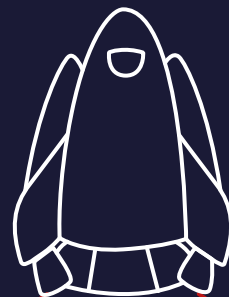
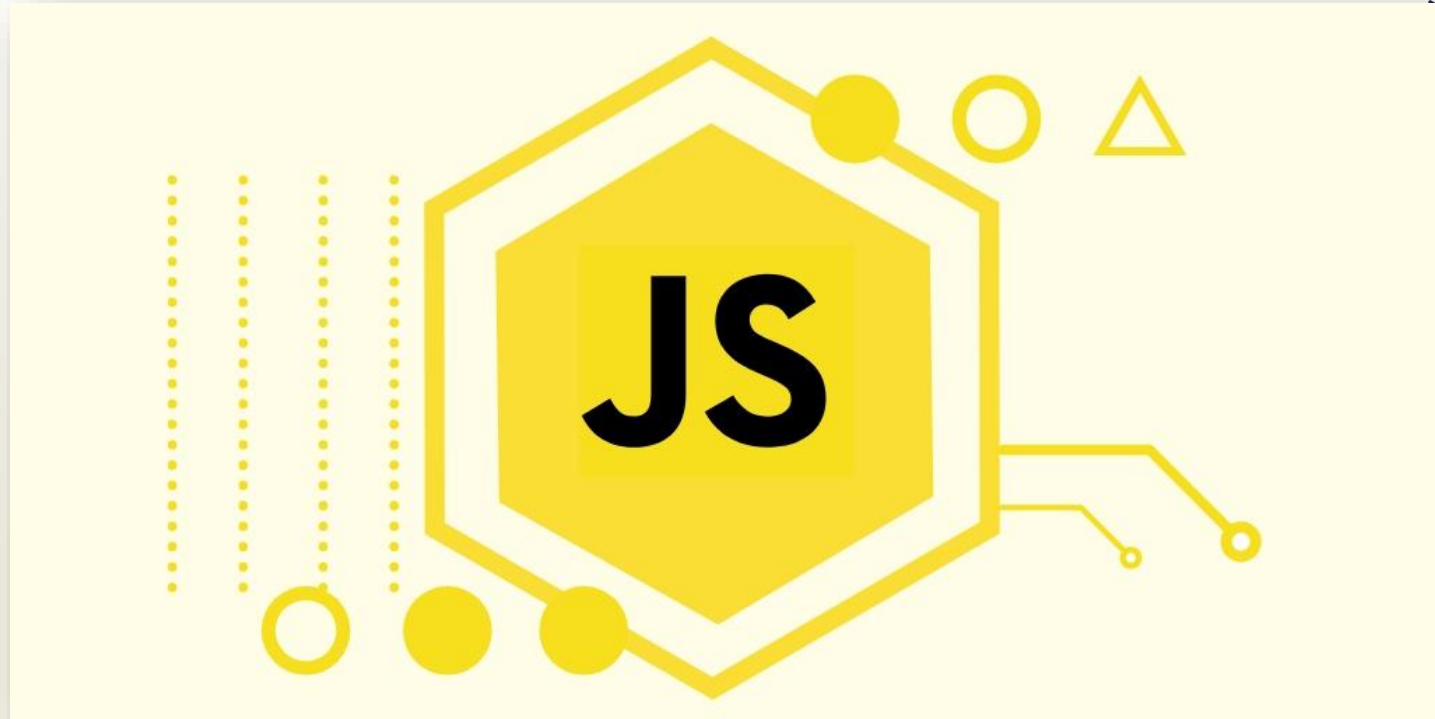


# Programa acadêmico CAMPUS



# JAVASCRIPT - CONDICIONALES



# JAVASCRIPT + CONDITIONALS



HTML



HTML + CSS



HTML + CSS  
+ JAVASCRIPT



JS

# JAVASCRIPT - CONDICIONALES



- **IF sencillos**

- Si la condición es verdadera, se ejecuta el código dentro del bloque "if".

- Si la condición es falsa, el código dentro del bloque "if" se omite y se continúa con la ejecución del programa.



# JAVASCRIPT - CONDICIONALES



- **IF sencillos**

```
if (x > 10) {  
    console.log("x es mayor que 10");  
}
```



# JAVASCRIPT - CONDICIONALES



- **If .. Else**

Esta estructura se utiliza para ejecutar un bloque de código si se cumple una condición y otro bloque de código si no se cumple esa condición.



# JAVASCRIPT - CONDICIONALES



- **If .. Else**

```
if (x > 10) {  
    console.log("x es mayor que 10");  
} else {  
    console.log("x es menor o igual que 10");  
}
```



JS

# JAVASCRIPT - CONDICIONALES



- **If .. Else if .. [Else]**

Esta estructura se utiliza para ejecutar un bloque de código si se cumple una condición entre una serie de condiciones. Si ninguna se cumple, entra al bloque del Else si existe.





# JAVASCRIPT - CONDICIONALES



- **Switch**

esta estructura se utiliza para ejecutar diferentes bloques de código según el valor de una expresión.



# JAVASCRIPT - CONDICIONALES



- ## Switch

```
switch (color) {  
  case "rojo":  
    console.log("El color es rojo");  
    break;  
  case "verde":  
    console.log("El color es verde");  
    break;  
  default:  
    console.log("El color no es ni rojo ni verde");  
}
```



JS

# JAVASCRIPT - CONDICIONALES



- **Operador Ternario:**

Ofrece una forma concisa de escribir una instrucción if-else en una sola línea.



# JAVASCRIPT - CONDICIONALES



```
let resultado = (edad ≥ 18) ? "Eres mayor de edad" : "Eres menor de edad";  
console.log(resultado);
```



# JAVASCRIPT - CONDICIONALES



## • Operador Nullish Coalescing

El operador Nullish Coalescing, también conocido como operador **??**, es un operador lógico introducido en **ECMAScript 2020** que permite comprobar si una variable o expresión es nula o indefinida y, en ese caso, devolver un valor predeterminado en su lugar.



# JAVASCRIPT - CONDICIONALES



## • Operador Nullish Coalescing

Sintaxis:

```
value ?? defaultValue;
```

Donde **value** es la variable o expresión que se va a comprobar, y **defaultValue** es el valor que se devolverá si value es nula o indefinida.



# JAVASCRIPT - CONDICIONALES



```
const nombre = undefined;  
const cadena = `Bienvenido, ${nombre ?? "Invitado"}!`;   
console.log(cadena)
```

Bienvenido, Invitado!

// Devuelve el valor de la variable `edad` si está definida, o 18 si no está definida.

```
const edad = 20;  
const edadEstandar = edad ?? 18  
console.log(edadEstandar)
```

20

[debugger eval code](#)

JS

# JAVASCRIPT - INTRODUCCIÓN



- **Ejercicio 1: Resolver una ecuación cuadrática**

El usuario ingresa los coeficientes de una ecuación cuadrática. El programa debe resolver la ecuación y mostrar las soluciones por pantalla.

$$ax^2 + bx + c = 0$$
$$x = \frac{-b \pm \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}$$





# JAVASCRIPT - INTRODUCCIÓN



## Ejercicio 2: Verificar si una fecha es válida

El usuario ingresa una fecha en formato DD/MM/AAAA.  
El programa debe verificar si la fecha es válida y mostrar el resultado por pantalla.



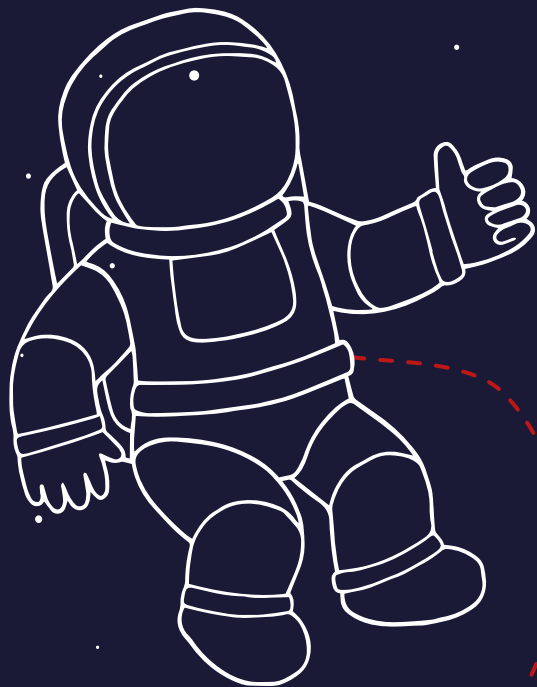
# JAVASCRIPT - INTRODUCCIÓN



## Ejercicio 3: Cuantitativo a Cualitativo

El usuario ingresa una calificación numérica. El programa debe imprimir la letra correspondiente a esa calificación.

Calificación numérica	Calificación alfabética
1-3	Insuficiente
4	Suficiente
5	Bueno
6	Notable
7-8	Sobresaliente



# Programa acadêmico CAMPUS

Ciclo 2

