

# INTRODUCCIÓN A JAVA

# Índice

## 1.- Java

1.1 - ¿Qué es Java?

1.2 – ¿Qué es una variable?

1.3 – Tipos de datos primitivos

1.4 – Creando nuestro primer proyecto en Java

1.5 – Programando “Hola Mundo”

1.6 - Ejercicios

## 1.1 - ¿Qué es Java?

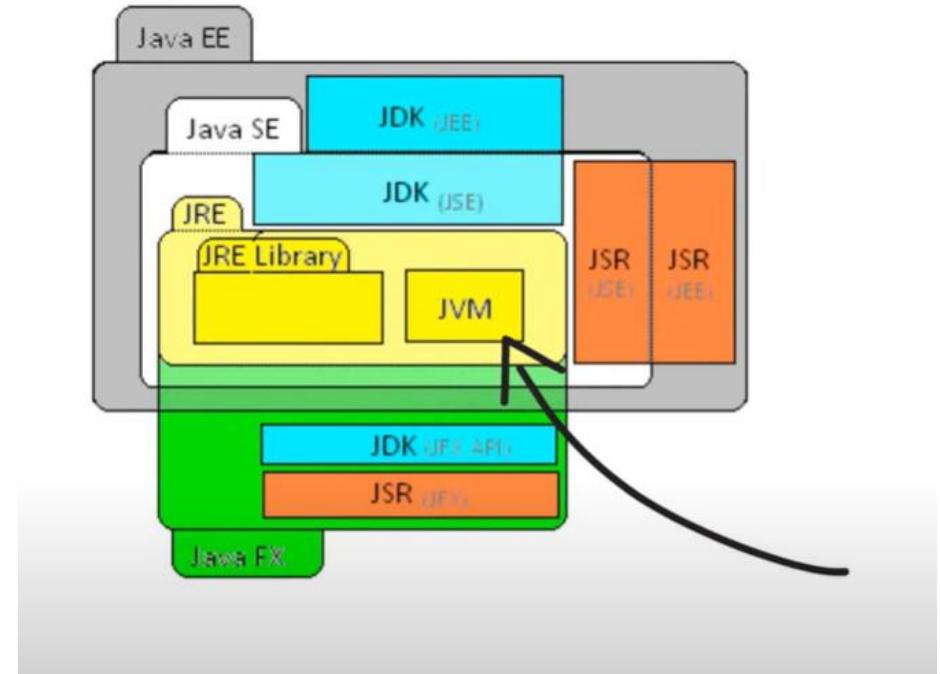
- Java es un lenguaje de programación basado en clases, orientado a objetos y concurrente. También llamamos Java a todo el entorno para ejecutar aplicaciones escritas en este lenguaje



PLATAFORMA DE JAVA

## 1.1 - ¿Qué es Java?

- Java es de propósito general. Esto quiere decir que sirve para programar en escritorio, web, servidores y hasta en dispositivos móviles.
- Hizo realidad el sueño de cualquier programador: escribir código fuente de un programa una sola vez y correrlo en cualquier plataforma
- Java logra esto gracias al corazón de toda la plataforma Java, la JVM (Java Virtual Machine)



## 1.1 - ¿Qué es Java?

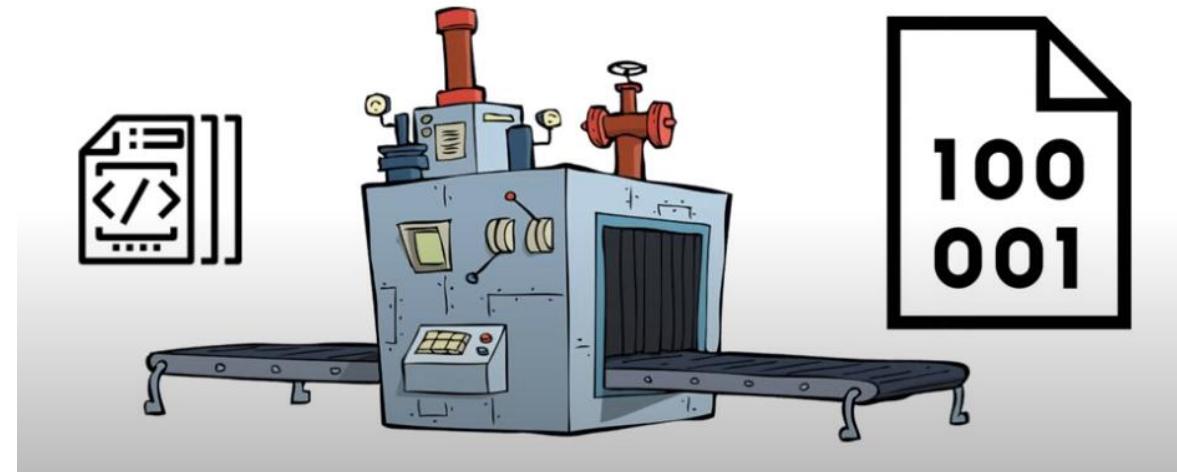
- Los ordenadores solamente entienden el lenguaje binario, también conocido como lenguaje máquina. Si no existieran los compiladores, nos veríamos obligados a programar usando solamente 0 y 1, lo cual es absurdo
- Afortunadamente, existen los lenguajes de programación, los cuales se asemejan más al lenguaje humano, los cuales se componen de palabras en inglés, números, caracteres especiales, etc...

```
00101110 00101110 00101110  
00101110 00101110 01110011  
01100100 00101110 00101110  
00101110 01010011 00101000  
01110011 00101110 00101110  
01101100 01101100 01101111  
01101100 01100100 01001110  
00101110 00101000 00101110
```

```
package rentalStore;  
import java.util.Enumeration;  
import java.util.Vector;  
  
class Customer {  
    private String _name;  
    private Vector<Rental> _rentals = new Vector<Rental>();  
  
    public Customer(String name) {  
        _name = name;  
    }  
    public String getMovie(Movie movie) {  
        Rental rental = new Rental(new Movie("", Movie.NEW_ RE  
        Movie m = rental._movie;
```

## 1.1 - ¿Qué es Java?

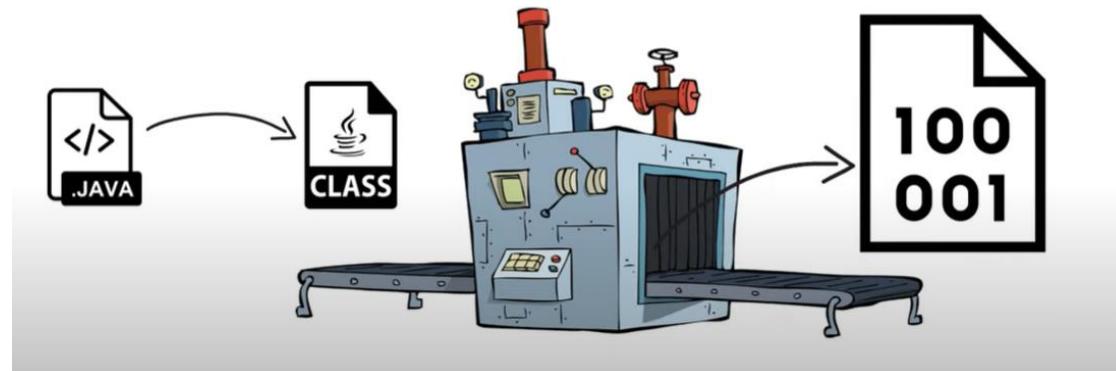
- El trabajo de los compiladores, pues, es traducir el lenguaje de programación que entendemos los humanos al lenguaje binario que entienden los ordenadores
- El problema es que las plataformas como Windows, Mac y Linux tienen sus propios dialectos de lenguaje máquina, de tal forma que un programa escrito para Linux, por ejemplo, no será entendido por el resto de plataformas



## 1.1 - ¿Qué es Java?

- Aquí es donde entra la JVM. El lenguaje Java no se compila directamente a código binario, si no a bytecode. Ese bytecode es interpretado por la JVM, que se encargará de generar el código binario.
- Existe una JVM por cada plataforma y es por esto que no debemos adaptar el código, si no que será esa JVM, una vez la instalamos, la que se encargará de traducir ese bytecode al binario de la plataforma en la que se encuentre

### JAVA VIRTUAL MACHINE



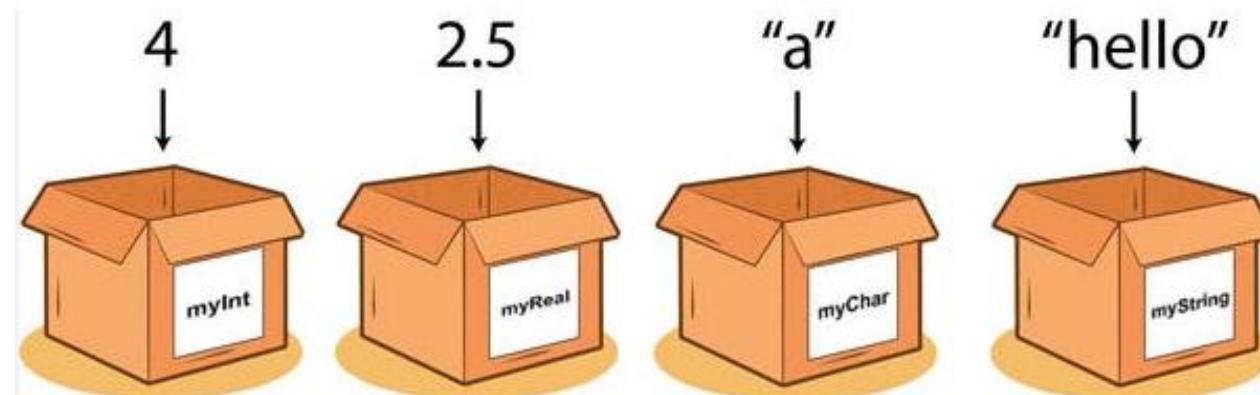
## 1.1 - ¿Qué es Java?

- La JVM ha tenido tanto éxito que otros lenguajes de programación diferentes de Java generan bytecode para ella



## 1.2 - ¿Qué es una variable?

- Cuando se habla de variable en Java se hace referencia a un contenedor de datos que almacena los valores durante la ejecución del programa Java. Dentro de este contenedor se le asigna a cada variable un tipo de dato que funciona para designar el tipo y la cantidad de valor que cada variable puede contener



## 1.3 – Tipos de datos primitivos

- Los más utilizados son **int**, **double**, **char**, **boolean**. **String** no es un dato primitivo, si no una clase. Ya lo explicaremos más adelante

| Nombre  | Tipo               | Ocupa   | Rango aprox.   |
|---------|--------------------|---------|----------------|
| byte    | Entero             | 1 byte  | -128 a 127     |
| short   | Entero             | 2 bytes | -32768 a 32767 |
| int     | Entero             | 4 bytes | 2,000,000,000  |
| long    | Entero             | 8 bytes | Muy grande     |
| float   | Decimal simple     | 4 bytes | Muy grande     |
| double  | Decimal doble      | 8 bytes | Muy grande     |
| char    | Carácter simple    | 2 bytes | ----           |
| String  | Cadena de texto    | ----    | ----           |
| boolean | Valor true o false | 1 byte  | ----           |

## 1.4 – Creando nuestro primer proyecto en Java

- Vamos a pulsar en File/New/Java Project (si no aparece en el listado, buscar en Other)
- En Project name pondremos el nombre del proyecto. Suele nombrarse empezando por minúsculas y usando camelCase (nuevoProyectoJava)
- En JRE nos deberá aparecer el que tenemos asignado por defecto. Es el Java Runtime Environment, una capa de software que se ejecuta encima del software del sistema operativo y proporciona las bibliotecas de clases y otros recursos que un programa Java específico necesita para ejecutarse.
- Desactivamos la casilla “Create module-info.java file” y le damos a Finish.

## 1.4 – Creando nuestro primer proyecto en Java

- En la carpeta del proyecto veremos que tenemos dos cosas: JRE System Library y una carpeta llamada src.
- JRE System Library es una librería que ha instalado nuestro JRE. Una librería, en términos generales, es una biblioteca llena de recursos que vamos a poder usar en nuestros proyectos. Nos facilita nuestra tarea con un conjunto de clases ya predefinidas y listas para ser usadas (como String o Scanner, por ejemplo). **LA LIBRERÍA NO SE TOCA**
- Botón derecho sobre la carpeta src, new, package y de nombre al paquete le vamos a poner controlador o controller
- Botón derecho sobre el paquete, new, Class y de nombre le vamos a poner Inicio o Main. **EL NOMBRE DE LAS CLASES SIEMPRE EMPIEZA POR MAYÚSCULA**
- Marcamos la casilla “public static void main(String[] args). Si se nos olvida, con poner main en la clase, el propio IDE nos lo va a sugerir

## 1.5 - Programando “Hola Mundo”

---

```
1 package primerosEjercicios;
2
3 public class PrimerEjercicio {
4
5     public static void main(String[] args) {
6
7         String saludo = "Hola Mundo";
8
9         System.out.println(saludo);
10
11
12
13     }
14 }
```

---

## 1.6 - Ejercicios

- 1 – Hacer un programa en el que pongamos 3 números enteros, los sume e imprima el resultado por pantalla
- 2 – Hacer un programa en el que pongamos una variable con un saludo y otra con nuestro nombre e imprima por pantalla la frase completa
- 3 – Hacer un programa en el que pongamos 4 notas de un alumno y su nombre y nos saque por pantalla una frase de este tipo: “El promedio de notas de (nombre del alumno) es de (promedio de notas)”