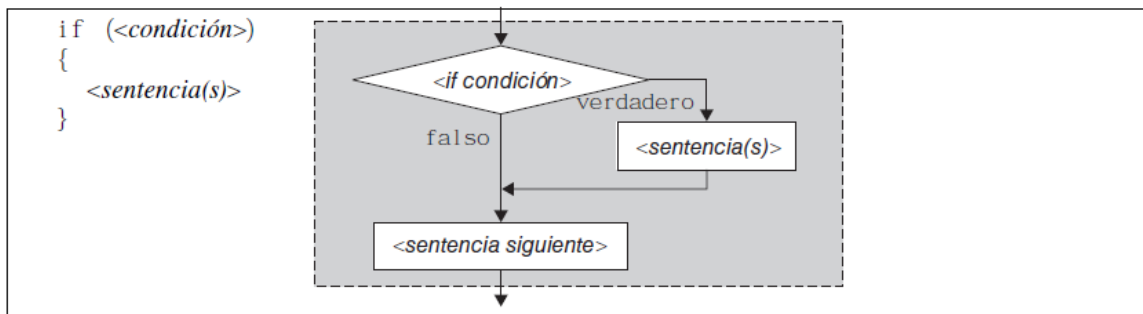


## CONDICION SIMPLE (IF ELSE)

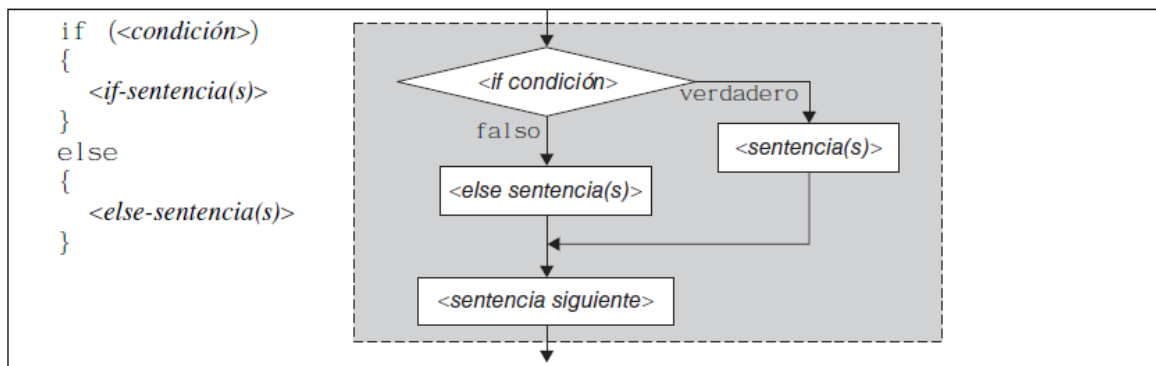
En JAVA la instrucción para realizar una pregunta de un valor (estado) de una variable se utiliza la instrucción if y existen tres formas básicas de una sentencia if:

- **if**: cuando se quiere hacer un proceso.
- **if else**: cuando se quiere hacer uno u otro proceso.
- **if else if**: cuando hay tres o más posibilidades de un proceso.

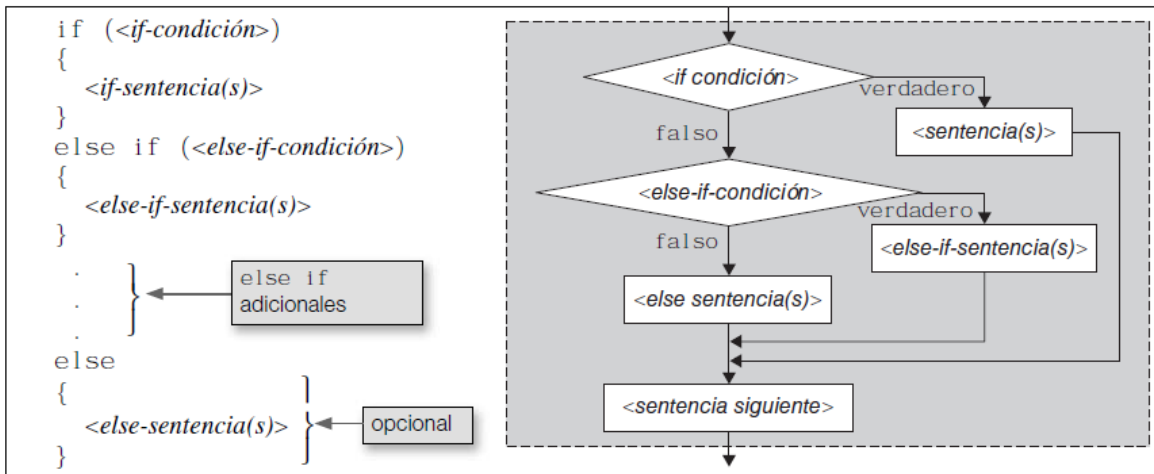
La siguiente figura ilustra la forma básica de una condición (primera opción).



La siguiente ilustra la segunda opción:



Esta ultima la tercera opción:



## OPERADORES RELACIONALES

En Java, para realizar una evaluación de un valor mediante la instrucción if, debemos de utilizar los operadores relacionales que se ilustra en la tabla:

| Operador | Utilización | El resultado es true            |
|----------|-------------|---------------------------------|
| >        | op1 > op2   | si op1 es mayor que op2         |
| >=       | op1 >= op2  | si op1 es mayor o igual que op2 |
| <        | op1 < op2   | si op1 es menor que op2         |
| <=       | op1 <= op2  | si op1 es menor o igual que op2 |
| ==       | op1 == op2  | si op1 y op2 son iguales        |
| !=       | op1 != op2  | si op1 y op2 son diferentes     |

## USO DE LOS OPERADOR

Si declaramos dos variables, podemos comparar los valores entre ellas.

## OPERADOR > (MAYOR QUE)

---

```
// Declaracion de variables
int op1 = 10;
int op2 = 20;

if (op1 > op2) {
    System.out.println (op1 + " es mayor que " + op2 );
} else {
    System.out.println (op2 + " es mayor que " + op1 );
}
```

### Resultado

```
General Output
-----Configu
20 es mayor o igual que 10
Process completed.
```

## OPERADOR >= (MAYOR O IGUAL QUE)

---

```
// Declaracion de variables
int op1 = 30;
int op2 = 20;

if (op1 >= op2) {
    System.out.println (op1 + " es mayor o igual que " + op2 );
} else {
    System.out.println (op2 + " es mayor o igual que " + op1 );
}
```

### Resultado

```
General Output
-----Configurat
30 es mayor o igual que 20
Process completed.
```

## OPERADOR < (MENOR QUE)

---

```
// Declaracion de variables
int op1 = 10;
int op2 = 20;

if (op1 < op2) {
    System.out.println (op1 + " es menor que " + op2 );
} else {
    System.out.println (op2 + " es menor que " + op1 );
}
```

### Resultado

```
General Output
-----
10 es menor que 20
Process completed.
|
```

## OPERADOR <= (MENOR O IGUAL QUE)

---

```
// Declaracion de variables
int op1 = 40;
int op2 = 20;

if (op1 <= op2) {
    System.out.println (op1 + " es menor o igual que " + op2 );
} else {
    System.out.println (op2 + " es menor o igual que " + op1 );
}
```

### Resultado

```
General Output
-----Configurati
20 es menor o igual que 40
Process completed.
```

## OPERADOR == ( ES IGUAL A)

```
// Declaracion de variables
int op1 = 20;
int op2 = 20;

if (op1 == op2) {
    System.out.println (op1 + " es igual a " + op2 );
} else {
    System.out.println (op2 + " no son iguales " + op1 );
}
```

## Resultado

```
General Output
-----
20 es igual a 20
Process completed.
```

**NOTA:** El error mas común que se comete al querer usar este operador es solo escribir un solo símbolo igual, vea el ejemplo siguiente:

```
// Declaracion de variables
int op1 = 20;
int op2 = 20;

if (op1 = op2) {
    System.out.println (op2 + " no son iguales " + op1 );
} else {
    System.out.println (op1 + " es igual a " + op2 );
}
```

## Resultado

```
Build Output
-----Confi
C:\Users\delio\Desktop\Nap
    if (op1 = op2) {
1 error
Process completed.
```

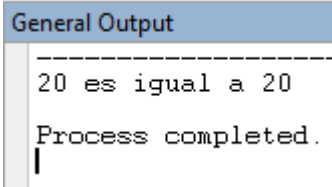
## OPERADOR != ( NO ES IGUAL A)

---

```
// Declaracion de variables
int op1 = 20;
int op2 = 20;

if (op1 != op2) {
    System.out.println (op2 + " no son iguales " + op1 );
} else {
    System.out.println (op1 + " es igual a " + op2 );
}
```

### Resultado



General Output

```
-----
20 es igual a 20
Process completed.
|
```