

«Talento Tech»

# Desarrollo Web 2

Clase 03





# Clase N° 3 | Condicionales y Operadores

## Temario:

- Introducción a diagramas de flujo
- Estructura if/else
- Operadores de comparación
- Operadores lógicos



## Introducción al diagrama de flujo.

Un diagrama de flujo es una herramienta visual que nos ayuda a entender y representar un proceso de forma mucho más clara. En nuestro día a día, podríamos decir que es un mapa que nos guía en las decisiones que vamos tomando.

Por ejemplo, imaginemos el proceso de si llevamos paraguas o no:

### Diagrama de Flujo: ¿Llevar un paraguas?

#### Objetivo:

Representar visualmente el proceso de decidir si debes llevar un paraguas al salir de casa, basado en las condiciones climáticas.

#### Diagrama en Pasos:

**Inicio:** Inicia el diagrama de flujo.

**Proceso:** Observar las condiciones climáticas.

**Decisión:** ¿Está lloviendo?

Sí: Lleva un paraguas.

No: Continúa con la siguiente condición.

**Decisión:** ¿Está nublado?

Sí: Considera llevar un paraguas por precaución.

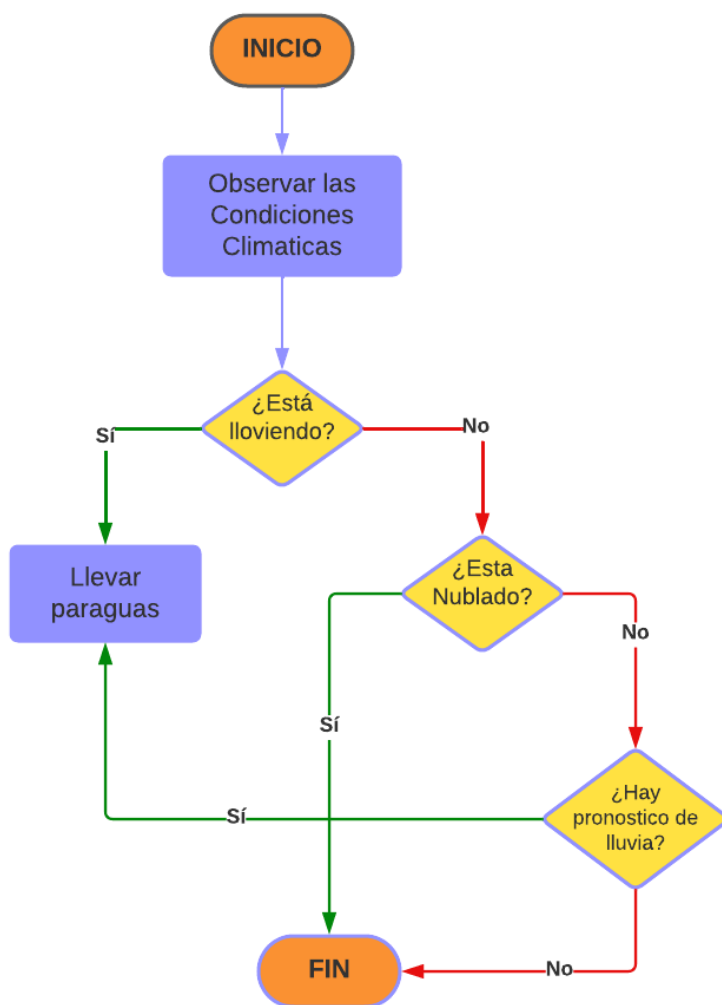
No: Continúa con la siguiente condición.

**Decisión:** ¿Hay pronóstico de lluvia?

Sí: Lleva un paraguas.

No: No es necesario llevar un paraguas.

**Fin:** Finaliza el proceso.



Este diagrama muestra de manera simplificada el proceso de tomar la decisión de llevar un paraguas. Puedes ver cómo las decisiones se toman basadas en las condiciones climáticas, lo que lleva a diferentes acciones. Los rombos representan las decisiones, los rectángulos son procesos y las flechas muestran la secuencia del flujo.

## Estructura if else:

En esta primera parte de la clase 3 veremos la estructura condicional más usada. Tan usada que es parte de nuestra vida cotidiana.

En la primera clase, vimos como funciona el algoritmo para lavarnos los dientes.

En un algoritmo no todo es lineal, se pueden presentar situaciones que desvían nuestro rumbo.

**Por ejemplo:** ¿Qué ocurriría en ese caso si vamos a buscar el dentífrico y nos encontramos con que no tenemos más?

Aquí es donde se hacen necesarias las estructuras condicionales:

```
SI hay dentífrico{  
    Me cepillo los dientes  
}
```

Pero, en cambio, si no hay dentífrico, tenemos que interrumpir momentáneamente el algoritmo, para ir al supermercado, comprar el dentífrico, volver a nuestro hogar y seguir los pasos del algoritmo.

```
SI hay dentífrico{  
    Me cepillo los dientes  
} SI NO{  
    Ir al supermercado  
    Comprar dentífrico  
    Volver al hogar  
    Cepillarme los dientes  
}
```

Para eso, se utilizan las estructuras de if-else (traducido, queda “si - sino”)

**Ahora vamos a trabajar otro ejemplo pero utilizando código:**

Vamos a determinar si una persona es mayor o menor de 18 años. El algoritmo es el siguiente:

```
const edad = 12;
if (edad >= 18) {
  console.log("Nick es un adulto");
} else {
  console.log("Nick es un niño");
}
```

### Resultado:

Es un niño

Habrán situaciones donde será necesario probar múltiples condiciones. Aquí entra el bloque `else if`:

Cuando la sentencia `if` es **false** (falsa), el algoritmo pasará a la sentencia **else if**. Si eso también es false, entonces se moverá al bloque `else`.

### Ejemplo:

```
const edad = 18;
if (edad < 18) {
  console.log("Catalina es menor de 18 años");
} else if (edad >= 18 && edad <= 21) {
  console.log("Catalina tiene entre 18 y 21 años de edad");
} else {
  console.log("Catalina tiene mas de 21 años");
}
```

### Resultado:

Catalina tiene entre 18 y 21 años de edad.

## Operadores de comparación

Estos operadores permiten comparar variables o expresiones entre sí. El resultado de la comparación es un valor booleano. Permiten determinar si una variable es menor, igual, mayor o simplemente distinta a otra.

EJEMPLO	NOMBRE	RESULTADO
a==b	Igual	True si a es igual a b.
a===b	Idéntico	True si a es igual a b, y son del mismo tipo.
a!=b a<>b	Diferente	True si a no es igual a b.
a!==b	No idénticos	True si a no es igual a b, o si no son del mismo tipo.
a<b	Menor que	True si a es estrictamente menor que b.
a>b	Mayor que	True si a estrictamente mayor que b.
a<=b	Menor o igual que	True si a es menos o igual que b.
a>=b	Mayor o igual que	True si a es mayor o igual que b.

Los operadores de comparación se utilizan en la práctica de esta manera:

```
let edadMario = 20;
let edadCarla = 15;
if (edadMario > edadCarla){
    console.log("Mario es mayor que Carla")
} else if (edadMario < edadCarla){
    console.log("Mario es menor que Carla");
} else {
    console.log("Mario y Carla tienen la misma edad");
}
```

**Resultado:**

Mario es mayor que Carla



## Operadores lógicos

Los operadores lógicos son herramientas en programación representadas con símbolos o palabras clave que son utilizados para combinar o modificar condiciones lógicas. Estos operadores son importantes para las estructuras de control y las expresiones condicionales, ya que permiten evaluar múltiples condiciones y determinar si una expresión es verdadera o falsa.

Los 3 principales son la **conjunción (AND)**, la **disyunción (OR)** y la **negación (NOT)**

### Conjunción - Operador AND (&&)

Este operador se representa con el símbolo “&&” y como regla requiere que dos condiciones sean verdaderas al mismo tiempo. Es decir, si tenemos 2 condiciones ambas deben ser Verdaderas para que la operación con el operador AND sea Verdadera.

Por ejemplo, supongamos que se realiza una Carrera de atletas en Buenos Aires pero los requisitos que se deben cumplir son:

Tener 18 años o más.

Vivir en CABA

Soliciantes	Edad	Residencia	Estado de Solicitud
Paula	19	CABA	APROBADA
Marcelo	17	CABA	DESAPROBADA
Lucas	21	La Plata	DESAPROBADA
Camila	16	Moron	DESAPROBADA

Con este ejemplo, sabemos que si o si deben cumplirse los 2 requisitos, en consecuencia la única persona habilitada a participar es Paula.





Escrito en código se vería de la siguiente manera:

```
let edad = 19;
let residencia = "CABA";

if(edad >= 18 && residencia == "CABA"){
  console.log("Solicitud APROBADA");
}else{
  console.log("Solicitud DENEGADA");
}
```

**Resultado:**

Solicitud APROBADA

## Operador OR (||)

Este operador se representa con el símbolo “||” y se caracteriza como el operador más flexible, debido a que, si tenemos 2 condiciones y al menos una se cumple, entonces, el resultado de la operación es Verdadera.

Por ejemplo, supongamos que en una tienda de repuestos de autos solo pueden comprar los clientes que cumplan con al menos uno de los siguientes requisitos:

Ser mayor de edad.  
Tener un Auto.

Si usas el operador OR, significa que podrán comprar quienes sean mayor de edad o tengan un auto. No necesitan ambos requisitos al mismo tiempo; cualquiera de ellos hará que puedan comprar. El operador OR permite que al menos una de las condiciones sea verdadera para que el resultado sea positivo.

Cientes	Es mayor de Edad?	Tiene auto?	¿Puede Comprar?
Paula	SI	SI	SI
Marcelo	NO	SI	SI
Jessica	SI	NO	SI
Camila	NO	NO	No

**Veamos el caso de Jessica en Código:**

```

if (esMayor || tieneAuto){
  console.log("Jessica puede comprar");
}else{
  console.log("Jessica puede comprar");
}
  
```

**Resultado:** Jessica puede comprar

## Operador NOT (!)

El operador NOT es representado por el símbolo ! y se utiliza para negar o invertir el valor de verdad de una expresión booleana. En otras palabras, convierte un valor verdadero en falso y un valor falso en verdadero.

**Por ejemplo:**

Si la variable **tieneTarea** es **verdadero**, entonces **!tieneTarea** es **falso**.  
 Si la variable **esMayor** es **falso**, entonces **!esMayor** es **verdadero**.

```

let esMayor = true;
if (!esMayor){ // SI NO (!) ES MAYOR (esMayor)
  //!esMayor = falso
  console.log("No es mayor")
}else{
  console.log("Es mayor")
}
  
```

**Resultado:** Es mayor.

Operación	Operador	Ejemplo	Comparación	Resultado/Binario
AND	&&	9 > 8 && 11 > 10	¿9 es mayor que 8 Y 11 es mayor que 10?	Verdadero (1)
		9 > 7 && 10 < 9	¿9 es mayor que 7 Y 10 es menor que 9?	Falso (0)
OR		5 > 4    3 > 3	¿5 es mayor que 4 O 3 es mayor que 3?	Verdadero (1)
		5 > 5    3 == 4	¿5 es mayor que 5 O 3 es igual a 4?	Falso (0)
NOT	!	!(5 > 4)	¿5 NO es mayor que 4?	Falso (0)
		!(5 < 4)	¿5 NO es menor que 4?	Verdadero (1)

## ? Pregunta de reflexión o de llamado de atención

Considerando la introducción a los diagramas de flujo, las estructuras **if/else**, los operadores de comparación y lógicos en JavaScript, así como su aplicación práctica en la toma de decisiones.

- ¿Cómo crees que estas herramientas y conceptos podrían ser útiles en el desarrollo de proyectos de programación o en situaciones cotidianas?
- ¿Se te ocurre alguna situación cotidiana o profesional en la que el uso de un diagrama de flujo sería beneficioso para representar un proceso o tomar decisiones?



## Desafío #3

Imagina que estás desarrollando la lógica de precios de un E-commerce.

Declara variables para el nombre del producto (`nombreProducto`), el precio unitario (`precioUnitario`) y la cantidad deseada (`cantidadDeseada`). Para esta última utiliza **`prompt()`** tal como hicimos en el desafío anterior.

Ahora vamos a implementar un condicional para aplicar descuentos según la siguiente regla:

Si la cantidad comprada es mayor o igual a 5, aplica un descuento del 10%.

Para finalizar, calcula el costo total de la compra, aplicando el descuento según la condición y mostrar el total en un **`alert()`**.



**Buenos Aires**  
*aprende*  
Agencia de Actividades para el Futuro

**BA** Buenos  
Aires  
Ciudad