

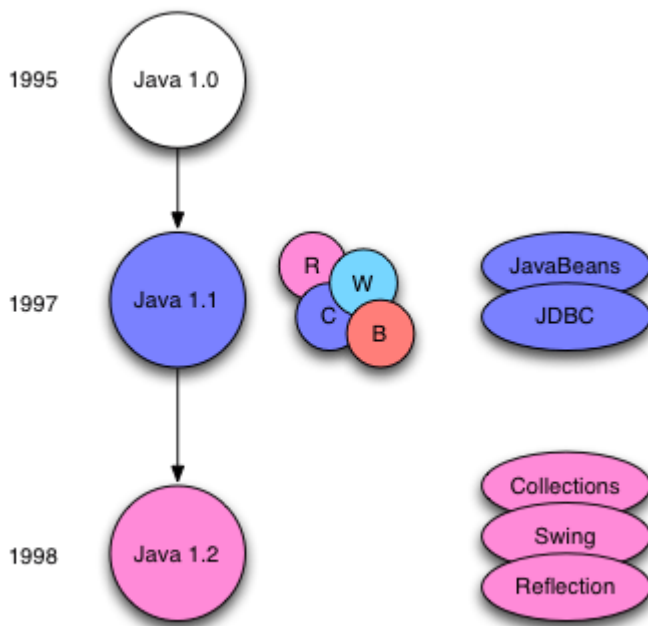


Las primeras versiones de Java han cumplido 25 años . Hemos pasado de la versión 1 a la versión 21 .¿Qué es lo que se ha ido añadiendo al lenguaje? . Vamos a echar un vistazo a cada una de las versiones y sus aportes

Versiones de Java (1-2)

Java aparece en 1995 como un nuevo lenguaje de programación con soporte

multiplataforma desarrollado por James Gosling y Sun Microsystems.



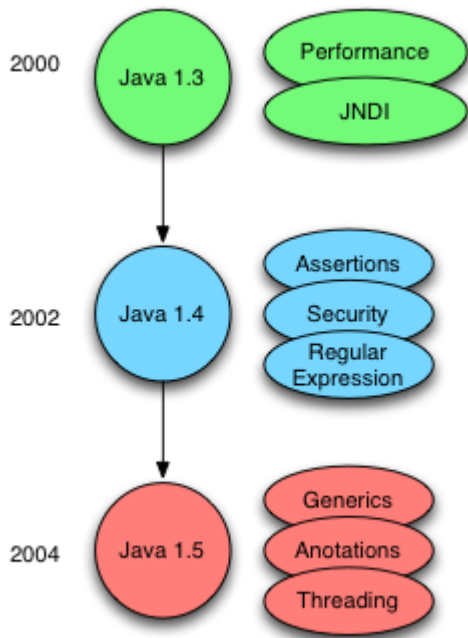
Versión 1.0: La primera versión del lenguaje contiene las clases principales , la maquina virtual y el API gráfico de AWT es un lenguaje que acaba de nacer.

Versión 1.1: Aparece en 1997 e incorpora al lenguaje varias clases que faltaban como Readers /Writers ,Calendars y Bundles . Pero sin ningún lugar a duda su mayor aportación es la inclusión del estandar de JavaBeans y el API de JDBC para conexión a bases de datos. Este último supone un salto importante en cuando a la utilización del lenguaje.

Versión 1.2: En 1998 aparece otra evolución importante con la llegada del framework de Collections y el API de Swing que permite desarrollar interfaces de ventanas más complejos.

Versiones de Java (1.3 -1.5)

Java en estos momentos ya es una plataforma madura a la cual el fabricante va añadiendo nuevas características.



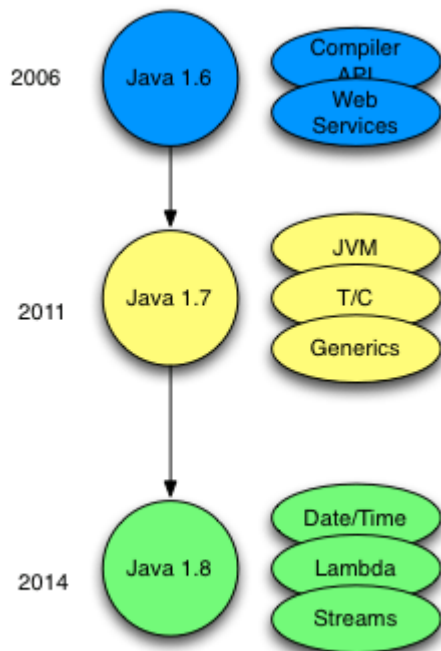
Versión 1.3: Avances pequeños en cuanto a APIs , se añade soporte JNDI. Sin embargo el avance en cuanto a la arquitectura de la maquina virtual es importante ya que aparece la máquina HotSpot con compilación JIT (Just-in Time).

Versión 1.4: Se produce un salto importante en cuanto a nuevas APIs. Se incorpora un fuerte soporte de XML , Expresiones Regulares , Criptografía etc.

Versión 1.5: También denominada Java 5 se producen dos saltos importantes a nivel del core del lenguaje. Por una parte la inclusión de tipos Genéricos que se echaban en falta en el mundo de las colecciones. Por el otro lado la inclusión del concepto de metadatos con el uso de anotaciones. Se amplía el soporte de APIs orientadas a programación concurrente.

Versiones de Java (1.6 -1.7)

Java ha madurado mucho y es una de las plataformas de referencia.



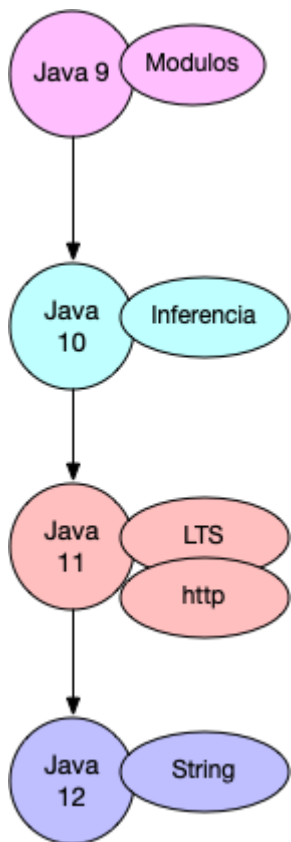
Versión 1.6: Esta versión contiene avances muy puntuales con la inclusión de un API de compilación “on-the-fly” que permitirá gestionar servicios web de forma cómoda.

Versión 1.7: Otra versión cuyos cambios a nivel del lenguaje son limitados. Se produce una mejora de la máquina virtual incluyendo nuevos recolectores de basura.

Version de Java (1.8)

Versión 1.8: Llega Java 8 el gran salto en cuanto al lenguaje se refiere. Se abren las puertas a la programación funcional con el uso de expresiones Lambda y Streams. Se realiza una revisión de APIs y se actualiza de forma importante la gestión de fechas.

Java 9 , 10 ,11,12



Versión:1.9 : En este release Java incluye el proyecto Jigsaw que permite modularizar el JDK.y la distribución de packages . Se trata de un proyecto clave para el futuro enfoque de MicroServicios.

Versión 10: Quizás la mejora más significativa es el añadido de la palabra var al lenguaje simplificando la inferencia de tipos cuando programamos de tal forma que a partir de ese momento es válido el uso de la siguiente estructura:

```
var lista= new ArrayList<Persona>()
```

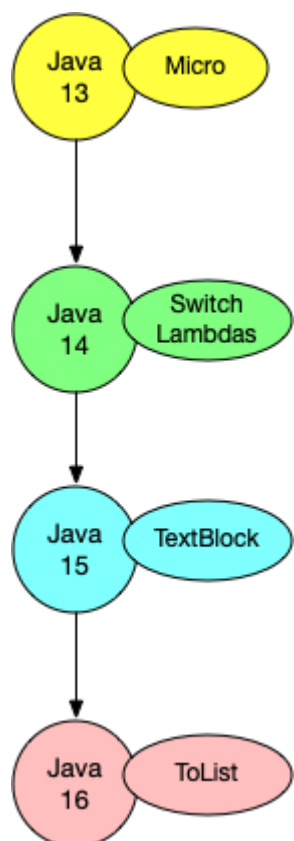
Aparte de esto mejoraron temas de recolector de basura y programación concurrente

Versión 11: Se estandariza el HTTPClient que se encontraba disponible desde la versión 9 pero en fase de pruebas y pasa a ser una de las librería de referencia para realizar peticiones HTTP en Java Eliminación del motor de Nashorn que servía para ejecutar JavaScript.

Version 12 : Añade la suite de MicroBenchMark que permite al desarrollador realizar pruebas de rendimiento dentro del propio JDK. Asi como nuevos métodos sencillos de manejo de Strings como `isBlank()` e `indent()`.

Java 13,14,15,16

Todas estas versiones salen en prácticamente 2 años y a nivel del desarrollador añaden algunas cosas a destacar:



Version 13 : Añade la suite de MicroBenchMark que permite al desarrollador realizar pruebas de rendimiento dentro del propio JDK.

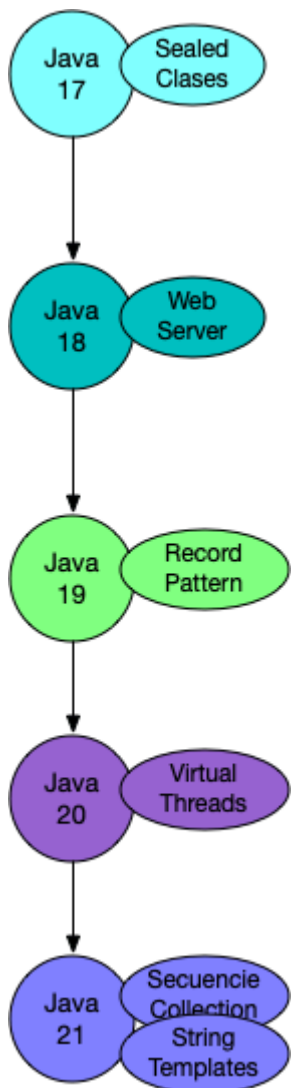
Version 14 : Mejora en las switch expression admitiendo lambdas

Version 15: Añade Los **TextBlock** que permiten definir String en formatos multilinea.

```
public class EjemploTextBlock {  
    public static void main(String[] args) {  
        String html = ""  
            <html>  
                <body>  
                    <p>Hola, mundo!</p>  
                </body>  
            </html>  
            "";  
        System.out.println(html);  
    }  
}
```

Version 16: Añade el método `toList()` a los Streams para simplificar el manejo de Collectores

Java 17,18,19,20



Java 17 : Quizás la característica más importante de Java 17 en cuando a los desarrolladores es que admite clases selladas que permiten limitar la herencia:

```
sealed interface Figura permits Circulo, Rectangulo, Triangulo {  
    double area();  
}
```

Java 18: Incluye un pequeño servidor web para pruebas sencillas

Java 19: Record patterns y el manejo de tipos de datos con ellos

Java 20: Los virtual Threads que permiten una gestión mucho más optima de los temas de programación concurrente.

Java 21: La novedad quizás más importante es el uso de template String que aparece como preview-

Otros artículos relacionados

- [¿Es el fin de los servidores Java EE?](#)
- [Java EE vs Spring](#)
- [Java Record Class y JDK 14](#)