

Introducción al lenguaje Java

Jorge I. Meza

jimezam@autonoma.edu.co



Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International

This license requires that reusers give credit to the creator. It allows reusers to distribute, remix, adapt, and build upon the material in any medium or format, for noncommercial purposes only. If others modify or adapt the material, they must license the modified material under identical terms.

👤 **BY:** Credit must be given to you, the creator.

🚫 **NC:** Only noncommercial use of your work is permitted.

Noncommercial means not primarily intended for or directed towards commercial advantage or monetary compensation.

🔄 **SA:** Adaptations must be shared under the same terms.

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 inició en 1991 y vio la luz en 1995 con la versión 1.0. Desarrollado por Sun Microsystems.
- Adquirido en 2010 por Oracle Corporation.

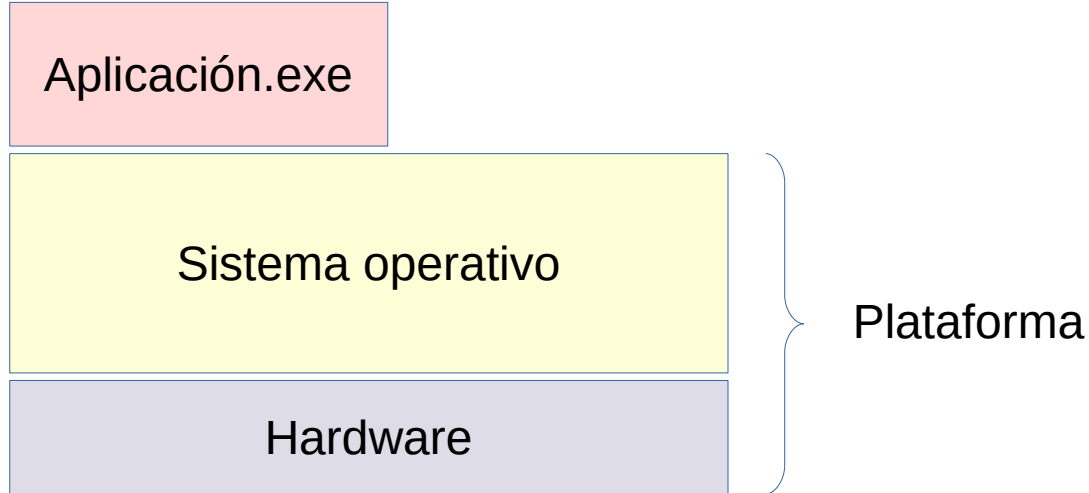
Motivación

- 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

Compilación

Código fuente
(*cc)



¿Hay errores?
(sintácticos)



Si



No

Código objeto
(*o)



Si



¿Hay errores?
(semánticos)

Enlace

Ejecutable
(*exe)

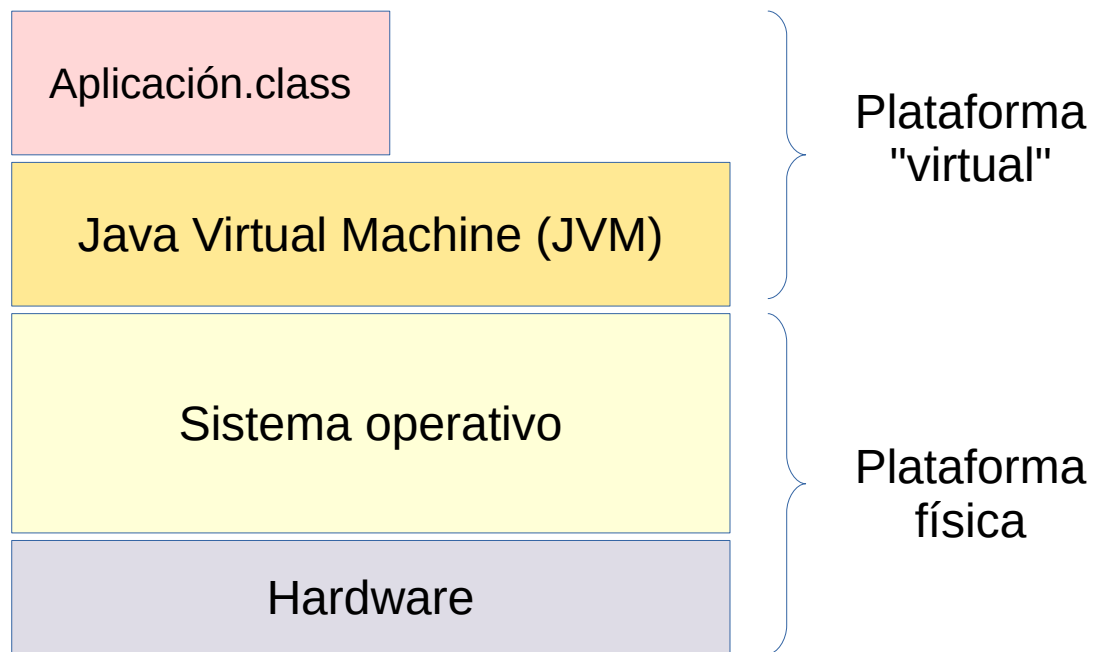


No

Plataforma

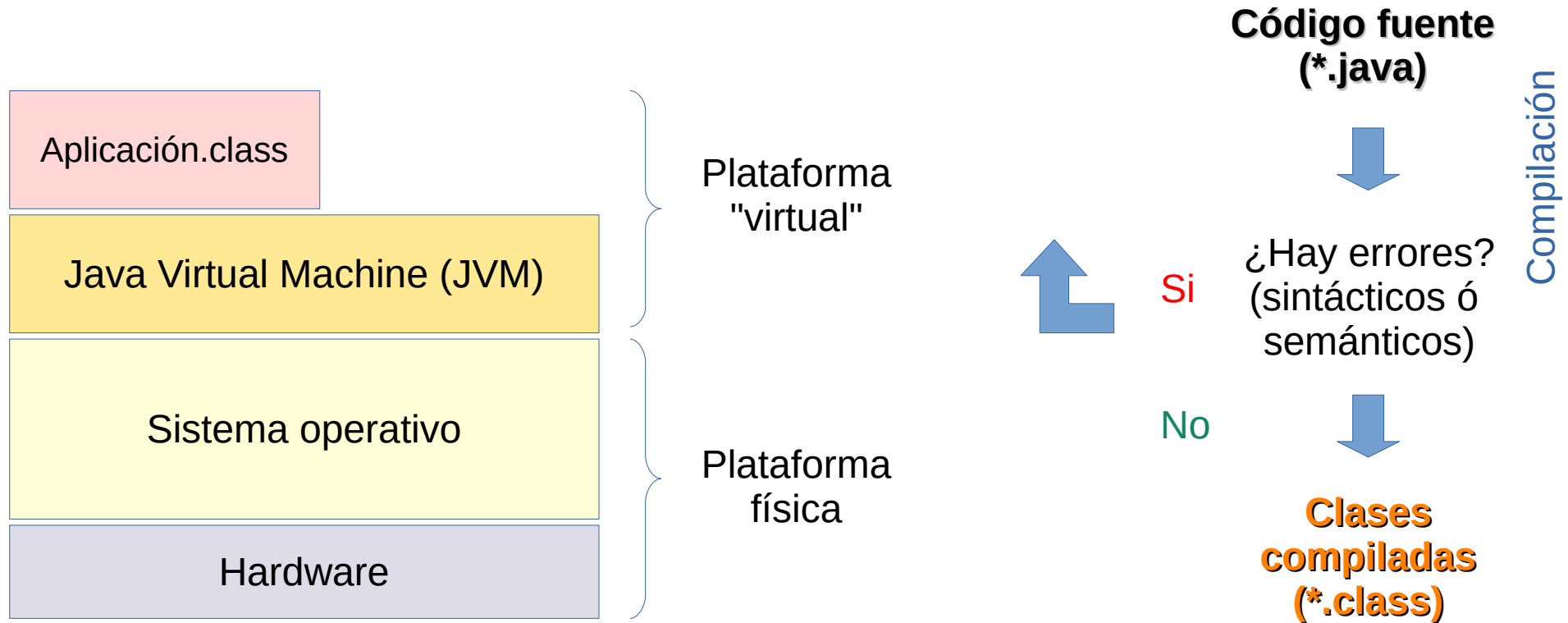


INdependencia de plataforma



- 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

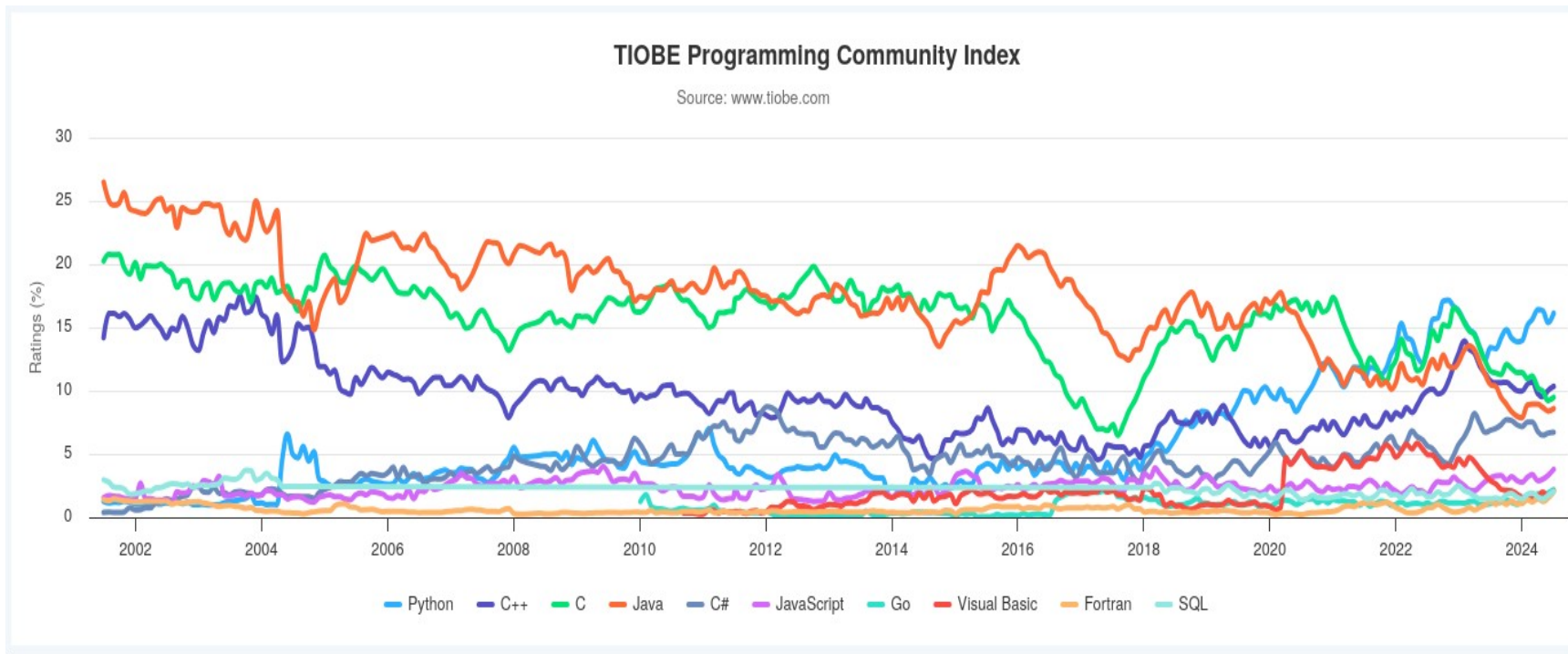


Swift



Assembly language

Caso práctico



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

Caso práctico: otras opciones

• Java	Fuertemente tipado Tipado de manera estática
• C#	
• C++	
• Go	
• Python	Débilmente tipado Tipado de manera dinámica
• Javascript	
• PHP	
• Ruby	

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

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

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?

