

Nombre: Stefany Chicaiza	
NRC: 1940	Fecha: 06/11/2024

ACTIVIDAD 1 – IP

JAVA

Java es un lenguaje de programación de propósito general, orientado a objetos, que fue desarrollado por Sun Microsystems (ahora parte de Oracle) en 1995. Está diseñado para ser sencillo, seguro y eficiente, con el objetivo de permitir que las aplicaciones funcionen en una amplia variedad de dispositivos y sistemas operativos.

DATOS PRIMITIVOS

E N T E R O	<div>int</div> <p>Este tipo de dato almacena números enteros (positivos, negativos y cero) sin decimales. Ocupa 4 bytes de memoria.</p> <pre>int numero = 25; System.out.println(numero); // Salida: 25</pre>	
	<div>double</div> <p>Este tipo almacena números decimales con más precisión y ocupa 8 bytes de memoria. Ideal para valores con decimales.</p> <pre>double precio = 19.99; System.out.println(precio); // Salida: 19.99</pre>	D E C I M A L
D E C I M A L	<div>Float</div> <p>También almacena números decimales, pero con menos precisión que <b>double</b>. Ocupa 4 bytes y se recomienda para datos decimales menos precisos.</p> <pre>float altura = 1.75f; System.out.println(altura); // Salida: 1.75</pre>	
	<div>Char</div> <p>Este tipo almacena un único carácter en formato Unicode. Ocupa 2 bytes de memoria.</p> <pre>char inicial = 'A'; System.out.println(inicial); // Salida: A</pre>	C A R A C T E R
B O L E A N O	<div>boolean</div> <p>Este tipo de dato almacena valores de verdad: <b>true</b> (verdadero) o <b>false</b> (falso). Ocupa 1 bit de memoria.</p> <pre>boolean esJavaFacil = true; System.out.println(esJavaFacil); // Salida: true</pre>	
	<div>Byte</div> <p>Un número entero muy pequeño, ocupa solo 1 byte y va de -128 a 127. Útil para ahorrar memoria en colecciones de datos grandes donde los valores no excedan ese rango.</p> <pre>byte edad = 30; System.out.println(edad); // Salida: 30</pre>	B Y T E

## SHORT

## short

Similar a `int` pero ocupa menos memoria (2 bytes) y es útil cuando no se necesitan números tan grandes. Va de -32,768 a 32,767.

```
short temperatura = -10;
System.out.println(temperatura); // Salida: -10
```

## Long

Almacena enteros grandes y ocupa 8 bytes de memoria, útil para valores que exceden el rango de `int`.

```
long distancia = 15000000000L;
System.out.println(distancia); // Salida: 15000000000
```

## LONG

## EJEMPLOS EN CÓDIGO

```
class pública Main {
    public static void principal ( cadena [] argumentos ) {
        int miNum = 15 ;
        Sistema . out . println ( myNum );
    }
}
```

15

```
class pública Main {
    public static void principal ( cadena [] argumentos ) {
        Cadena nombre = "Juan" ;
        Sistema . out . println ( "Hola " + nombre );
    }
}
```

Hola juan

```
public class Main {
    public static void main(String[] args) {
        String firstName = "John ";
        String lastName = "Doe";
        String fullName = firstName + lastName;
        System.out.println(fullName);
    }
}
```

John Doe

```
public class Main {
    public static void main(String[] args) {
        int x = 5;
        int y = 6;
        System.out.println(x + y); // Print the value of x + y
    }
}
```

11

```
public class Main {
    public static void main(String[] args) {
        // Student data
        String studentName = "John Doe";
        int studentID = 15;
        int studentAge = 23;
        float studentFee = 75.25f;
        char studentGrade = 'B';

        // Print variables
        System.out.println("Student name: " + studentName);
        System.out.println("Student id: " + studentID);
        System.out.println("Student age: " + studentAge);
        System.out.println("Student fee: " + studentFee);
        System.out.println("Student grade: " + studentGrade);
    }
}
```

```
Student name: John Doe
Student id: 15
Student age: 23
Student fee: 75.25
Student grade: B
```