

Condicionales



¿Que son?

- ✓ Son estructuras que permiten elegir entre la ejecución de una acción u otra.
- ✓ Permiten que se ejecuten conjuntos distintos de instrucciones, en función de que se verifique o no determinada condición.

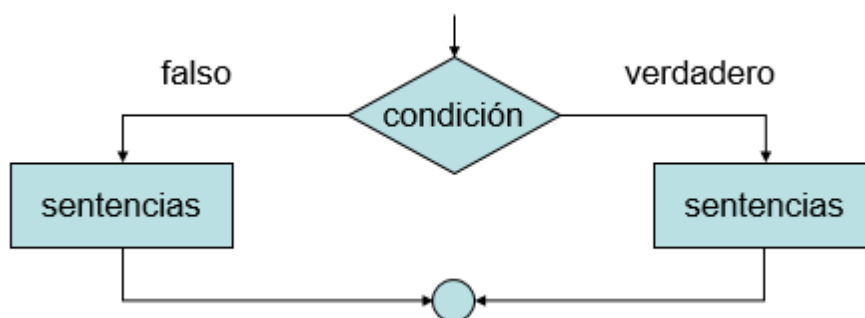


Ilustración 1. Condicional IF

¿Para qué sirven?

- ✓ Permite establecer una serie de condiciones al interior de nuestro programa, que nos ayudan a determinar que acciones llevará cabo dadas ciertas circunstancias.

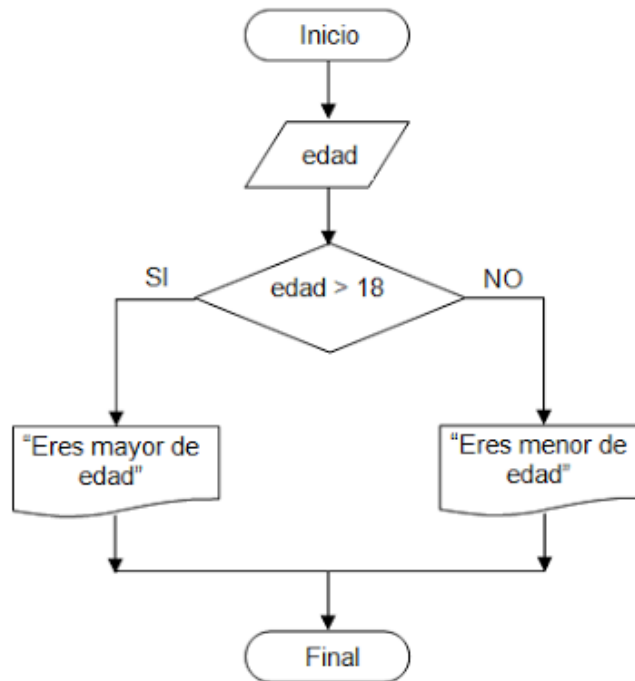
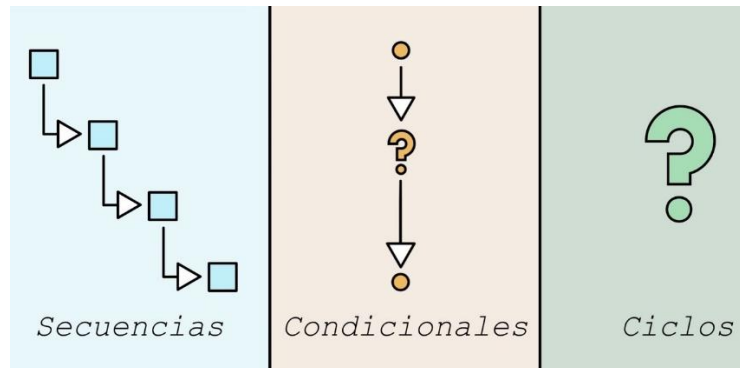


Ilustración 2. Ejemplo en diagrama

```
1 //Condiciona l que los operadores == || != recordad igual a o distinto a
2 var nombre="Juan"
3 var nombre2="Pedro"
4 if(nombre=='Juan' || nombre2!='Pedro'){
5     console.log("Eres juan y no pedro")
6 }else{
7     console.log("No eres Juan y puede que pedro")
8 }
9 //Condiciona l que los operadores == && == recordad igual a Y distinto a
10 if(nombre=='Juan' && nombre2=='Pedro'){
11     console.log("Eres juan y pedro")
12 }else{
13     console.log("No estais los dos juntos")
14 }
```

Ilustración 3. Ejemplo con código de programación.

Tipos



1. Sentencias condicionales

Es encontramos ante una bifurcación(derivación) de un camino, en la que seguiremos por uno u otro camino dependiendo de la respuesta a una pregunta.

- **Sentencia if-else:**

- Evaluará primero su condición y si se evalúa como *true* ejecutará la sentencia/bloque de sentencias que se encuentre justo después de la condición.
- Si la condición se evalúa como *false* y existe una cláusula *else*, entonces se ejecutará la sentencia/bloque de sentencias que siguen al *else*.
- La restricción que se impone a la expresión condicional es que sea una expresión válida de tipo booleano.

- **Sintaxis**

```
if (expresión_condicional)
    sentencia1
else
    sentencia2
```

Ilustración 4. Sintaxis de la condición IF-ELSE

2. Sentencias switch

Se emplea para efectuar selecciones múltiples que bifurcarán a diferentes segmentos de código dependiendo del valor de una variable o una expresión entera.

- Evalúa una expresión entera cuyo valor se usa para encontrar una etiqueta case adecuada entre las presentes. Una etiqueta case adecuada, se ejecutan las sentencias comenzando por la primera sentencencia que la sigue hasta que encontremos una cláusula break

- **Sintaxis**

```
switch (expresion) {  
    case valor1:  
        bloque_código_1;  
        break;  
    case valor2:  
        bloque_código_2;  
        break;  
    ...  
    case valorN:  
        bloque_código_N;  
        break;  
    default:  
        bloque_código_default;  
        break;  
}
```

Ilustración 5. Sintaxis de la condición Switch

3. Sentencias repetidas

Permiten la ejecución repetida de los bloques de sentencias. Existen tipos de sentencias:

- ✚ **Bucles while:** comprueban si deben continuar repitiendo la sentencia/bloque de sentencias que forman el cuerpo del bucle antes de que el cuerpo del se ejecute,
- ✚ **Do-while:** lo hace después de haberlo ejecutado.
- ✚ **For:** cuando sepamos el número de veces se ha repetir la ejecución del cuerpo del bucle.

- Sintaxis

4.2.1 Bucles `while` y `do-while`

La forma general de un bucle `while` es la siguiente:

```
while (expresiónBooleana)
    sentencia;
```

Ilustración 6. Sintaxis del bucle While y do-while

4.2.2 Bucles `for`

La forma general de un bucle `for` es la siguiente:

```
for (exprInicialización; exprBooleana; exprIncremento)
    sentencia;
```

Ilustración 7. Sintaxis del bucle For