

Práctica #3 – Java: Lógica condicional parte 1



RESULTADOS DE APRENDIZAJE:

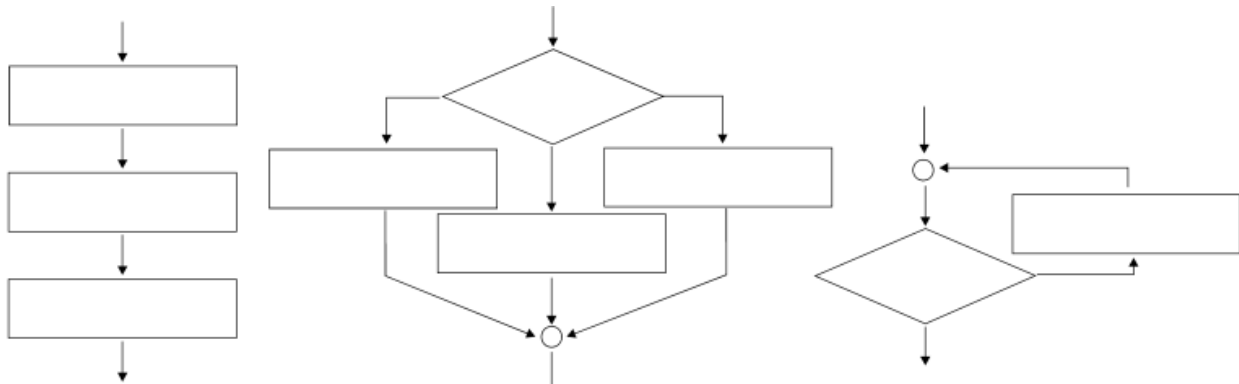
Al finalizar esta sesión de práctica el estudiante deberá:

- ✓ 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.

CONCEPTOS BÁSICOS

Para realizar un programa estructurado este deberá contar con las siguientes características:

1. Secuencial: Ejecuta una sentencia detrás de otra.
2. Condicional: Se evalúa una expresión y, dependiendo del resultado, se decide la siguiente sentencia a ejecutar.
3. Iterativa: Repetimos un bloque de sentencias hasta que sea verdadera una determinada condición.



ESTRUCTURAS DE CONTROL

Muchas veces cuando se programa una aplicación o un programa complejo es necesario mantener el control sobre el flujo del programa, haciendo suposiciones sobre eventos que podrían ocurrir de modo que podamos realizar y establecer una previa respuesta, en la programación existe un recurso que podemos usar para tener más control sobre el flujo del programa y su forma de ejecución.

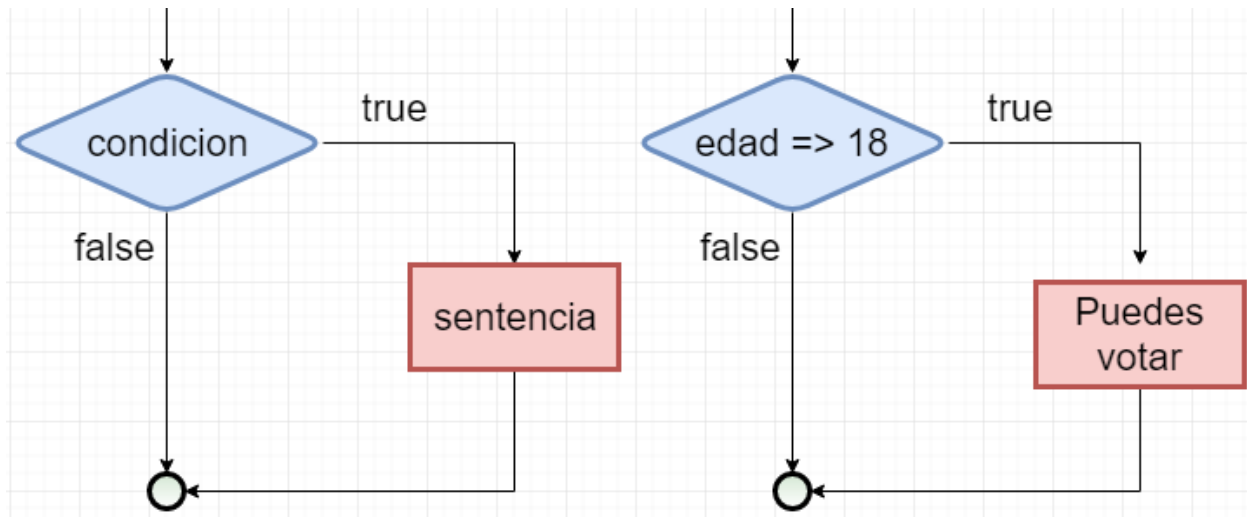
A estos se les conoce como **Estructuras de control** las cuales se dividen en:

1. **Estructuras Condicionales**
2. **Estructuras Iterativas o repetitivas**

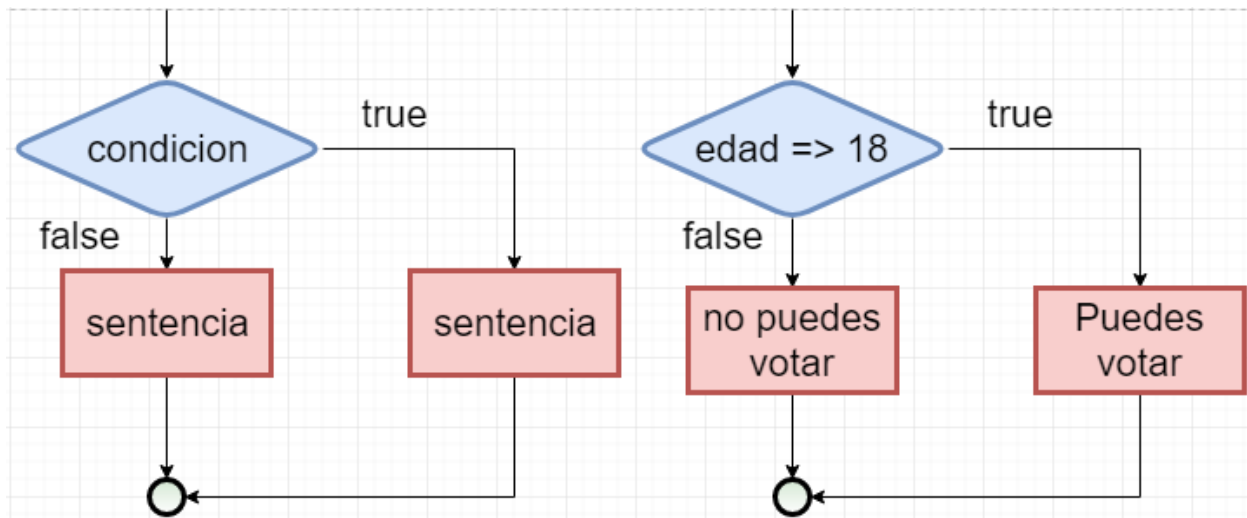
ESTRUCTURAS CONDICIONALES

En programación, una sentencia condicional es una instrucción o grupo de instrucciones que se pueden ejecutar o no en función del valor de una condición.

Alternativa Simple:



Alternativa doble:



Los tipos más conocidos de sentencias condicionales son el SI..ENTONCES (if..then), el SI..ENTONCES..SI NO (if..then..else) y el SEGÚN (case o switch), aunque también podríamos mencionar al manejo de excepciones como una alternativa más moderna para evitar el "anidamiento" de sentencias condicionales.

Las sentencias condicionales constituyen, junto con los bucles, los pilares de la programación estructurada, y su uso es una evolución de una sentencia en lenguaje ensamblador que ejecutaba la siguiente línea o no en función del valor de una condición.

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 EN JAVA

La estructura condicional más simple en **Java** es el **if**, se evalúa una condición y en caso de que se cumpla se ejecuta el contenido entre las llaves **{}** o en caso de que se omitan se ejecuta el código hasta el primer **“;”** por lo tanto si no se usan los **{}** la condición aplica solo a la siguiente instrucción al **if**.

```

if (condicion) {
    ejecutar();
}

// Los {} son opcionales
if (condicion)
    ejecutar();

// Y mas compacto
if (condicion) ejecutar();

```

Para que se ejecute el código dentro de un **if** se debe de cumplir la condición que puede ser directamente un boolean o una comprobación que genere uno (**==, <, <=, >, >=, !=, .equals()**) o cualquier otra función que devuelva **true** o **false** pudiendo ser la condición todo lo complejo que queramos.

ACTIVIDAD 1

1. Crea un proyecto llamado Condicionales1 y escribe el siguiente código

```
package condicionales1;

public class Condicionales1 {

    public static void main(String[] args) {

        int temperatura = 30;
        boolean haceSol = true;
        boolean nevando = false;

        // Si la temperatura es mayor que 25
        if (temperatura > 25) {
            System.out.println("A la playa!!!");
        }

        // Si haceSol es verdadero
        if (haceSol) {
            System.out.println("No te olvides la sombrilla");
        }

        // Si esta nevando o hace sol
        if (nevando || haceSol) {
            System.out.println("Que bien");
        }

        // Si nevando y la temperatura esta entre 20 y 30
        if (nevando && (temperatura >= 20 && temperatura <= 30)) {
            System.out.println("No me lo creo");
        }

        // Si la temperatura es menor que 0 o mayor que 30 y hace sol
        if ((temperatura < 0 || temperatura > 30) && haceSol) {
            System.out.println("Mejor me quedo en casa");
        }
    }
}
```

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

IF ELSE EN JAVA

Con el **if** solo podemos hacer que se ejecute un fragmento de código o no, pero en el caso de que no se cumpla la condición no se hace nada (sigue el flujo normal de ejecución) por lo que si queremos que se ejecute otra cosa cuando no se cumpla la condición solo con el **if** tenemos que hacer otro con la condición inversa provocando que se tenga que comprobar la condición 2 veces mientras que si usamos el **else** solo necesitamos hacer la comprobación una sola vez.

ACTIVIDAD 2

1. Crea un proyecto llamado Condicionales2 y escribe el siguiente código

```
package condicionales2;

public class Condicionales2 {

    public static void main(String[] args) {

        int temperatura = 30;

        // Usando solo ifs
        // Si la temperatura es mayor que 25
        if (temperatura > 25) {
            System.out.println("A la playa!!!");
        }

        if (temperatura <= 25) {
            System.out.println("Esperando al buen tiempo...");
        }

        // Usando if con else
        // Si la temperatura es mayor que 25 ... y si no ...
        if (temperatura > 25) {
            System.out.println("A la playa!!!");
        } else {
            System.out.println("Esperando al buen tiempo...");
        }
    }
}
```

2. Modifica el valor de la variable temperatura, para obtener diferentes resultados y toma nota de ellos.

IF ELSE IF EN JAVA

El siguiente nivel es hacer el **else** condicional añadiendo un **if** para que si no se cumple la primera condición (la del **if** principal) se evalúe esta nueva condición de modo que se puede concatenar la cantidad de **if else** que se necesiten para cubrir todos los distintos escenarios que precisen ser tratados de una forma particular, siendo el último **else** el que se ejecute cuando no se cumpla ninguna condición (el **else** es siempre opcional).

ACTIVIDAD 3

1. Crea un proyecto llamado Condicionales3 y escribe el siguiente código

```
package condicionales3;

public class Condicionales3 {

    public static void main(String[] args) {
        int temperatura = 30;
        boolean nevando = false;

        if (temperatura > 25) {
            // Si la temperatura es mayor que 25 ...
            System.out.println("A la playa!!!");
        } else if (temperatura > 15) {
            // si es mayor que 15 y no es mayor que 25 ...
            System.out.println("A la montaña!!!");
        } else if (temperatura < 5 && nevando) {
            // si es menor que 5 y está nevando y no es mayor que 15 ni mayor que 25
            System.out.println("A esquiar!!!");
        } else {
            // si la temperatura no es mayor que 25 ni que 15 ni menor que 5 si está
            // nevando
            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.

ACTIVIDAD 4

1. Utilizando variables condicionales **if-else-if** escribe un programa que solicite la edad al usuario, si esta es menor de 18 años que imprima en pantalla un mensaje con el texto “Eres demasiado joven para votar”. Si la edad es igual o superior de 100 años que imprima en pantalla un mensaje con el texto “Eres demasiado viejo para votar”. En caso contrario deberá imprimir en pantalla un mensaje con el texto “Tu puedes votar”.
2. Realice un programa que compare dos números introducidos por el usuario. Deberá imprimir en pantalla dos mensajes, indicando cual número es el mayor y cuál es el menor. En caso de que ambos números sean iguales deberá imprimir un mensaje indicando dicha igualdad.