El Lenguaje y la Plataforma JAVA

La Plataforma abarca dos aspectos:

- Una Plataforma de Software
- Un Lenguaje de Programación

La Plataforma de ejecución JAVA

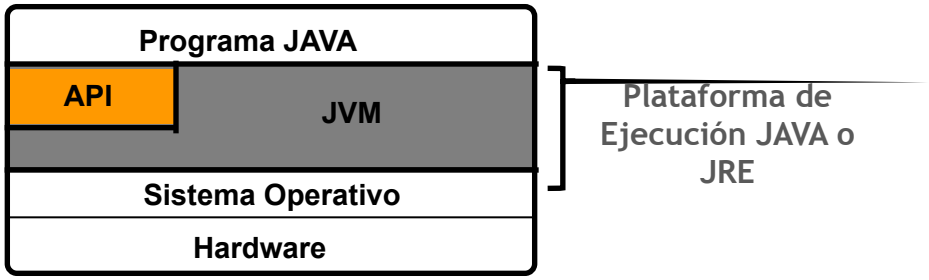
Una ***plataforma de ejecución*** es la combinación de un hardware y un sistema operativo que provee los servicios necesarios para ejecutar programas. Ejemplos de plataformas de ejecución son: Linux sobre una computadora Intel I7, Solaris ejecutándose sobre hardware Sun, etc.

La ***plataforma de ejecución JAVA*** provee los servicios necesarios para ejecutar programas escritos en JAVA. La plataforma JAVA se ejecuta sobre cualquier plataforma de ejecución.

A la plataforma de ejecución JAVA se la denomina JRE (Java Runtime Environment) o Entorno de Ejecución JAVA.

La plataforma de ejecución JAVA está compuesta por una ***Máquina Virtual JAVA*** o ***JVM*** (Java Virtual Machine) y un conjunto de librerías de código JAVA compilado, comúnmente denominado Interface de Programación de Aplicaciones JAVA o API (del inglés Application Programming Interface). La JVM es parte del JRE.

Los programas escritos en JAVA se ejecutan sobre la plataforma de ejecución JAVA.



Repaso: compilar e interpretar

El ***lenguaje de máquina*** o ***código binario*** consiste en instrucciones muy simples que la CPU de la computadora ejecuta directamente. Cada tipo de procesador tiene su propio lenguaje de máquina.

Los programas se escriben en ***lenguajes de programación de alto nivel***, como Java, C++, Delphi, Phyton, etc. Un programa escrito en un lenguaje de alto nivel no puede ejecutarse directamente en una computadora, necesita ser traducido al lenguaje de máquina de la computadora dónde se ejecutará. Este proceso de traducción lo realiza un programa llamado ***compilador***.

Típicamente, el código ejecutable es particular de la plataforma de ejecución donde se ejecutará el programa; es dependiente de la plataforma de ejecución.

Una alternativa a compilar un programa escrito en un lenguaje de alto nivel, es interpretarlo: un ***intérprete*** es un programa que traduce y ejecuta un programa escrito en un lenguaje de alto nivel, instrucción por instrucción. A diferencia del compilador, el intérprete no genera un código ejecutable, sino que traduce a código binario y ejecuta, de a una línea por vez, cada vez que se ejecuta el programa.

La Máquina Virtual

La ***Máquina Virtual Java*** o ***JVM*** es software y constituye el corazón del JRE.

El lenguaje de máquina de la Máquina Virtual Java es la ***codificación de bytes*** o ***bytecodes***. Es un lenguaje intermedio entre la codificación binaria específica del procesador y el lenguaje de alto nivel.

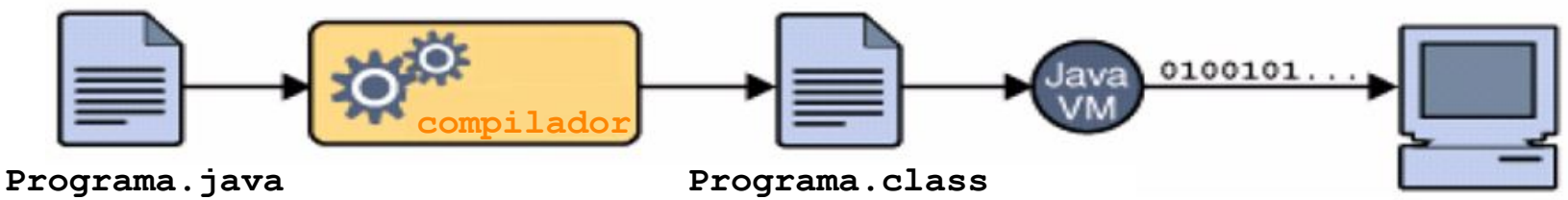
Una de las componentes fundamentales de la JVM es el intérprete JAVA, su responsabilidad es traducir *bytescodes* a código binario específico y luego ejecutarlo.

Como ya vimos, el código ejecutable es dependiente de la plataforma de ejecución particular -> necesitamos una JVM particular para cada tipo de plataforma de ejecución. Una máquina virtual para Windows, para Linux, para Solaris, etc.

La JVM es un mediador entre los programas JAVA compilados y la plataforma de ejecución específica de la computadora. Sus principales funciones son: traducir y ejecutar bytecodes, administrar la memoria del sistema de ejecución, proveer un sistema de seguridad y balancear la ejecución de múltiples threads.

Los Programas JAVA son de Plataforma Neutral

En JAVA el código fuente tiene extensión *.java*. Los archivos fuentes *.java* se compilan al lenguaje de máquina de la máquina virtual JAVA (JVM). Los archivos compilados tienen extensión *.class*. Un archivo *.class* no contiene código binario para un procesador específico, sino que contiene *bytecodes*, que es la codificación que entiende la máquina virtual de JAVA.



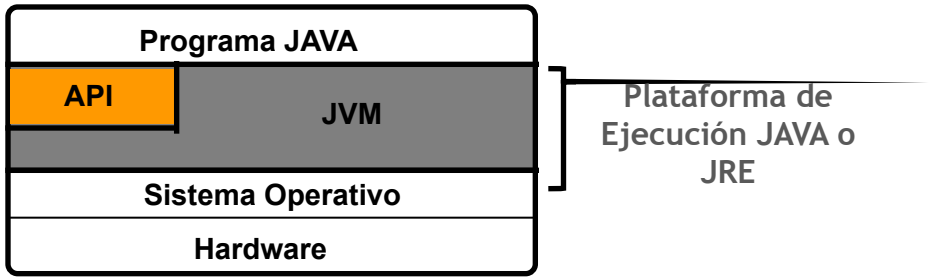
- JAVA es un lenguaje *compilado* e *interpretado*.
- Para ejecutar un programa JAVA compilado (.class) en una computadora SPARC con Solaris es necesario disponer de la JVM para Solaris, si lo vamos a ejecutar en una computadora Intel con Windows necesitamos la JVM para Windows. El mismo programa JAVA compilado a bytecodes lo podemos ejecutar en diferentes computadoras, no es necesario volver a compilarlo.
- La JVM garantiza que todo programa JAVA es de plataforma neutral.

Más sobre la máquina virtual

Característica principal

- Aislar al programa Java del sistema operativo y del hardware sobre el que se está ejecutando => independencia de la plataforma de ejecución.

Los programas escritos en JAVA se ejecutan sobre la plataforma de ejecución JAVA.

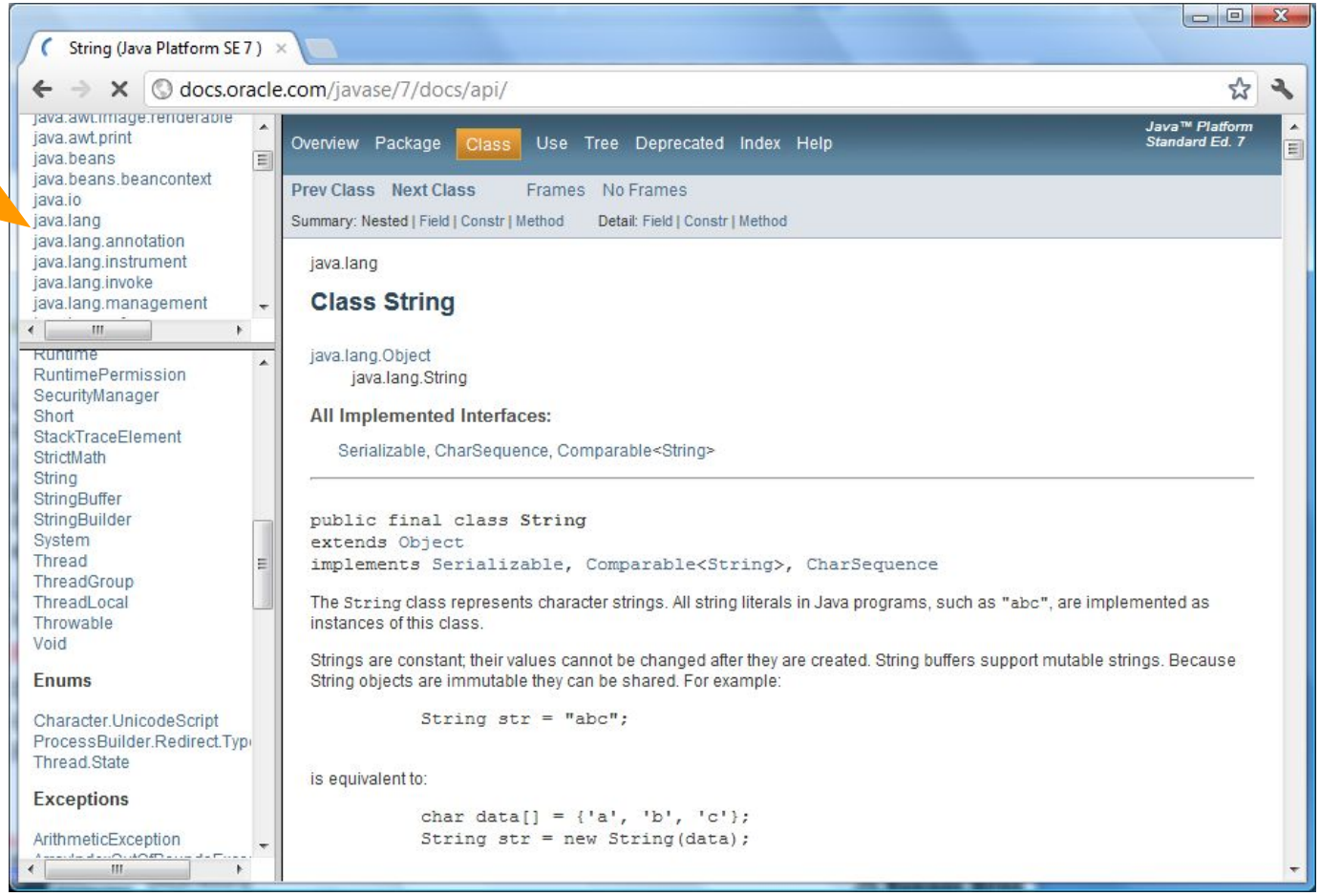


- La especificación de la JVM es única y permite que los programas Java sean independientes de la plataforma de ejecución ya que se compilan para una máquina genérica, la JVM, y se ejecutan en cualquier computadora que disponga de la JVM.
- La JVM asegura la portabilidad de los programas Java.
- La especificación de la JVM es un estándar. Cada sistema operativo tiene su propia implementación de la JVM.

API (Application Programming Interface)

La API JAVA es una colección de clases y otras componentes de software compiladas (archivos .class) que proveen una amplia gama de funcionalidades como componentes de GUIs, I/O, manipulación de colecciones, etc.

La API está agrupada en librerías de clases e interfaces Java relacionadas, llamadas paquetes. El programador puede combinar las componentes de la API JAVA con su código para crear una aplicación.

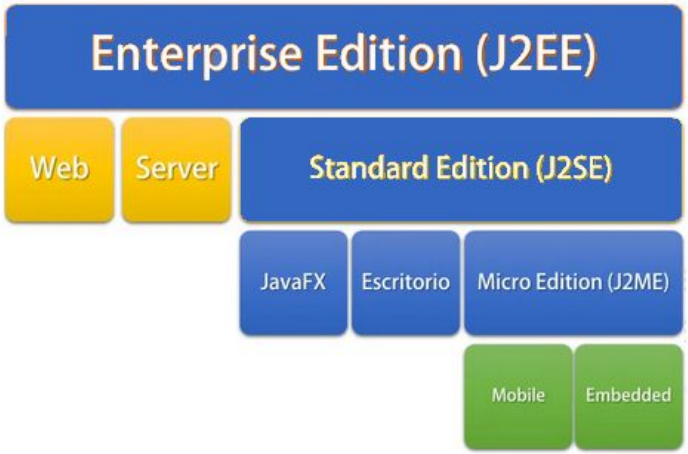


API (Application Programming Interface)

Algunos paquetes de la API JAVA son:

- `java.lang`: contiene clases esenciales como números, strings, objetos, seguridad y threads. Es el único paquete que se incluye automáticamente en todo programa Java.
- `java.io`: contiene las clases que manejan la Entrada/Salida, Serialización de objetos.
- `java.util`: contiene clases útiles que permiten manejar estructuras de datos o colecciones, fechas, hora, etc.
- `java.net`: contiene clases como URL, TCP, UDP, IP, etc. que permiten implementar aplicaciones de red. Provee soporte para sockets, direcciones IP, etc.
- `java.awt`: contiene clases para el manejo de interfaces gráficas de usuario (GUI) y manejo de eventos.
- `javax.swing`: proporciona clases para construir GUI mas sofisticadas con componentes como botones, tablas, listas y ventanas.
- `java.sql`: contiene clases para el manejo de base de datos relaciones.
- `javax.security`: proporciona clases e interfaces para implementar seguridad en aplicaciones Java, como autenticación, autorización, cifrado y manejo de firmas digitales.

La Plataforma Java: ediciones



- *JEE (Java Enterprise Edition)*: está diseñada para programar y ejecutar aplicaciones empresariales, caracterizadas por ser multiusuario y distribuidas. El procesamiento de estas aplicaciones se realiza en un servidor. Usualmente son aplicaciones web.
- *JSE (Java Standard Edition)*: está diseñada para programar y ejecutar applets y aplicaciones de escritorio JAVA. Típicamente son programas que se ejecutan en una PC. Es el fundamento de las 2 restantes ediciones. Está compuesta por el JRE y el JDK.
- *JME (Java Micro Edition)*: está diseñada para programar y ejecutar aplicaciones para dispositivos con recursos de cómputo limitados, como los dispositivos móviles antiguos, dispositivos embebidos y microcontroladores. Estos dispositivos cuentan con poca memoria RAM, pantallas muy chicas y muchas veces conexiones de red intermitentes.

Versiones de la edición Java SE

Oracle Java SE Support Roadmap*†			
Release	**GA Date**	**Premier Support Until**	**Extended Support Until**
8 (LTS)**	March 2014	March 2022	December 2030*****
9 - 10 (non-LTS)	September 2017 - March 2018	March 2018 - September 2018	Not Available
11 (LTS)	September 2018	September 2023	January 2032*****
12 - 16 (non-LTS)	March 2019 - March 2021	September 2019 - September 2021	Not Available
17 (LTS)	September 2021	September 2026****	September 2029****
18 (non-LTS)	March 2022	September 2022	Not Available
19 (non-LTS)	September 2022	March 2023	Not Available
20 (non-LTS)	March 2023	September 2023	Not Available
21 (LTS)	September 2023	September 2028****	September 2031****
22 (non-LTS)***	March 2024	September 2024	Not Available
23 (non-LTS)***	September 2024	March 2025	Not Available
24 (non-LTS)***	March 2025	September 2025	Not Available
25 (LTS)***	September 2025	September 2030	September 2033

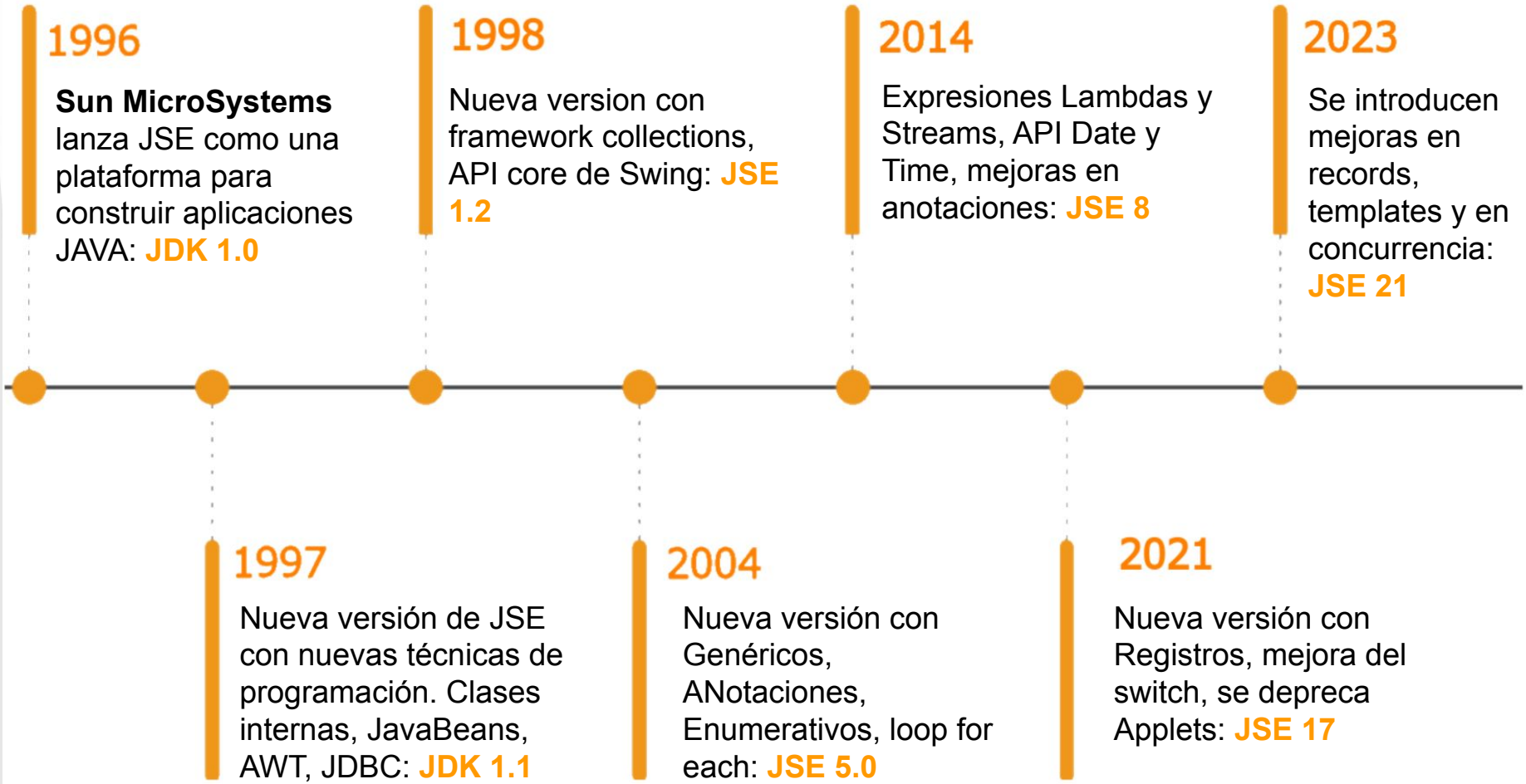
LTS significa **Long-Term Support** (Soporte a Largo Plazo) y se refiere a versiones de Java que reciben soporte y actualizaciones por un período prolongado, generalmente de varios años, en comparación con las versiones regulares que tienen un ciclo de vida más corto.

Las versiones **LTS** son elegidas por muchas empresas para sus entornos de producción debido a su estabilidad y soporte a largo plazo.

Oracle introdujo un ciclo de liberación programada para Java, donde cada 6 meses se lanza una nueva versión. Sin embargo, no todas estas versiones son **LTS**. Las versiones LTS se lanzan aproximadamente cada tres años.

Versiones de la edición JSE

Java Standard Edition



Java EE -> Jakarta EE

Java-Jakarta Enterprise edition