

Apunte – Actividad 1

Operadores Relacionales, Lógicos y Estructuras Condicionales

1. Operadores Relacionales

Los operadores relacionales se utilizan para **comparar valores** y siempre devuelven un resultado booleano (true o false).

Operador	Descripción	Ejemplo	Resultado
==	Igual a	5 == 5	true
!=	Distinto de	5 != 3	true
>	Mayor que	7 > 2	true
<	Menor que	4 < 6	true
>=	Mayor o igual que	5 >= 5	true
<=	Menor o igual que	3 <= 8	true

Importante: Al comparar objetos como String, **NO se usa ==**, sino el método .equals().

```
String nombre1 = "Juan";  
String nombre2 = "Juan";  
  
System.out.println(nombre1 == nombre2); // Puede ser true o false (compara referencias)  
System.out.println(nombre1.equals(nombre2)); // Siempre true (compara contenido)
```

2. Operadores Lógicos

Se utilizan para **combinar condiciones booleanas**.

Operador	Descripción	Ejemplo	Resultado
&&	AND lógico (y)	(5 > 2) && (4 < 6)	true
`		`	OR lógico (o)
!	Negación (no)	!(5 > 3)	false

Ejemplo práctico:

```
int edad = 20;
boolean tieneEntrada = true;

if (edad >= 18 && tieneEntrada) {
    System.out.println("Puede ingresar al evento.");
} else {
    System.out.println("No puede ingresar.");
}
```

3. Estructuras Condicionales

3.1. *if*

Permite ejecutar un bloque de código si una condición se cumple.

```
int numero = 10;

if (numero > 0) {
    System.out.println("El número es positivo.");
}
```

3.2. *if - else*

Ejecuta un bloque u otro según la condición.

```
if (numero % 2 == 0) {
    System.out.println("El número es par.");
} else {
    System.out.println("El número es impar.");
}
```

3.3. *if - else if - else*

Permite múltiples condiciones.

```
int nota = 75;

if (nota >= 90) {
    System.out.println("Excelente");
} else if (nota >= 60) {
    System.out.println("Aprobado");
} else {
    System.out.println("Desaprobado");
}
```

4. switch

Se utiliza cuando hay varias opciones posibles para una misma variable.

```
int dia = 3;

switch (dia) {
    case 1:
        System.out.println("Lunes");
        break;
    case 2:
        System.out.println("Martes");
        break;
    case 3:
        System.out.println("Miércoles");
        break;
    default:
        System.out.println("Día no válido");
}
```

- El break evita que se ejecuten los demás casos después de encontrar el correcto.
- El default es opcional y se ejecuta si no coincide ningún caso.

5. Operador Ternario

Es una forma resumida de escribir un if - else.

Sintaxis:

condición ? valor_si_true : valor_si_false;

Ejemplo:

```
int numero = 8;
String resultado = (numero % 2 == 0) ? "Par" : "Impar";
System.out.println(resultado);
```

6. Buenas Prácticas

- Usar **sangría y llaves {}** para mejorar la legibilidad.
- Recordar que == compara **referencias** en objetos, usar .equals() para comparar contenido.
- En switch, incluir siempre break salvo que se desee ejecutar varios casos seguidos.
- Usar comentarios (//) para explicar condiciones complejas.