

PROMETEO

Estructuras de Control

Índice

01

Expresiones Lógicas

Operadores Relacionales

Operadores Lógicos

02

Condicional Simple 'if'

Sintaxis

Ejemplo

03

Condicional Doble 'if-else'

Sintaxis

Ejemplo

Anidación de Condicionales 'if-else if'

04

Condicional Múltiple 'switch'

Sintaxis clásica

Sintaxis nueva



1

Expresiones Lógicas

PROMETEO

01

Introducción

- En el flujo de un programa utilizaremos con frecuencia **expresiones lógicas** (condicionales) para determinar cuál será el siguiente bloque de instrucciones a ejecutar.
- La **evaluación** de una expresión lógica siempre ofrece como resultado un valor booleano (**true/false**)
- Las expresiones lógicas se pueden componer con dos tipos de **operadores: relacionales y lógicos**

01

Operadores Relacionales

Los operadores relacionales son símbolos utilizados para **comparar dos valores y determinar** si una relación entre ellos es **verdadera o falsa**.

Los valores a comparar pueden ser: variables, constantes o expresiones aritméticas

Símbolo	Descripción
<code>==</code>	Igual que
<code>!=</code>	Distinto que
<code><</code>	Menor que
<code><=</code>	Menor o igual que
<code>></code>	Mayor que
<code>>=</code>	Mayor o igual que

Los operadores relacionales **siempre** devuelven **true o false**

01

Operadores Lógicos

Los operadores lógicos son símbolos que **combinan** expresiones booleanas (verdadero o falso) para producir un único **resultado booleano**.

Dichas expresiones booleanas pueden ser el resultado de una expresión relacional, variables booleanas o el resultado de la ejecución de métodos que retornen un valor booleano.

Símbolo	Descripción
&&	Operador Y (AND)
	Operador O (OR)
!	Operador negación (NOT)

Los operadores lógicos permiten construir expresiones lógicas más complejas

01

Operadores Lógicos

Una **tabla de verdad** es una tabla que muestra todas las posibles combinaciones de valores de verdad (verdadero o falso) para una proposición compuesta, así como el resultado de cada una de dichas combinaciones.

a	b	a b	a && b	!a
F	F	F	F	T
T	F	T	F	F
F	T	T	F	T
T	T	T	T	F

Tabla de verdad para OR, AND, y NOT



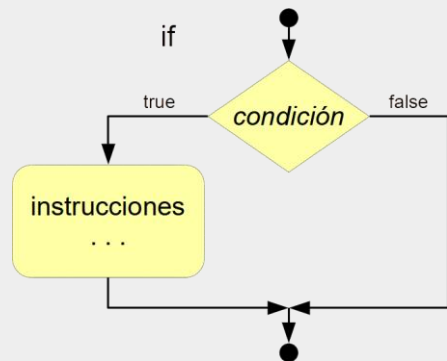
2

Condicional
simple 'if'

02

if

Solo si la evaluación de la condición es *true*, se ejecutará el bloque de instrucciones (de cualquier tipo) que contiene el *if*.



```
if (condición) {  
    bloque de instrucciones;  
}
```

```
import java.util.Scanner;  
  
//Determina si el número introducido por el usuario es positivo  
public class EjemploCondicionales {  
  
    public static void main(String[] args) {  
        Scanner entrada = new Scanner(System.in);  
        int numero;  
  
        System.out.println("Introduzca un número: ");  
        numero = entrada.nextInt();  
  
        if (numero >= 0){ //si la condición es verdadera, se imprime el mensaje  
            System.out.println("El número es positivo");  
        }  
  
        System.out.println("Final del proceso");  
        entrada.close();  
    }  
}
```



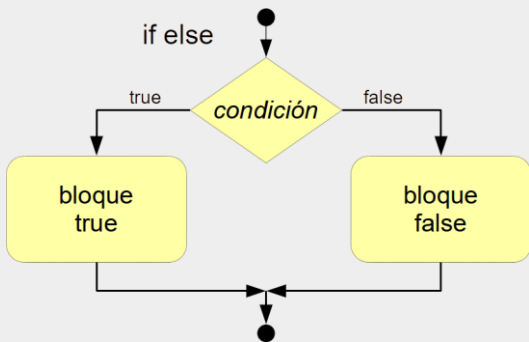
3

Condicional
doble 'if-else'

03

if – else

Si la evaluación de la condición es *true*, se ejecutará el bloque *if*.
En caso contrario, se ejecutará el bloque *else*.



```
if (condición) {  
    bloque if;  
} else {  
    bloque else;  
}
```

```
import java.util.Scanner;  
  
//Determina si el número es par o impar  
public class EjemploCondicionales {  
  
    public static void main(String[] args) {  
        Scanner entrada = new Scanner(System.in);  
        int numero;  
  
        System.out.println("Introduzca un número: ");  
        numero = entrada.nextInt();  
  
        if (numero % 2 == 0){ //si el número es par  
            System.out.println("El número es par");  
        }  
        else {  
            System.out.println("El número es impar");  
        }  
  
        System.out.println("Final del proceso");  
        entrada.close();  
    }  
}
```

03

if - else

Es posible **anidar** condicionales tanto el en bloque *if* como en el *else*.

Nota : Cuando el bloque *if* o *else* tienen una sola instrucción, se pueden omitir las llaves {}

```
if (n1 > n2 && n2 > n3)
    System.out.println(n1+" - "+n2+" - "+n3);
else
    if(n1 > n3 && n3 > n2)
        System.out.println(n1+" - "+n3+" - "+n2);
    else
        if(n2 > n1 && n1 > n3)
            System.out.println(n2+" - "+n1+" - "+n3);
        else
            if(n2 > n3 && n3 > n1)
                System.out.println(n2+" - "+n3+" - "+n1);
            else
                if(n3 > n1 && n1 > n2)
                    System.out.println(n3+" - "+n1+" - "+n2);
                else //solo queda n3>n2 && n2>n1
                    System.out.println(n3+" - "+n2+" - "+n1);
```


```
import java.util.Scanner;

//Ordena de mayor a menor
public class EjemploCondicionales {

    public static void main(String[] args) {
        Scanner entrada = new Scanner(System.in);
        int n1, n2, n3;

        System.out.println("Introduzca el primer número: ");
        n1 = entrada.nextInt();
        System.out.println("Introduzca el segundo número: ");
        n2 = entrada.nextInt();
        System.out.println("Introduzca el tercer número: ");
        n3 = entrada.nextInt();

        if (n1 > n2 && n2 > n3) {
            System.out.println(n1+" - "+n2+" - "+n3);
        }
        else {
            if(n1 > n3 && n3 > n2) {
                System.out.println(n1+" - "+n3+" - "+n2);
            }
            else {
                if(n2 > n1 && n1 > n3) {
                    System.out.println(n2+" - "+n1+" - "+n3);
                }
                else {
                    if(n2 > n3 && n3 > n1) {
                        System.out.println(n2+" - "+n3+" - "+n1);
                    }
                    else {
                        if(n3 > n1 && n1 > n2) {
                            System.out.println(n3+" - "+n1+" - "+n2);
                        }
                        else { //solo queda n3>n2 && n2>n1
                            System.out.println(n3+" - "+n2+" - "+n1);
                        }
                    }
                }
            }
        }
    }
}
```



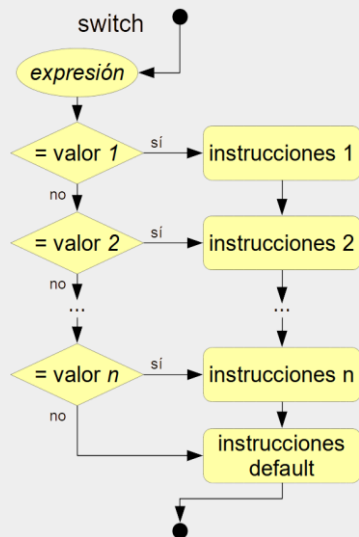
4 Condicional múltiple 'switch'

PROMETEO

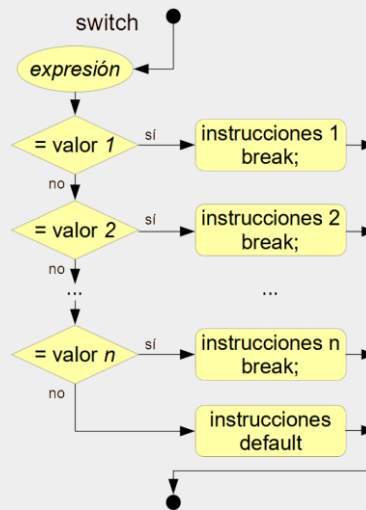
04

switch

Utilizaremos *switch* cuando se necesite comparar el resultado de una expresión con múltiples valores, de un conjunto finito y conocido.



Se puede añadir el uso del **break** para interrumpir la ejecución de los bloques del switch según se desee.



04

switch

Ejemplo

```
import java.util.Scanner;

public class EjSwitch {

    public static void main (String [] args){

        Scanner teclado = new Scanner(System.in);

        System.out.println("Introduce las siglas del ciclo");
        String siglas = teclado.next();

        switch (siglas){
            case "DAM":
                System.out.println("Desarrollo Aplicaciones Multiplataforma");
                break;
            case "DAW":
                System.out.println("Desarrollo Aplicaciones Web");
                break;
            case "ASIR":
                System.out.println("Administración Sistemas Informáticos en Red");
                break;
            default:
                System.out.println("Ciclo inexistente");
        }
        teclado.close();
    }
}
```

1. **Evalúa** expresión y obtiene su valor.
2. **Compara**, uno a uno, el valor obtenido con cada valor de las cláusulas case.
3. En el momento en que **coincide** con alguno de ellos, ejecuta el conjunto de instrucciones de **esa** cláusula case y de todas las siguientes, salvo que se añada un 'break'.
4. Si no existe coincidencia alguna, se ejecuta el conjunto de instrucciones de la cláusula **default**, siempre y cuando esta esté presente.

04

switch

- En las últimas versiones de java se ha incluido una **nueva forma de usar el switch**:
- No es necesario utilizar 'break' porque solo se ejecuta el bloque de instrucciones del valor que coincida con la expresión evaluada.
- Se sustituyen los : del case por ->
- Se utiliza como una **expresión**, es decir:
 - Se asigna el resultado a una variable con 'yield'
 - Finaliza con ;

```
import java.util.Scanner;

public class EjemploSwitchNuevo {
    public static void main (String [] args){

        Scanner teclado = new Scanner(System.in);

        System.out.println("Introduce las siglas del ciclo");
        String siglas = teclado.next();

        String curso = switch (siglas){
            case "DAM"-> {yield "Desarrollo Aplicaciones Multiplataforma";}
            case "DAW" -> {yield "Desarrollo Aplicaciones Web";}
            case "ASIR"-> {yield "Administración Sistemas Informáticos en Red";}
            default -> {yield "Ciclo inexistente";}
        };

        System.out.println(curso);
        teclado.close();
    }
}
```



PROMETEO