



# Curso de Java Standard



Ing. Octavio Robleto



octavio.robleto@gmail.com



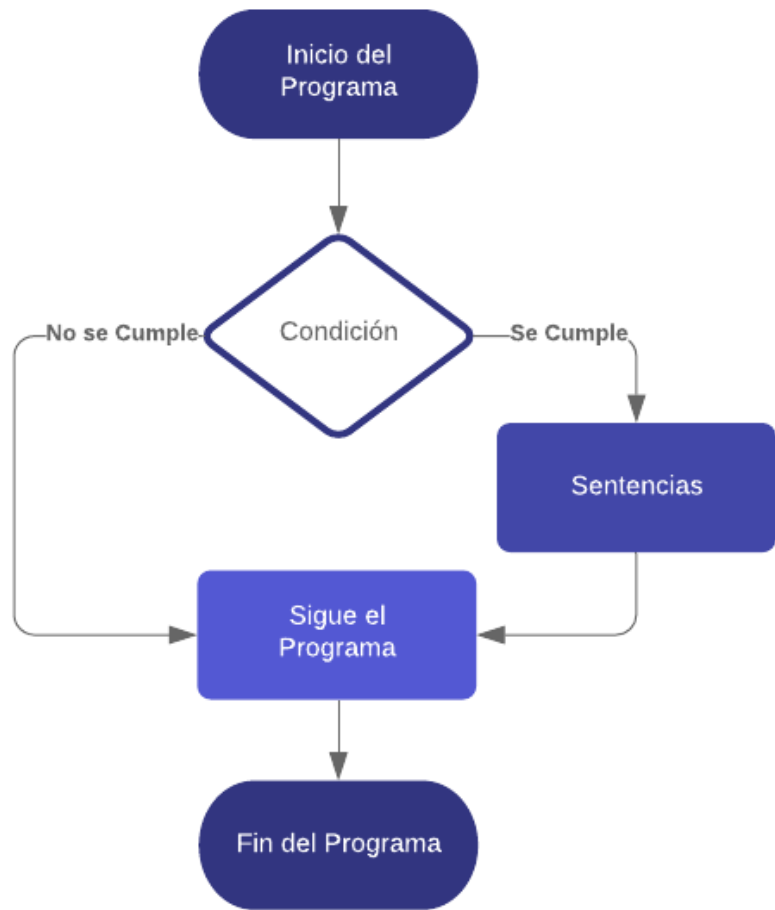
<https://octaviorobleto.com>



# Introducción

En java las instrucciones se ejecutan de forma lineal, en algunos momentos vamos a necesitar ejecutar un código si se cumple o no una pregunta que le hacemos al programa.

La idea es ayudarnos a solucionar un problema considerando los posibles casos a presentarse.



# if

La estructura condicional más simple en la programación es el if y consiste en evaluar una o mas condiciones que deben estar entre paréntesis"()" y en caso que devuelva true se ejecuta lo que se encuentra dentro del bloque.

```
// con llaves
if(condicion){
    sentencia1;
    sentencia2;
}

// sin llaves
if(condicion)
    sentenciaUnica;

// sin llaves, mas compacto
if(condicion) sentenciaUnica;
```

```
// variable a evaluar
byte edad;
edad = 19;

// si la variables es mayor o igual a 18
if (edad >= 18) {
    // mostrar en la consola que es mayor de edad
    System.out.println("Mayor de Edad");
}
```

# if - else

El if solo nos garantiza que si se cumple la condición ejecutara la solución planteada en el bloque, pero que pasa si no se cumple y necesitamos tomar otra solución? Para eso tenemos la estructura if-else que de no cumplirse la condición o condiciones en el if ejecutara lo que se encuentre en el bloque del else

```
// con llaves
if(condicion){
    sentencia1;
    sentencia2;
}else{
    sentencia1;
    sentencia2;
}

// sin llaves
if(condicion)
    sentenciaUnica;
else
    sentenciaUnica;

// sin llaves, mas compacto
if(condicion) sentenciaUnica;
else sentenciaUnica;
```

```
// variable a evaluar
byte edad;
edad = 19;

// si la variables es mayor o igual a 18
if (edad >= 18) {
    // mostrar en la consola que es mayor de edad
    System.out.println("Mayor de Edad");
} else {
    // mostrar en la consola que es menor de edad
    System.out.println("Nino");
}
```

# if anidados

Y si tenemos que evaluar múltiples opciones y para cada una de ellas mostrar una solución distinta? Java nos proporciona el anidamiento de condicionales.

```
// con llaves
if(condicion1){
    sentencia1;
    sentencia2;
} else if(condicionN){
    sentencia1;
    sentencia2;
} else {
    sentencia1;
    sentencia2;
}

// sin llaves
if(condicion1)
    sentenciaUnica;
else if(condicionN)
    sentenciaUnica;
else
    sentenciaUnica;

// sin llaves, mas compacto
if(condicion1) sentenciaUnica;
else if(condicionN) sentenciaUnica;
else sentenciaUnica;
```

```
// variable a evaluar
byte edad;
edad = 19;

// si la variables es mayor o igual a 18
if (edad >= 18) {
    // mostrar en la consola que es mayor de edad
    System.out.println("Mayor de Edad");
} else if (edad >= 12 && edad < 18) { // si la variable se encuentra en un rango
    // mostrar en la consola que es joven
    System.out.println("Joven");
} else {
    // mostrar en la consola que es menor de edad
    System.out.println("Nino");
}
```

# switch

Esta estructura nos ayuda a evaluar múltiples opciones que puede poseer una variable o constante.

Este bloque es un poco distinto ya que comienza con los dos puntos ":" y finaliza con la palabra "**break**", adicionalmente el break le indicara al software que salga del switch y deje de evaluar.

Al entrar en una de las condiciones y no encontrar la sentencia "**break**" se ejecutarán todas las instrucciones de los demás casos, por eso es importante usar la sentencia "**break**", en algunos casos omitimos esta sentencia para simular un operador lógico OR.

```
switch (variable_o_constante) {  
    case ValorPosible1: // if del switch  
        sentencia1;  
        sentencia2;  
        break;  
    case ValorPosibleN: // else if del switch  
        sentencia1;  
        sentencia2;  
        break;  
    default: // el else del switch  
        sentencia1;  
        sentencia2;  
}
```

```
// variable a evaluar  
byte diaSemana;  
diaSemana = 2;  
  
switch (diaSemana) {  
    case 1:  
        System.out.println("Lunes");  
        break;  
    case 2:  
        System.out.println("Martes");  
        break;  
    default: // el else del switch  
        System.out.println("otro dia");  
}
```

# Operador Ternario

Aunque es un operador, se parece mas a una estructura condicional que podemos usar cuando nos encontramos con un if-else, en cada bloque hay una sola sentencia y lo que cambia la sentencia es la misma variable.

```
variable = (condicion) ? valorSiCumple: valorSiNoCumple;
```

```
byte edad;  
boolean mayorEdad;  
  
edad = 19;  
  
// con operador ternario  
mayorEdad = (edad >= 18) ? true : false;  
  
// con un if-else  
if (edad >= 18) {  
    mayorEdad = true;  
} else {  
    mayorEdad = false;  
}
```