

# Estructuras de control

## Selección

```
1 // IF-ELSE
2
3 if(condicion) {
4     // Sentencias si la condición es verdadera
5 } else {
6     // Sentencias si la condición es falsa
7 }
8
9 // SWITCH
10 // La expresión puede ser un valor
11 switch(expresion) {
12     case condicion1:
13         // Sentencias para condicion 1
14         break;
15     case condicion2:
16         // Sentencias para condicion 2
17         break;
18
19     ...
20
21     case condicionN:
22         // Sentencias para condicion N
23         break;
24 }
25
26 // Instrucción switch con tipo de datos int
27 int day = 5
28
29 switch (day)
30 {
31     case 1:
32         System.out.println("Lunes");
33         break;
34     case 2:
35         System.out.println("Martes");
36         break;
37     case 3:
38         System.out.println("Miércoles");
39         break;
40     case 4:
41         System.out.println("Jueves");
42         break;
43     case 5:
44         System.out.println("Viernes");
45         break;
46     case 6:
47         System.out.println("Sábado");
48         break;
49     case 7:
50         System.out.println("Domingo");
51         break;
52     default:
53         System.out.println("Día inválido");
54         break;
55 }
56
```

## Iteración

```

1 // DO-WHILE
2 do {
3     // Sentencias a ejecutar al menos una vez y mientras se cumpla la
      condicion
4 } while(condicion)
5
6 // WHILE
7 while(condicion) {
8     // Sentencias a ejecutar mientras se cumpla la condición
9 }
10
11 // FOR
12 for(variable; condicion; incremento/decremento) {
13     // Sentencias a ejecutar mientras se cumpla la condición
14 }
15
16 // Imprimir un número 10 veces
17 for(int x = 0; x <= 10; x++) {
18     System.out.println(x);
19 }

```

## Tipos de datos

Primitivos	<i>char</i>	<code>char caracter = 'a';</code>
	<i>int</i>	<code>int entero = 10;</code>
	<i>float</i>	<code>float decimal = 12.5;</code>
	<i>boolean</i>	<code>Boolean bool = true;</code>
Compuestos	<i>String</i>	<code>String cadena = "Hola mundo!";</code>
	<i>Array</i>	<code>tipo_dato nombre_array[] = new tipo_dato[tamaño];</code>
	<i>Matriz</i>	<code>tipo_dato nombre_matriz[][] = new tipo_dato[tamaño][tamaño];</code>

# Operadores

## Operadores matemáticos

Operador	Operación
+	Suma
-	Resta
*	Multiplicación
/	División
%	Módulo (resto)

## Ejemplo

```

1 class Main {
2     public static void main(String[] args) {
3
4         // Variables
5         int a = 12, b = 5;
6
7         // Operador suma
8         System.out.println("a + b = " + (a + b));
9
10        // Operador resta
11        System.out.println("a - b = " + (a - b));
12
13        // Operador multiplicacion
14        System.out.println("a * b = " + (a * b));
15
16        // Operador division
17        System.out.println("a / b = " + (a / b));
18
19        // Operador módulo
20        System.out.println("a % b = " + (a % b));
21
22        // Operador exponente
23        System.out.println("a ^ b = " + Math.pow(a, b));
24    }
25 }

```

```

a + b = 17
a - b = 7
a * b = 60
a / b = 2
a % b = 2
a ^ b = 248832.0

```

**OJO!**  $a / b = 2$  porque utilizamos enteros.

En Java:

- $(9 / 2) = 4$
- $(9.0 / 2) = 4.5$
- $(9 / 2.0) = 4.5$
- $(9.0 / 2.0) = 4.5$

## Operadores lógicos

Operador	Ejemplo	Significado
&&	expresion1 && expresion2	<i>Verdadero</i> sólo si las dos expresiones son verdaderas.
	expresion1    expresion2	<i>Verdadero</i> si alguna de las expresiones son verdaderas
!	!expresion	<i>Verdadero</i> si la expresión es <i>falsa</i> y viceversa (cambia el valor de un booleano)

```

1 class Main {
2     public static void main(String[] args) {
3
4         // Operador &&
5         System.out.println((5 > 3) && (8 > 5)); // true
6         System.out.println((5 > 3) && (8 < 5)); // false
7
8         // Operador ||
9         System.out.println((5 < 3) || (8 > 5)); // true
10        System.out.println((5 > 3) || (8 < 5)); // true
11        System.out.println((5 < 3) || (8 < 5)); // false
12
13        // Operador !
14        System.out.println(!(5 == 3)); // true
15        System.out.println(!(5 > 3)); // false
16    }
17 }
```

## Operadores relacionales

Operador	Descripción	Ejemplo
<code>==</code>	Igual que	<code>3 == 5</code> devuelve <i>falso</i>
<code>!=</code>	No es igual que	<code>3 != 5</code> devuelve <i>verdadero</i>
<code>&gt;</code>	Mayor que	<code>3 &gt; 5</code> devuelve <i>falso</i>
<code>&lt;</code>	Menor que	<code>3 &lt; 5</code> devuelve <i>verdadero</i>
<code>&gt;=</code>	Mayor o igual	<code>3 &gt;= 5</code> devuelve <i>false</i>
<code>&lt;=</code>	Menor o igual	<code>3 &lt;= 5</code> devuelve <i>verdadero</i>

## Operadores de asignación

Operador	Ejemplo	Equivale a
<code>=</code>	<code>a = b;</code>	<code>a = b;</code>
<code>+=</code>	<code>a += b;</code>	<code>a = a + b;</code>
<code>-=</code>	<code>a -= b;</code>	<code>a = a - b;</code>
<code>*=</code>	<code>a *= b;</code>	<code>a = a * b;</code>
<code>/=</code>	<code>a /= b;</code>	<code>a = a / b;</code>
<code>%=</code>	<code>a %= b;</code>	<code>a = a % b;</code>