

Práctica #4 – Java: Lógica condicional parte 2



RESULTADOS DE APRENDIZAJE:

Al finalizar esta sesión de práctica el estudiante deberá:

- ✓ Crear código que permita la captura de datos.
- ✓ Realizar operaciones de asignación de datos.
- ✓ Comprender el uso de la clase MATH de Java.
- ✓ Realizar operaciones matemáticas en Java.

OPERADORES DE IGUALDAD Y COMPARACIÓN

Operador	Significado	Expresión
>	mayor que	a > b
<	menor que	a < b
>=	mayor que o igual a	a >= b
<=	menor que o igual a	a <= b
==	es igual a	a == b
!=	no igual a	a != b

OPERADORES LÓGICOS

Operador	Descripción	Expresión	Función correspondiente
	true si cualquiera de las expresiones booleanas es true	(a>b) (a<c)	(a>b) o (a<c)
&&	true si todas las expresiones booleanas are true	(a>b) && (a<c)	(a>b) y (a<c)

IF ANIDADOS EN JAVA

Es posible anidar ifs para reorganizar las condiciones o hacerlas más sencillas lo que en muchas ocasiones hace el código más fácil de leer, a continuación, una de las posibles formas posibles para reescribir el último ejemplo de la guía anterior.

ACTIVIDAD 1

1. Crea un proyecto llamado Condicionales4 y escribe el siguiente código

```
package condicionales4;

public class Condicionales4 {

    public static void main(String[] args) {
        if (temperatura > 15) {
            if (temperatura > 25) {
                // Si la temperatura es mayor que 25 ...
                System.out.println("A la playa!!!");
            } else {
                System.out.println("A la montaña!!!");
            }
        } else if (temperatura < 5) {
            if (nevando) {
                System.out.println("A esquiar!!!");
            }
        } else {
            System.out.println("A descansar... zZz");
        }
    }
}
```

2. Modifica los valores de las variables temperatura y nevando para obtener diferentes resultados y toma nota de ellos.

EL OPERADOR TERNARIO ?: EN JAVA

En ocasiones un **if** se utiliza simplemente para obtener un valor u otro en función de si una se cumple una condición o no, en estos casos se puede sustituir la estructura **if else** por el operador ternario u operador condicional que tiene la sintaxis **resultado = condicion ? valor_si_se_cumple :**

valor_si_no_se_cumple los valores tanto para el caso de que se cumpla como para el que no pueden ser valores directamente como un texto o un entero o llamadas a funciones que devuelvan un objeto del tipo esperado.

ACTIVIDAD 2

1. Crea un proyecto llamado Condicionales5 y escribe el siguiente código

```
package condicionales5;

public class Condicionales5 {
    // Ejemplo 1
    String queHacer = (temperatura > 25) ? "A la playa!!!" : "Esperando el
    buen tiempo...";

    // Ejemplo 2
    double numero = (temperatura > 10) ? (Math.random() * 10) : 0;

    // Ejemplo 3
    Comida miComida = cansado ? pedirComida("china") : hacerComida();
}
```

2. Modifica los valores de las variables temperatura para obtener diferentes resultados y toma nota de ellos.

SWITCH EN JAVA

Con el **switch** se puede hacer un control del tipo **if else if...** más estructurado, pero en realidad no exactamente igual puesto que con el **switch** lo que se hace es definir un conjunto de casos que van a tener una ejecución distinta y se ejecutará el caso que coincida con el valor indicado en el **switch**.

```
switch (variable) {  
    case valor_1:  
    case valor_2:  
        // Ejecutar si el valor de variable es igual a valor_1 o a valor_2  
        ejecutarA();  
        break; // Salir del switch  
    case valor_3:  
        // Ejecutar si el valor de variable es igual a valor_3  
        ejecutarB();  
        break; // Salir del switch  
    default:  
        // Ejecutar si el valor de variable es distinto del resto de casos  
        ejecutarC();  
        break; // Salir del switch  
}
```

Por lo tanto el **switch** es la mejor opción cuando tenemos un conjunto de valores esperados (**if** con **==** o **.equals()**) porque es más legible y más limpio, imagina por ejemplo sacar el nombre del día de la semana a partir de su número, con un **if else** necesitamos 7 **elses...** pero es completamente inservible para condiciones en las que se hacen comparaciones del tipo **<** o **>** por que controlar todos los posibles valores puede ser una locura, suponiendo que el rango no esté muy acotado que entonces si sería una opción válida.

ACTIVIDAD 3

Para los siguientes ejemplos crea proyectos independientes y toma nota de los resultados.

```
// Switch con int como condicion  
Integer dia = Calendar.getInstance().get(Calendar.DAY_OF_WEEK);  
switch (dia) {  
    case 1:  
        System.out.println("Domingo");  
        break;  
    case 2:  
        System.out.println("Lunes");  
        break;  
    case 3:  
        System.out.println("Martes");  
        break;  
    case 4:  
        System.out.println("Miercoles");  
        break;  
    case 5:  
        System.out.println("Jueves");  
        break;  
    case 6:  
        System.out.println("Viernes");  
}
```

```
        break;
    case 7:
        System.out.println("Sabado");
        break;
    // default: es opcional
}
```

```
String tipoVehiculo = "coche";
switch (tipoVehiculo) {
    case "coche":
        System.out.println("Puedes pasar de 00:00 a 08:00");
        break;
    case "camion":
        System.out.println("Puedes pasar de 08:00 a 16:00");
        break;
    case "moto":
        System.out.println("Puedes pasar de 16:00 a 24:00");
        break;
    default:
        System.out.println("No se puede pasar con un " + tipoVehiculo);
        break;
}
```

De los dos ejemplos anteriores podemos sacar que el número de casos es variable según las necesidades y que el caso por defecto (el que se ejecuta cuando no se cumple otro igual que el **else** de un **if**) es opcional por lo que lo podemos omitir si no se requiere ninguna acción especial para ese caso o si simplemente sabemos que no existe otra opción como sucede con los días de la semana.

Y si se quiere que en varios casos se ejecute la misma lógica no es necesario repetirla una y otra vez para cada caso, en su lugar podemos agrupar los casos y eliminar los **break** intermedios que hacen que se salga del **switch**. Por ejemplo para realizar una acción para los días laborables y otra distinta para los festivos se hace del siguiente modo.

```
switch (dia) {
    case 2:
    case 3:
    case 4:
    case 5:
    case 6:
        System.out.println("Dia laboral");
        break;
    case 1:
    case 7:
        System.out.println("Fin de semana");
        break;
    default:
        System.out.println("La semana solo tiene 7 dias");
        break;
}
```

Por ejemplo si fuese martes entraría por el 3, sigue por el 4,5 y 6 imprime el texto "Dia laboral" y termina, es decir cuando entra en el **switch** ejecuta todo el código que encuentre hasta que se termine o hasta que se encuentre un **break** por lo que podemos aprovechar esto para casos en los que los casos no ejecutan lógica excluyente si no que lo que hacen es añadir más funcionalidad. Un ejemplo muy simple para ver esto es sacar el número de días restantes hasta el fin de semana partiendo del ejemplo anterior.

```
Integer diasHastaFinSemana = 0;
switch (dia) {
    case 2:
        diasHastaFinSemana++;
    case 3:
        diasHastaFinSemana++;
    case 4:
        diasHastaFinSemana++;
    case 5:
        diasHastaFinSemana++;
    case 6:
        System.out.println("Dia laboral: Dias restantes hasta el fin de
semana: " + diasHastaFinSemana);
        break;
    case 1:
    case 7:
        System.out.println("Fin de semana");
        break;
    default:
        System.out.println("La semana solo tiene 7 dias");
        break;
}
```

Y para terminar comentar que cualquier **switch** se puede hacer usando **ifs else** porque al fin y al cabo son la base sobre la que se sustenta su funcionamiento. Podemos verlo con la conversión del último ejemplo que es el más complejo de los **switch** que hemos visto.

```
Integer diasHastaFinSemana = 0;
if (dia > 1 && dia < 7) {
    if (dia < 6) {
        diasHastaFinSemana++;
        if (dia < 5) {
            diasHastaFinSemana++;
            if (dia < 4) {
                diasHastaFinSemana++;
                if (dia < 3) {
                    diasHastaFinSemana++;
                }
            }
        }
    }
}
System.out.println("Dia laboral: Dias restantes hasta el fin de semana: "
+ diasHastaFinSemana);
} else if (dia == 1 || dia == 7) {
    System.out.println("Fin de semana");
} else {
    System.out.println("La semana solo tiene 7 dias");
}
```

O de la siguiente forma que refleja de un modo más claro las condiciones que se evalúan en el **switch** y que sería igualmente valido si la variable fuese un string en lugar de un número.

```
if (dia == 2) {
    diasHastaFinSemana++;
}

if (dia == 2 || dia == 3) {
    diasHastaFinSemana++;
}

if (dia == 2 || dia == 3 || dia == 4) {
    diasHastaFinSemana++;
}

if (dia == 2 || dia == 3 || dia == 4 || dia == 5) {
    diasHastaFinSemana++;
}

if (dia == 2 || dia == 3 || dia == 4 || dia == 5 || dia == 6) {
    System.out.println("Dia laboral: Dias restantes hasta el fin de semana: "
+ diasHastaFinSemana);
}

if (dia == 1) {
    // No hace nada
}

if (dia == 1 || dia == 7) {
    System.out.println("Fin de semana");
}

if (dia != 1 && dia != 2 && dia != 3 && dia != 4 && dia != 5 && dia != 6 &&
dia != 7) {
    System.out.println("La semana solo tiene 7 dias");
}
```