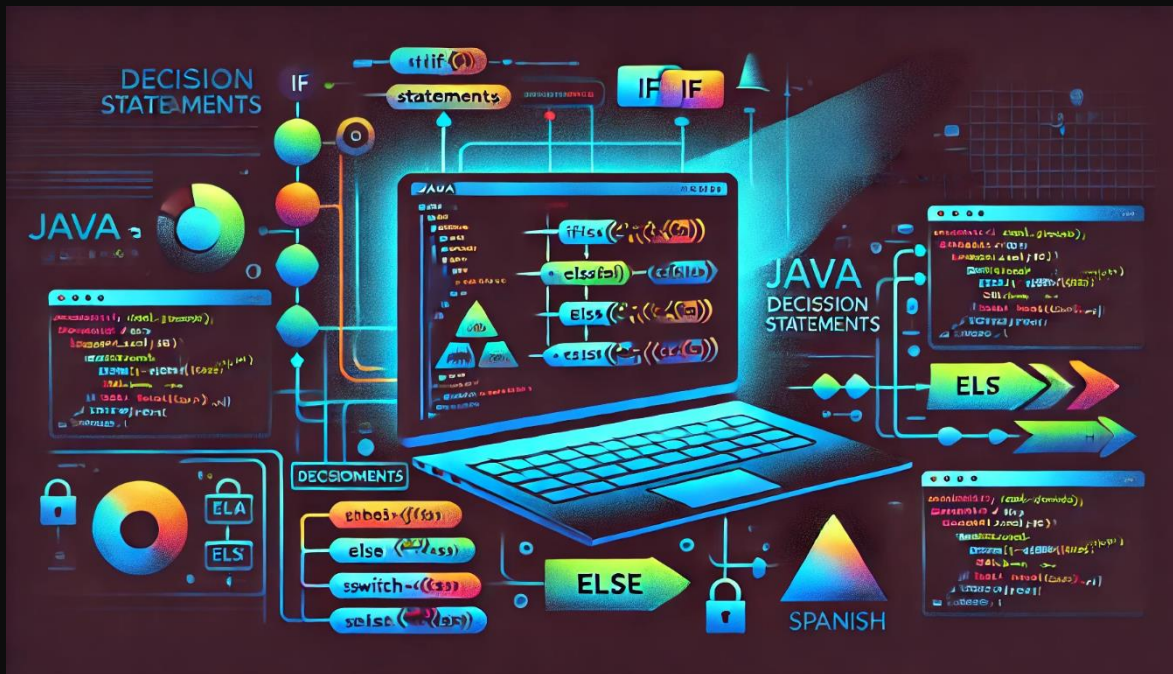


# Sentencias de Decisión en Java



## Sentencias de Decisión en Java

Las sentencias de decisión en Java permiten a los programas tomar diferentes acciones basadas en condiciones específicas. Las sentencias de decisión más comunes en Java son `if`, `if-else`, `if-else if-else` y `switch`.

### Sentencia `if`

La sentencia `if` se utiliza para ejecutar un bloque de código solo si una condición específica es verdadera.

#### Sintaxis:

```
if (condicion) {
    // Bloque de código a ejecutar si la condición es verdadera
}
```

#### Ejemplo:

```
public class EjemploIf {
    public static void main(String[] args) {
        int edad = 20;

        if (edad >= 18) {
            System.out.println("Eres mayor de edad.");
        }
    }
}
```

```
}  
}
```

## Sentencia if-else

La sentencia if-else se utiliza para ejecutar un bloque de código si la condición es verdadera y otro bloque de código si la condición es falsa.

### Sintaxis:

```
if (condicion) {  
    // Bloque de código a ejecutar si la condición es verdadera  
} else {  
    // Bloque de código a ejecutar si la condición es falsa  
}
```

### Ejemplo:

```
public class EjemploIfElse {  
    public static void main(String[] args) {  
        int edad = 16;  
  
        if (edad >= 18) {  
            System.out.println("Eres mayor de edad.");  
        } else {  
            System.out.println("Eres menor de edad.");  
        }  
    }  
}
```

## Sentencia if-else if-else

La sentencia if-else if-else se utiliza para evaluar múltiples condiciones en secuencia.

### Sintaxis:

```
if (condicion1) {  
    // Bloque de código a ejecutar si la condición1 es verdadera  
} else if (condicion2) {  
    // Bloque de código a ejecutar si la condición2 es verdadera  
} else {  
    // Bloque de código a ejecutar si ninguna condición anterior es verdadera  
}
```

### Ejemplo:

```
public class EjemploIfElseIfElse {  
    public static void main(String[] args) {  
        int nota = 85;  
  
        if (nota >= 90) {  
            System.out.println("Calificación: A");  
        }  
    }  
}
```

```
    } else if (nota >= 80) {
        System.out.println("Calificación: B");
    } else if (nota >= 70) {
        System.out.println("Calificación: C");
    } else if (nota >= 60) {
        System.out.println("Calificación: D");
    } else {
        System.out.println("Calificación: F");
    }
}
```

## Sentencia switch

La sentencia `switch` se utiliza para seleccionar una de entre muchas opciones de ejecución. Es especialmente útil cuando se desea comparar una variable con múltiples valores.

### Sintaxis:

```
switch (expresion) {
    case valor1:
        // Bloque de código a ejecutar si expresion == valor1
        break;
    case valor2:
        // Bloque de código a ejecutar si expresion == valor2
        break;
    // ...
    default:
        // Bloque de código a ejecutar si ninguno
        // de los casos anteriores es verdadero
}
```

### Ejemplo:

```
public class EjemploSwitch {
    public static void main(String[] args) {
        int diaSemana = 3;

        switch (diaSemana) {
            case 1:
                System.out.println("Lunes");
                break;
            case 2:
                System.out.println("Martes");
                break;
            case 3:
                System.out.println("Miércoles");
                break;
            case 4:
                System.out.println("Jueves");
                break;
            case 5:
                System.out.println("Viernes");
                break;
        }
    }
}
```

```
        case 6:
            System.out.println("Sábado");
            break;
        case 7:
            System.out.println("Domingo");
            break;
        default:
            System.out.println("Día inválido");
    }
}
```

## Detalles Importantes

1. **Uso del bloque {}:**
  - o Aunque no es obligatorio usar {} para un solo enunciado después de if, else if, else, o case, es una buena práctica hacerlo para evitar errores y mejorar la legibilidad.
2. **break en switch:**
  - o La sentencia break se utiliza para salir del bloque switch después de ejecutar un caso. Si se omite, el flujo de ejecución continuará con el siguiente caso, lo cual puede no ser el comportamiento deseado.
3. **Asociatividad y Precedencia:**
  - o Al usar múltiples operadores en una condición, es importante recordar las reglas de asociatividad y precedencia de operadores para asegurar que las condiciones se evalúen correctamente.

## Ejemplos Combinados

```
public class EjemploDecision {
    public static void main(String[] args) {
        int edad = 20;
        int nota = 85;
        int diaSemana = 3;

        // Ejemplo de if-else
        if (edad >= 18) {
            System.out.println("Eres mayor de edad.");
        } else {
            System.out.println("Eres menor de edad.");
        }

        // Ejemplo de if-else if-else
        if (nota >= 90) {
            System.out.println("Calificación: A");
        } else if (nota >= 80) {
            System.out.println("Calificación: B");
        } else if (nota >= 70) {
            System.out.println("Calificación: C");
        } else if (nota >= 60) {
            System.out.println("Calificación: D");
        } else {
            System.out.println("Calificación: E");
        }
    }
}
```

```
        System.out.println("Calificación: F");
    }

    // Ejemplo de switch
    switch (diaSemana) {
        case 1:
            System.out.println("Lunes");
            break;
        case 2:
            System.out.println("Martes");
            break;
        case 3:
            System.out.println("Miércoles");
            break;
        case 4:
            System.out.println("Jueves");
            break;
        case 5:
            System.out.println("Viernes");
            break;
        case 6:
            System.out.println("Sábado");
            break;
        case 7:
            System.out.println("Domingo");
            break;
        default:
            System.out.println("Día inválido");
    }
}
```

La sentencia switch tiene una sintaxis mejorada, la cual estudiaremos a detalle en lecciones posteriores.

## Resumen

Las sentencias de decisión son fundamentales para controlar el flujo de ejecución de un programa en Java. La capacidad de tomar decisiones basadas en condiciones permite a los desarrolladores crear programas dinámicos y adaptables. Al dominar if, if-else, if-else if-else, y switch, estarás bien equipado/a para enfrentar una amplia variedad de desafíos en la programación. ¡Espero que esta lección se te muy útil!

Saludos!

**Ing. Ubaldo Acosta**

Fundador de [GlobalMentoring.com.mx](https://www.globalmentoring.com.mx)