. Meza
jimezam@autonoma.edu.co

Contenidos

- Origen de Java
- Motivación
  - Máquina virtual de Java
- Ventajas / desventajas
- Herramientas de software
- Palabras reservadas de Java
- Unidad mínima de compilación
  - Hola Mundo

Origen de Java

- Su desarrollo inición en 1991 y vió la luz en 1995 con la versión 1.0. Desarrollado por Sun Microsystems.
- Adquirido en 2010 por Oracle Corporation.

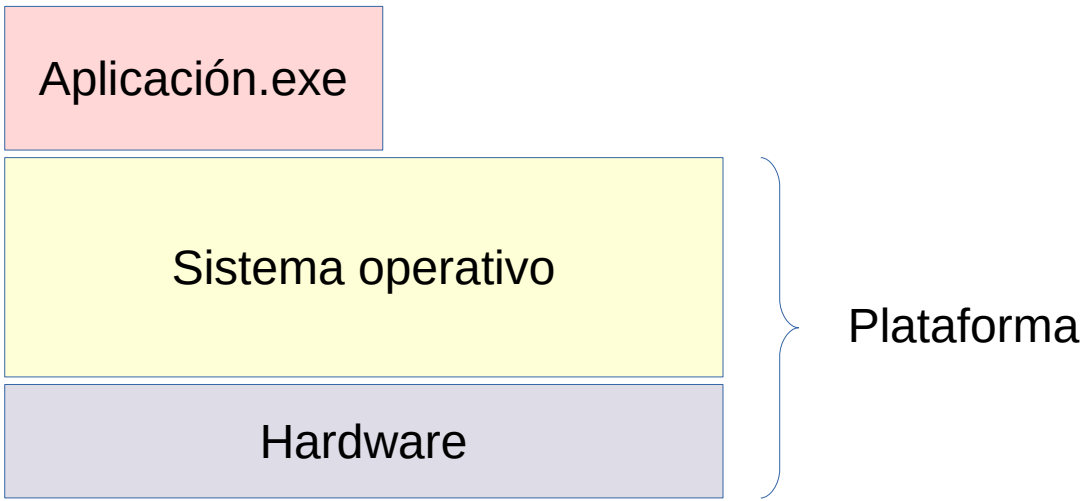
Motivación

- Aprovechar el paradigma orientado a objetos en auge en los 90's.
- Aprovechar el crecimiento de las redes, en particular Internet y la web (*Applets*).
- **Solventar la *dependencia de plataforma* de lenguajes como C y C++**.

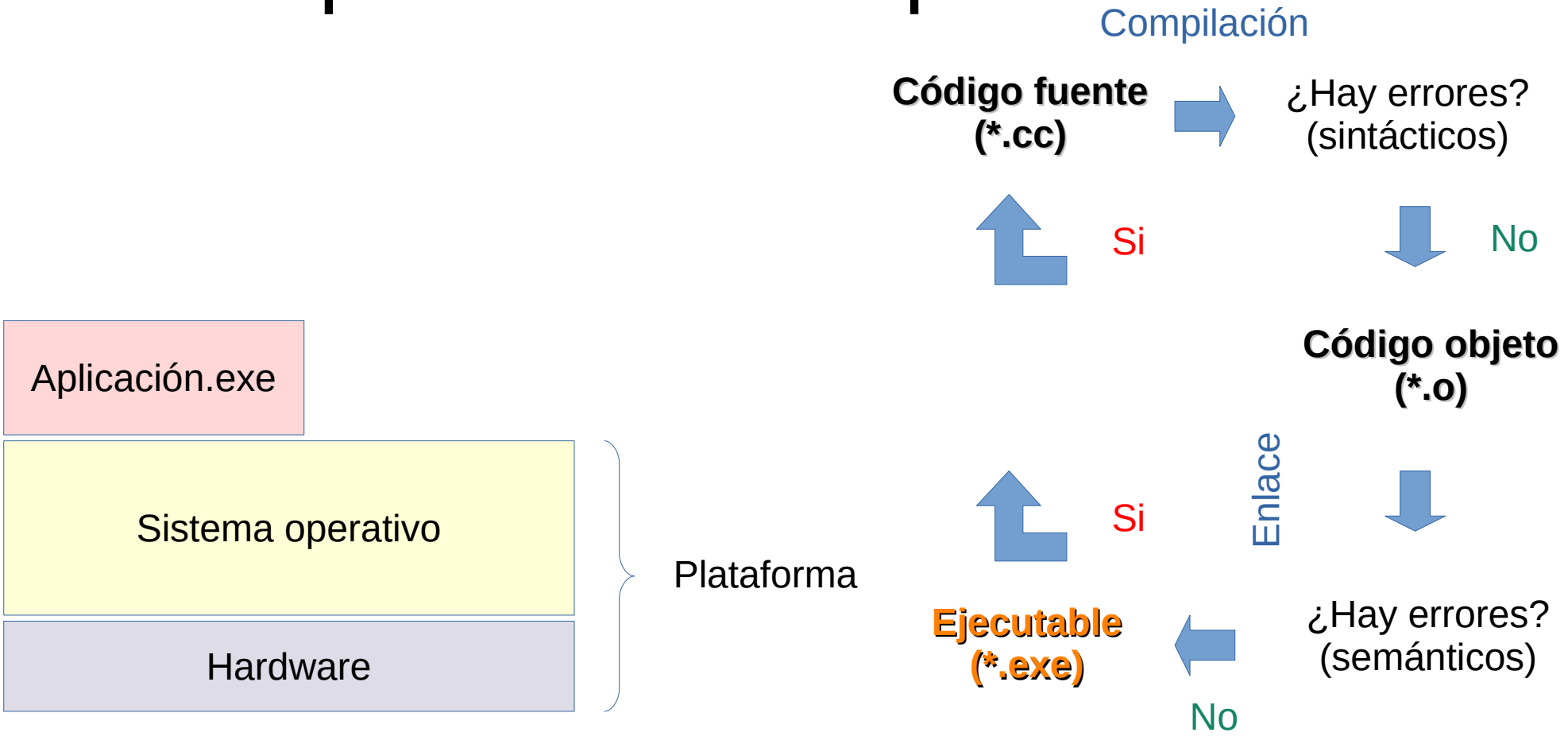
Motivación

"*Write it once, run it anywhere*".

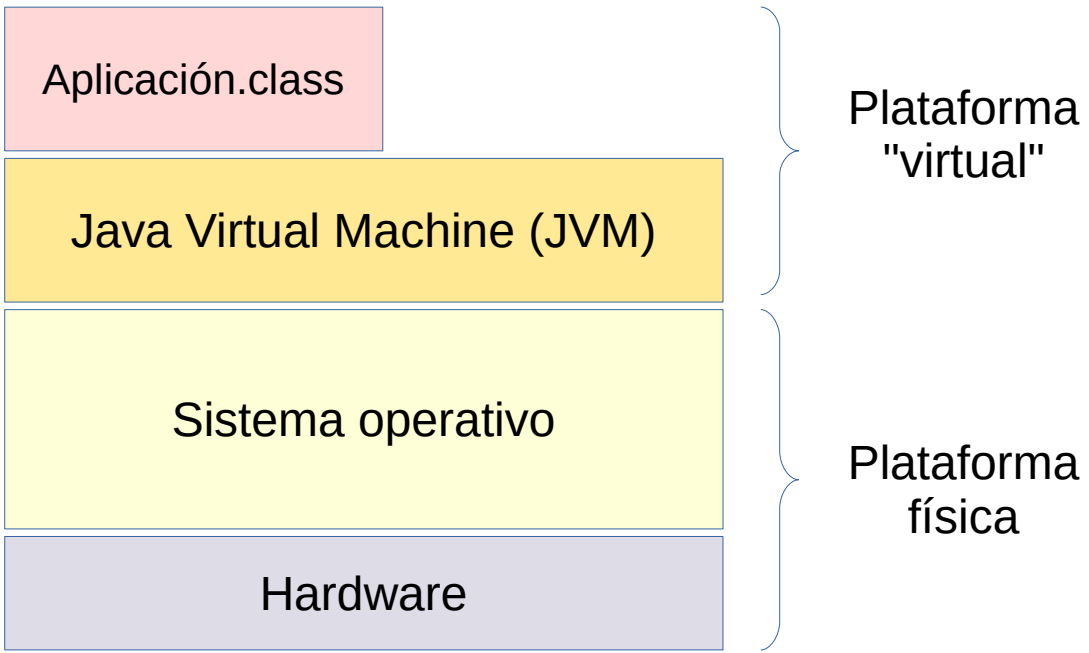
Dependencia de plataforma



Dependencia de plataforma

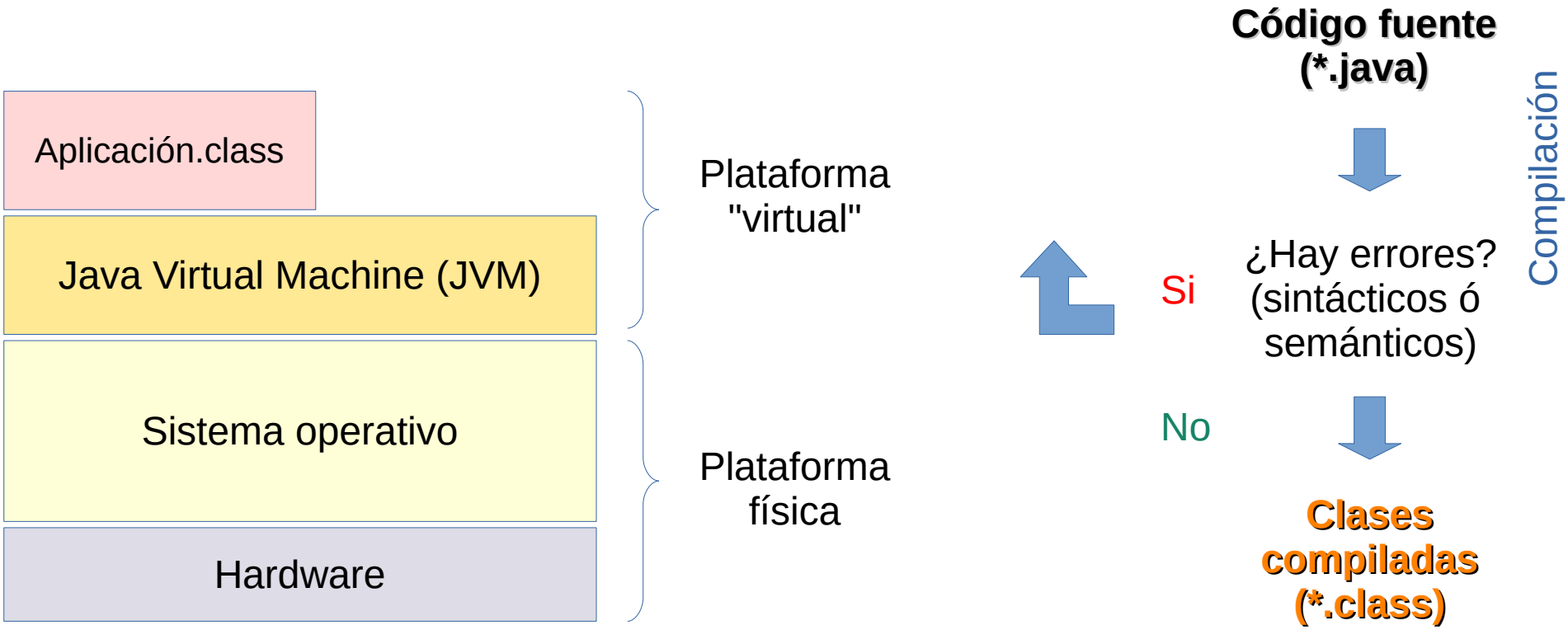


INdependencia de plataforma



- La JVM es dependiente de la *plataforma física*.
- Las aplicaciones hechas en Java son dependientes de la *plataforma virtual*.
- Las aplicaciones Java se podrán ejecutar en cualquier máquina que tenga la *plataforma virtual*, es decir, la JVM.

INdependencia de plataforma



INdependencia de plataforma

- La estrategia que utiliza Java para garantizar la independencia de plataforma en sus aplicaciones o *portabilidad* es utilizar una Máquina Virtual o JVM.
- Esta es un *middleware*, es decir, una capa adicional que separa a las demás creando una abstracción/separación entre ellas, facilitando su uso mediante una interfaz unificada.

Ventajas de Java

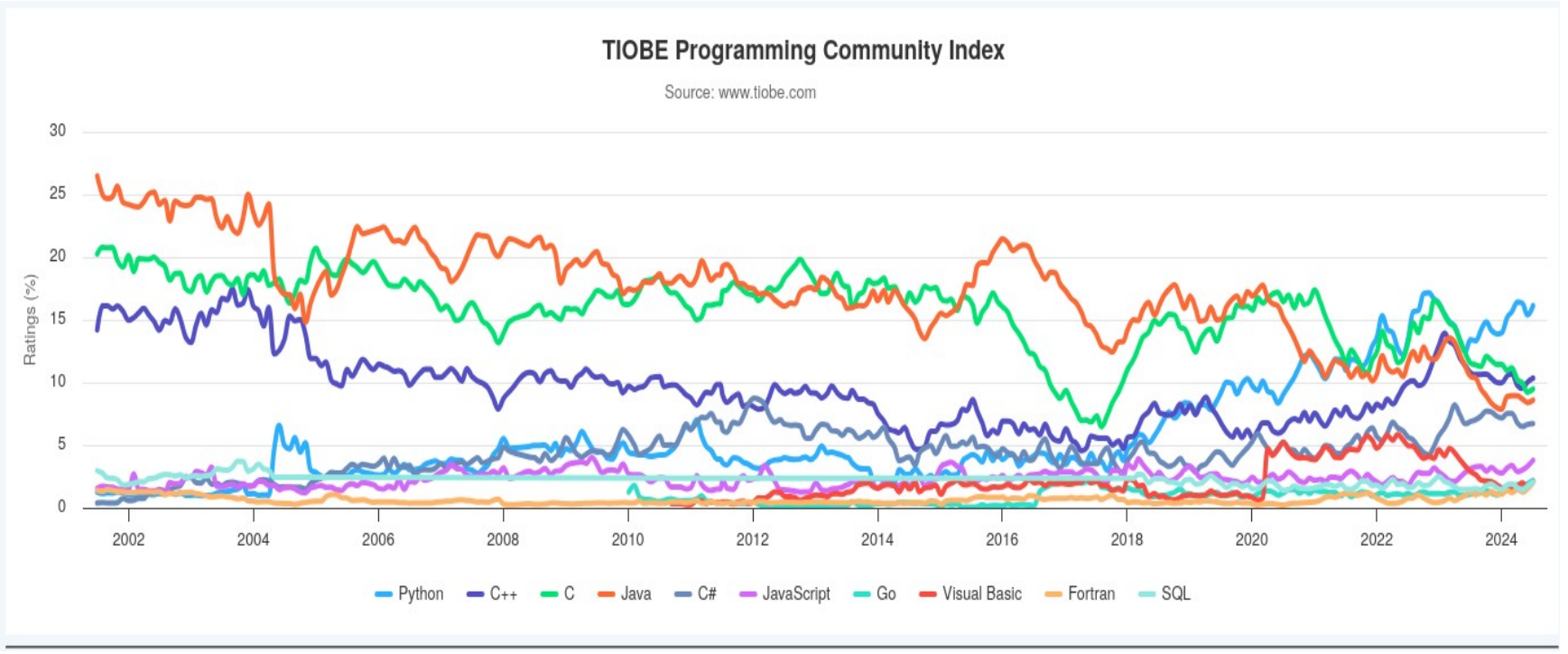
- Es uno de los lenguajes orientados a objetos moderno más utilizado en diferentes ámbitos, incluyendo el empresarial.
- Es portable.
- Incluye amplias herramientas que lo hacen un lenguaje seguro.
- Ofrece un amplio API al desarrollador.
- Tiene un gran ecosistema y comunidades activas.

Desventajas de Java

- Menor rendimiento y mayor uso de recursos.
- Código algunas veces complejo (*verbose*).
- Actualizaciones -demasiado- frecuentes.
- Problemas de seguridad (popularidad).
- No hay un GUI de escritorio oficial.
- Compra por parte de Oracle → Licenciamiento

Programming Language
Python
C++
C
Java
C#
JavaScript
Go
Visual Basic
Fortran
SQL
Delphi/Object Pascal
MATLAB
Rust
Ruby
Scratch
PHP
Swift
Assembly language

Caso práctico



Julio de 2024 - https://www.tiobe.com/tiobe-index/

Caso práctico: otras opciones

- Java
- C#
- C++
- Go

Fuertemente tipado

Tipado de manera estática

- Python
- Javascript
- PHP
- Ruby

Débilmente tipado

Tipado de manera dinámica

Herramientas de software

- Para desarrolladores
  - Java Development Kit (JDK)
    - https://jdk.java.net/
- Para usuarios
  - Java Runtime Edition (JRE)
    - https://www.java.com/download/manual.jsp

Integrated Development Environment

- Es un "editor con vitaminas" que trae herramientas que agilizan el desarrollo.
- Su uso es opcional pero muy conveniente.
- Existen varias opciones para Java: IntelliJ IDEA, Visual Studio Code, Eclipse.
- Para este curso se utilizar **Netbeans**. https://www.codelerity.com/netbeans/

Palabras reservadas de Java

```
abstract    assert      boolean     break       byte
case        catch       char        class       const
continue    default     do          double      else
enum        extends     final       finally     float
for         goto        if          implements  import
instanceof  int         interface   long        native
new         null        package     private     protected
public      return      short       static      strictfp
super       switch      synchronized this       throw
throws      transient   try         void        volatile
while
```

Unidad mínima de compilación

- Consiste en la estructura de código mínima para un programa válido, el cual puede ser compilado, enlazado (si es el caso) y ejecutado.
- En Java, consiste en una clase con su respectivo método `main` como se muestra a continuación.
- Estos conceptos de clase y método se verán más adelante en el curso.

Unidad mínima de compilación

```
class MiClase {
    public static void main(String[] args) {
        // Código fuente
    }
}
```

Hola Mundo

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

Condiciones mínimas

- Se debe escribir una clase por archivo de código fuente.
- El archivo se debe llamar exactamente igual a la clase con la extensión `.java`.
- En este ejemplo, la clase se llama `HolaMundo`, debe estar almacenada en el archivo `HolaMundo.java`.

Compilación y ejecución manual

- Compilación

```
$ javac HolaMundo.java
```

- Ejecución (interpretación)

```
$ java HolaMundo
```

`java` viene con JDK y JRE mientras que `javac` viene con JDK únicamente.

Demostración de uso con Netbeans

¿Alguna  pregunta?