

```
1 import { useState } from "react";
2
3 // Programación I - Operadores lógicos y
4 // sentencias Condicionales.
```

```
5
6 class Condicionales() {
7
```

```
8
9     const [prof, setProf] = useState({
```

```
10         setProf: "Miguel Silva."
11     });
```

```
12
```

```
13
14
15     return prof;
16
```

```
17
18
19 };
```

```
20
21
22 export default Condicionales;
23
```



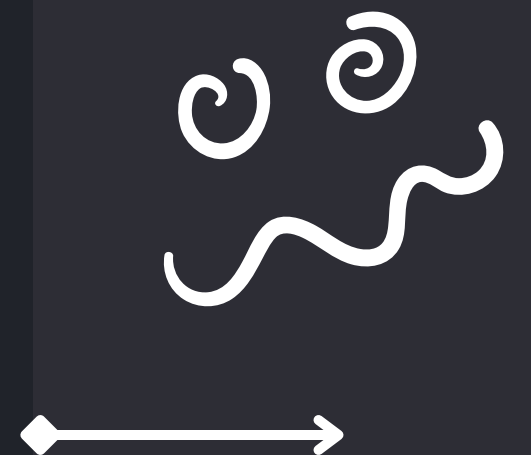
```
1 // Programación I - Op. lógicos y condicionales
```

```
2  
3  
4  
5 operadores Logicos() {
```

- Son herramientas que permiten combinar o modificar expresiones lógicas.
- Utilizan principalmente estructuras de control condicionales para tomar decisiones.

```
18 };
```

```
21  
22 export default Logicos;
```



Los operadores
lógicos más
comunes son:



"The ticher"

NOT (!): Se utiliza para negar una expresión lógica. Si la expresión original es verdadera, la convierte en falsa, y viceversa.

AND (&&): Devuelve verdadero (true) si ambas expresiones que combina son verdaderas. En caso contrario, devuelve falso (false):

OR (||): Devuelve verdadero (true) si al menos una de las expresiones es verdadera. Si ambas expresiones son falsas, devuelve falso (false):



```
// Programación I - Op. lógicos y condicionales
```

```
operaciones Logicas() {
```

Negación Conjunción Disyunción
(NO, -) (Y, AND, &&) (O, OR, ||)

A	B	! A	A && B	A B
Cierto	Cierto	Falso	Cierto	Cierto
Cierto	Falso	Falso	Falso	Cierto
Falso	Cierto	Cierto	Falso	Cierto
Falso	Falso	Cierto	Falso	Falso

```
};
```

```
export default Logicas;
```



```
// Programación I - Op. lógicos y condicionales
```

```
tipos operadoresLogicos() {
```

Menor que

<

Mayor que

>

Menor o igual que

<=

Mayor o igual que

>=

Igual que

==

Distinto que

!=

Conjunción ó Y lógico

&&

Disyunción u O lógico

||

Negación ó NO lógico

!

```
};
```

```
export default operadoresLogicos;
```



```
// Programación I - Op. lógicos y condicionales
```

WARNING !!!

**Los operadores lógicos
solo pueden confirmar o
negar el resultado...**



Realidad simulada.



```
1 // Programación I - Op. lógicos y condicionales
```

WARNING !!!

**... entonces tomemos 1
y 0 como respuestas!!!**



```
22 Realidad simulada.
```



```
1 // Programación I - Op. lógicos y condicionales
```

Ejemplo 01:

```
1 #include <stdio.h>
2 #define p printf
3
4 int main()
5 {
6     int num1, num2, resultado;
7
8     num1 = 1;
9     num2 = 2;
10
11     resultado = num1 >= num2;
12
13     p("\n1er resultado: %d", resultado);
14
15     return 0;
16 }
17
```




```
1 // Programación I - Op. lógicos y condicionales
```

```
2  
3 1º salida...
```

```
4  
5  
6  
7  
8  
9  
10 1er resultado: 0
```

```
11  
12  
13 ...Program finished with exit code 0  
14 Press ENTER to exit console.█  
15  
16
```



// Programación I - Op. lógicos y condicionales

Ejemplo 02:

```
1  #include <stdio.h>
2  #define p printf
3
4  int main()
5  {
6      int num1, num2, resultado;
7
8      num1 = 1;
9      num2 = 2;
10
11     resultado = num1 && num2;
12
13     p("\n2do resultado: %d", resultado);
14
15     return 0;
16 }
17
```



1 // Programación I - Op. lógicos y condicionales

2
3 2º salida...

4
5
6
7
8
9
10 2do resultado: 1

11
12
13 ...Program finished with exit code 0
14 Press ENTER to exit console.█
15
16



```
// Programación I - Op. lógicos y condicionales
```

```
concepto Condicionales() {
```

- Son estructuras de control que permiten tomar decisiones basadas en la evaluación de una condición.
- El resultado determina qué acciones se ejecutarán.

```
};
```

```
export default Condicionales;
```



// Programación I - Op. lógicos y condicionales

¿Entonces esto me
ayuda a tomar
decisiones
de vida?



"Cosme Fulanito"

Realidad simulada.



```
// Programación I - Op. lógicos y condicionales
```

Decisiones de vida
no sé...
Pero sí con respecto a
los parámetros que
quieras evaluar en tu
código.



"The ticher"

Realidad simulada.



```
1 // Programación I - Op. lógicos y condicionales
```

```
2  
3  
4 Concepto Condicionales() {  
5  
6  
7
```

```
8     En la mayoría de los lenguajes de  
9     programación, las condicionales  
10    se implementan utilizando  
11    estructuras como if, else y  
12    switch.  
13  
14  
15  
16
```

```
17  
18  
19 };  
20
```

```
21  
22 export default Condicionales;  
23
```



```
1 // Programación I - Op. lógicos y condicionales
```

```
2  
3  
4 Concepto Condicionales() {  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23
```

if: Evalúa una condición y ejecuta un bloque de código si la condición es verdadera. Si es falsa, el bloque de código no se ejecuta.

```
};
```

```
export default Condicionales;
```




```
1 // Programación I - Op. lógicos y condicionales
```

```
2  
3  
4 Concepto Condicionales() {  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23
```

else: Se utiliza en conjunto con *if* para especificar un bloque de código que se ejecutará si la condición en ese *if* es falsa.

```
};
```

```
export default Condicionales;
```



```
1 // Programación I - Op. lógicos y condicionales
```

Ejemplo if:

```
1 #include <stdio.h>
2 #define p printf
3 #define s scanf
4
5 int main()
6 {
7     int edad;
8
9     edad = 20;
10
11     if(edad >= 18)
12     {
13         p("Edad: %d", edad);
14         p("\nEs mayor de edad.");
15     }
16     else
17     {
18         p("Edad: %d", edad);
19         p("\nNo es mayor de edad.");
20     }
21
22     return 0;
23 }
```

Las sentencias tanto dentro de if como de else van dentro de llaves si son más de una...



```
1 // Programación I - Op. lógicos y condicionales
```

Ejemplo if:

```
2 #include <stdio.h>
3 #define p printf
4 #define s scanf
5
6 int main()
7 {
8     int edad;
9
10    edad = 20;
11
12    if(edad >= 18)
13        p("Es mayor de edad.");
14    else
15        p("No es mayor de edad.");
16
17    return 0;
18 }
```

**... pero si es
solo una
sentencia no
necesita las
llaves.**



```
1 // Programación I - Op. lógicos y condicionales
```

Ejemplo if:

Puede haber if
dentro de otro if
o un else...

```
2 #include <stdio.h>
3 #define p printf
4 #define s scanf
5
6 int main()
7 {
8     int num1, num2;
9
10    num1 = 20;
11    num2 = 5;
12
13    if(num1 >= num2)
14    {
15        if(num1 > num2)
16        {
17            p("%d > %d", num1, num2);
18        }
19        else
20        {
21            p("%d = %d", num1, num2);
22        }
23    }
24    else
25    {
26        p("%d < %d", num1, num2);
27    }
28
29    return 0;
30 }
```

... se le conoce
como if
anidados



```
1 // Programación I - Op. lógicos y condicionales
```

Ejemplo if:

... Y

principalmente

```
2 #include <stdio.h>
3 #define p printf
4 #define s scanf
5
6 int main()
7 {
8     int num1, num2;
9
10    num1 = 20;
11    num2 = 5;
12
13    if(num1 > num2)
14        p("%d > %d", num1, num2);
15    else
16        if(num1 < num2)
17            p("%d < %d", num1, num2);
18        else
19            p("%d = %d", num1, num2);
20
21    return 0;
22 }
23
```

...Existen muchas formas de solucionar un mismo problema.



```
// Programación I - Op. lógicos y condicionales
```

Ejemplo if:

```
1 #include <stdio.h>
2 #define p printf
3 #define s scanf
4
5 int main()
6 {
7     int num;
8
9     num = 4;
10
11
12     if(num == 0)
13         p("Valor 0");
14     else if(num == 1)
15         p("Valor 1");
16     else if(num == 2)
17         p("Valor 2");
18     else if(num == 3)
19         p("Valor 3");
20     else
21         p("Otro valor: %d", num);
22
23     return 0;
24 }
```

¡ATENCIÓN!

**Si se van a hacer
varias
comparaciones,
existe la sintaxis
"else if".**



EJERCITACIÓN (level 1):

Realizar las siguientes operaciones lógicas y aritméticas y mostrar los resultados por pantalla:

1. $(25 \ \&\& \ 3) + 2$

2. $3 * (7.6 \ || \ 0)$

3. $75 / (1 \ \&\& \ 0)$

4. $(0 \ || \ 5) + (3.14159 \ \&\& \ 2.14)$

5. $5 * (0x5 \ \&\& \ 3)$

6. $(0x5 \ \&\& \ 0xA7) * (0 \ || \ 0x0)$

7. $(\text{preg. 4}) + (\text{preg. 2}) \ \&\& \ (\text{preg. 1})$

Luego compararlas a través del código del ejemplo anterior ingresándola en la variable "***num***".



EJERCITACIÓN (level 2):

1. Escriba un programa que pida un año y que escriba si es bisiesto o no.
2. Escriba un programa que pida dos números enteros y que calcule su división, escribiendo si la división es exacta o no.
3. Mejore el programa anterior haciendo que tenga en cuenta que no se puede dividir por cero.
4. Escriba un programa que pida dos números y que conteste cuál es el menor y cuál el mayor o que escriba que son iguales.
5. Escriba un programa que pida una distancia en centímetros y que escriba esa distancia en kilómetros, metros y centímetros (escribiendo todas las unidades).



EJERCITACIÓN (level 2.1):

1. Para tributar un determinado impuesto se debe ser mayor de 21 años y tener unos ingresos iguales o superiores a \$80000 pesos mensuales. Escribir un programa que pregunte al usuario su edad y sus ingresos mensuales y muestre por pantalla si el usuario tiene que tributar o no.



EJERCITACIÓN (level 2.4):

1. Escribir un programa que pregunte al usuario su renta mensual y muestre por pantalla el porcentaje impositivo que le corresponde.

Los tramos impositivos para la declaración de la renta son los siguientes:

RENTA	TIPO IMPOSITIVO
Menos de \$1000000	5%
Entre \$1000000 y \$3500000	15%
Entre \$3500000 y \$7000000	20%
Entre \$7000000 y \$10000000	30%
Más de \$10000000	45%



EJERCITACIÓN (level 2.5):

1. Escriba un programa que pregunte primero si se quiere calcular el área y perímetro de un rectángulo o la de un círculo. Si se contesta que se quiere calcular el área y perímetro de un rectángulo (escribiendo 1), el programa tiene que pedir entonces la base y la altura y mostrar el área y el perímetro. Si se contesta que se quiere calcular el área de un círculo (escribiendo 2), el programa tiene que pedir entonces el radio y mostrar el área y el perímetro. Se debe indicar si la opción ingresada es incorrecta.

(Tomar ***PI*** = ***3.14***)



EJERCITACIÓN (level 3):

1. Escriba un programa que pida el año actual y un año cualquiera y que escriba cuántos años han pasado desde ese año, cuántos años faltan para llegar a ese año o si es el mismo año.
2. Mejore el programa anterior haciendo que cuando la diferencia sea exactamente un año, escriba la frase en singular.
3. Escriba un programa que pida dos números enteros y que escriba si el mayor es múltiplo del menor.
4. Mejore el programa anterior haciendo que el programa avise cuando se escriben valores negativos o nulos.
5. Escriba un programa que pida tres números y que escriba si son los tres iguales, si hay dos iguales o si son los tres distintos.



EJERCITACIÓN (level 4):

1. Escriba un programa que pida los coeficientes de una ecuación de primer grado ($a x + b = 0$) y escriba la solución.

Se recuerda que una ecuación de primer grado puede no tener solución, tener una solución única, o que todos los números sean solución. Se recuerda que la fórmula de las soluciones es $x = -b / a$



EJERCITACIÓN (level 4):

Estos son algunos ejemplos de posibles respuestas (el orden de los ejemplos no tiene por qué corresponder con el orden de las condiciones).

ECUACIÓN $A X + B = 0$

Escriba el valor del coeficiente a: 0

Escriba el valor del coeficiente b: 3

La ecuación no tiene solución.

ECUACIÓN $A X + B = 0$

Escriba el valor del coeficiente a: 4.2

Escriba el valor del coeficiente b: 21

La ecuación tiene una solución: -5.0

ECUACIÓN $A X + B = 0$

Escriba el valor del coeficiente a: 0

Escriba el valor del coeficiente b: 0

Todos los números son solución.



REALIZADO POR MIGUEL SILVA C.

 miguel.silva@inspt.utn.edu.ar

© Esta presentación cuenta con derechos de autor.

