

# Fundamental

## 1. Lenguajes de Programación

“Un lenguaje de programación es un lenguaje formal diseñado para realizar procesos que pueden ser llevados a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila (de ser necesario) y se mantiene el código fuente de un programa informático se le llama programación.” - Wikipedia

### 1.1. Implementación

Se tiene dos formas principales de implementar un lenguaje de programación:

1. Compilado
2. Interpretado

#### 1.1.1. Compilado

Se refiere a traducir un programa en un lenguaje de programación a otro lenguaje de programación, de forma que se consideren equivalentes y de esta forma pueda ser interpretado por la máquina. El programa que realizan esta traducción se les denomina compiladores.

Con respecto a lenguaje de programación que puede ser interpretado por la máquina, se le denomina lenguaje máquina y este varía según el microprocesador que lo interpreta. A este conjunto de instrucciones es parte de lo que se denomina arquitectura de la computadora.

#### 1.1.2. Interpretado

En este caso se interpreta un conjunto de símbolos (el lenguaje de programación), dado sentido a estas instrucciones, pudiendo traducirlo directamente a lenguaje máquina conforme lo interpreta o simplemente ejecutándolo. Los lenguajes interpretados otorgan mucha flexibilidad porque suelen ser independientes de la máquina que lo ejecuta.

## 2. Java

Java es un lenguaje de programación orientado a objetos. permite que los desarrolladores escriban el programa una vez y lo ejecuten en cualquier dispositivo (WORA, o "*write once, run anywhere*"), es decir, el código que es ejecutado en una plataforma no tiene que ser recompilado para ejecutarse en otra.

Fue originalmente desarrollada por James Gosling, de Sun Microsystems, y publicado en 1995 como un componente fundamental de la plataforma Java de Sun Microsystems. Su sintaxis deriva de C y C++. Una aplicación Java se suele compilar a *bytecode* y puede ejecutarse en cualquier máquina virtual Java (JVM Java Virtual Machine) sin importar la arquitectura de la computadora subyacente.

Gráficamente:

Aplicación (en bytecode)
JVM
S.O.

Para poder crear un programa en Java, se requiere de un JDK (*Java Development Kit*) el cual varía por versiones del lenguaje. Para poder ejecutar un programa se requiere un entorno de ejecución conocido como JRE (*Java Runtime Environment*) correspondiente a una versión.

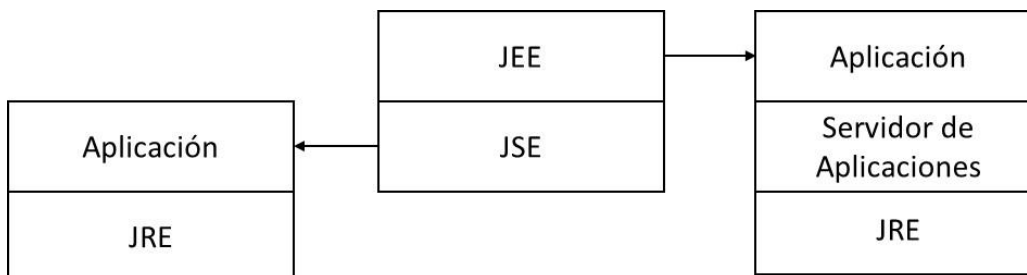
## 2.1. Especificaciones Java

La evolución del lenguaje es regulada por la JCP (*Java Community Process*) en la cual se usa las JSR (*Java Specification Request*) para proponer y especificar mejoras. El lenguaje está especificado en la JLS (*Java Language Specification*) el cual se gestiona en el JSR 901.

## 2.2. JSE y JEE

JSE proviene de *Java Standard Edition* y JEE de *Java Enterprise Edition*. La principal diferencia entre estos es que el segundo requiere un contenedor tal como un servidor de aplicaciones para poder ejecutarse.

Gráficamente, teniendo en cuenta el tipo de aplicación a ejecutar (JSE o JEE) tendríamos:



Como se puede observar, una aplicación JSE requiere solo un JRE para ejecutarse, mientras que una aplicación JEE requiere un contenedor (el servidor de aplicaciones).

Dado que la JSE como JEE están definidos en una especificación, y teniendo la relación gráfica anterior, esto conlleva a que una aplicación JEE hereda las especificaciones que se habilitan para una aplicación JSE pero no al revés. Por ejemplo, JAXB es una especificación para JSE y por tanto se puede utilizar en una JEE. Sin embargo, CDI es una especificación para JEE, puede utilizarse en este tipo de aplicaciones, pero no en una JSE. Para poder realizar esto último, se requiere encontrar un contenedor que le permita habilitar dicha especificación.

### 2.2.1. JSE

Las especificaciones JSE suele ser la que orienta la mejora del lenguaje y esta se desarrolla bajo alguna especificación. El nombre J2SE reemplaza al de JDK, posteriormente se denominará Java SE.

1. J2SE 1.4 – Nombre clave Merlin, JSR 59.

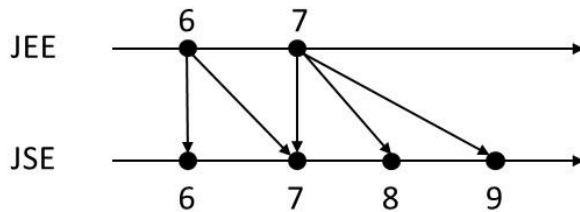
2. J2SE 5.0 – Nombre clave Tiger, JSR 176. También se le denomina 1.5
3. Java SE 6.0 – Nombre clave Mustang, JSR 270.
4. Java SE 7.0 – Nombre clave Dolphin, JSR 336
5. Java SE 8.0 – Nombre clave Spider, JSR 337
6. Java SE 9.0 – Nombre clave (none), JSR 379
7. Java SE 10.0 – Nombre clave (none), JSR 384

### 2.2.2. JEE

De forma similar, JEE es una especificación, pero en sí mismo agrupa a un conjunto de especificaciones para diversos propósitos. Algunas versiones fueron:

1. JEE 6 – JSR 313
2. JEE 7 – JSR 342
3. JEE 8 – JSR 366

Lo anterior también produce que se posible que un servidor de aplicaciones compatible con una versión Java EE, pueda ejecutarse en distintas versiones JSE:



Lo anterior tiene un efecto las configuraciones soportadas y entornos de ejecución para un servidor JEE compatible con una versión específica.