PROTECO

control de flujo en js

intersemestral 2024-1

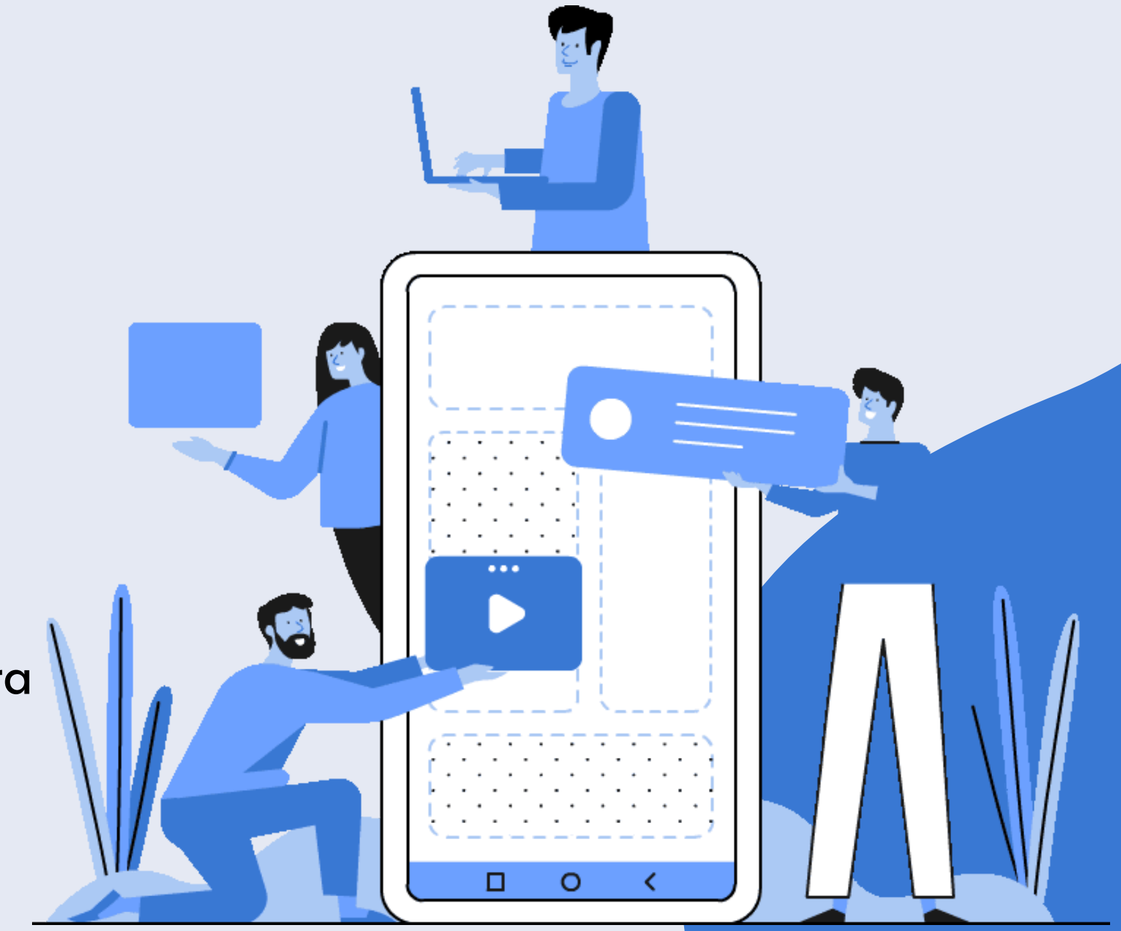
ESTRUCTURAS DE CONTROL

- El control de flujo es esencial en programación para gestionar cómo se ejecutan las instrucciones en un programa.
- Permite tomar decisiones, repetir tareas y responder de manera dinámica a las condiciones del entorno.

WHILE

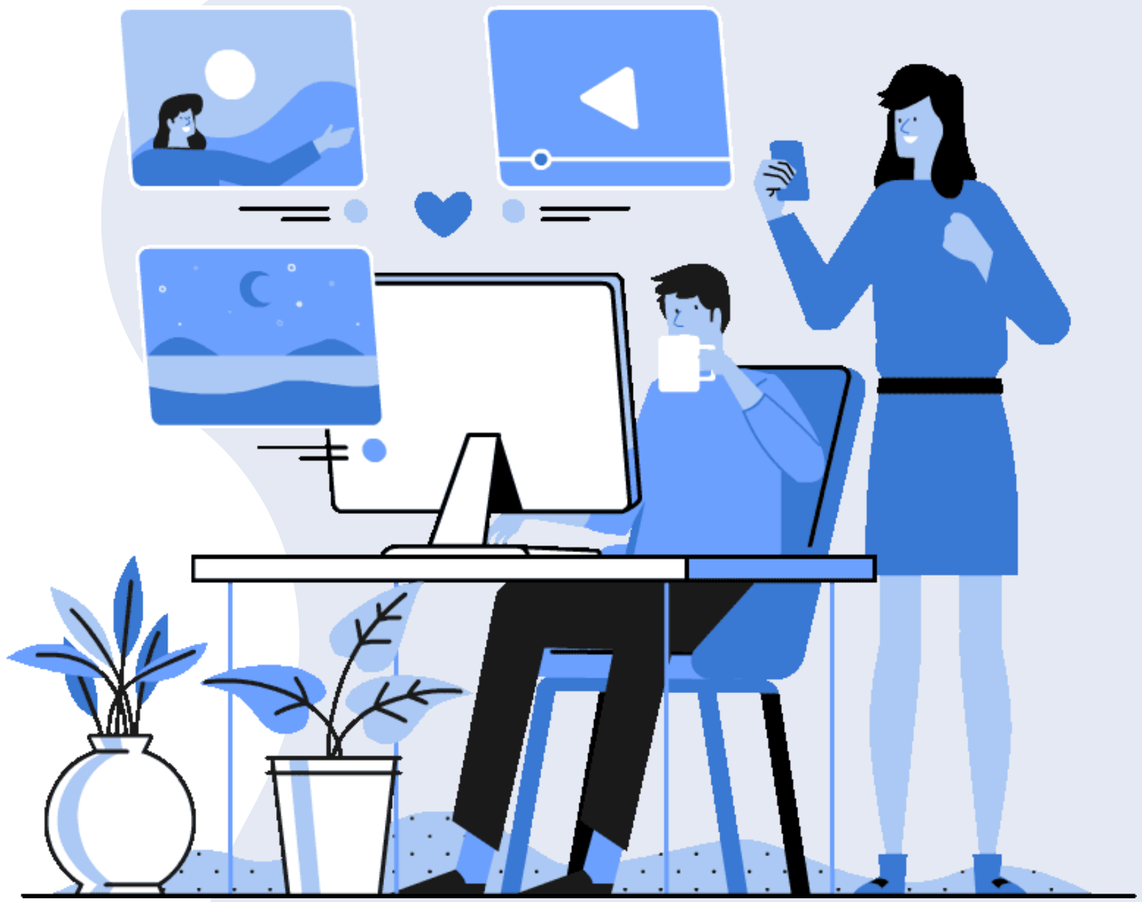
El bucle while en JavaScript es una estructura de control que repite un bloque de código mientras una condición dada sea verdadera.

```
while (condición) {
    // Código a ejecutar mientras la condición sea verdadera
}
```



IMPLEMENTACION

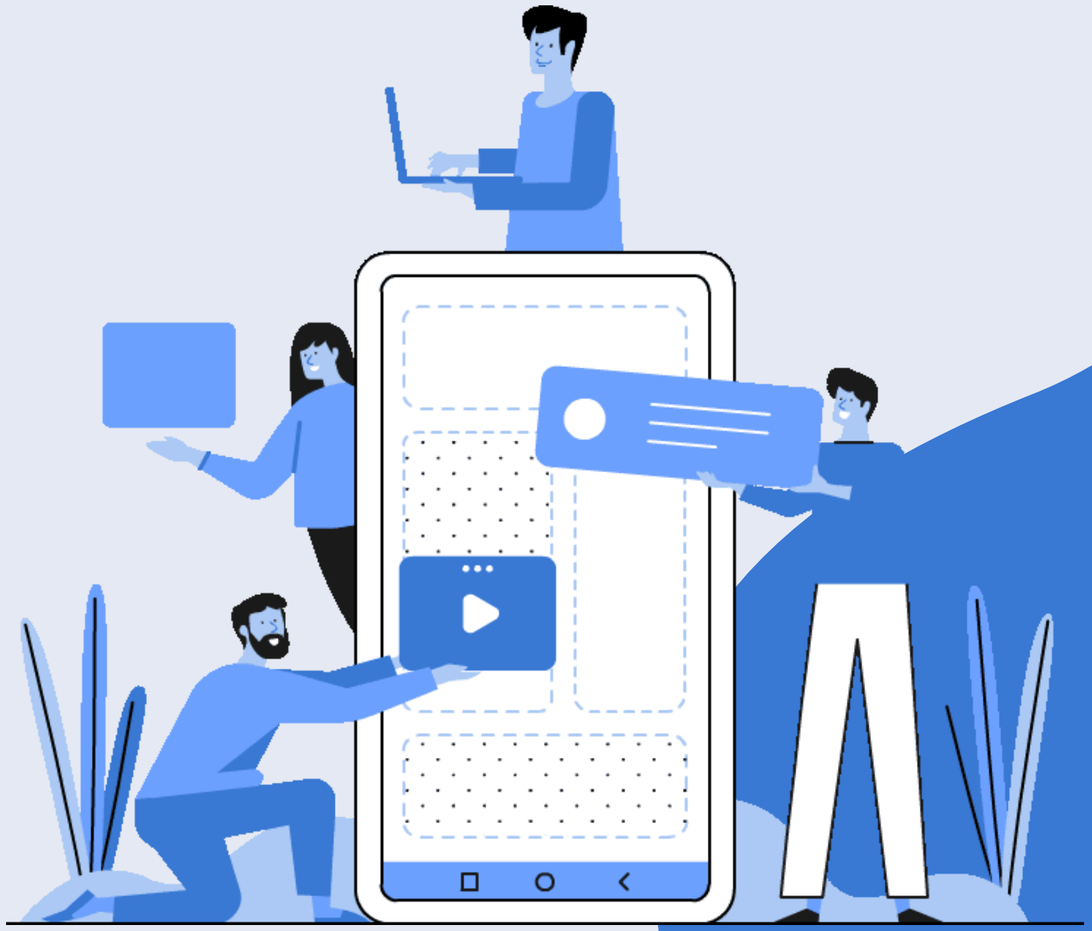
```
let contador = 1;

while (contador <= 5) {
    console.log(`Contador: ${contador}`);
    contador++;
}
```



DO-WHILE

El bucle do-while es similar al while, pero garantiza que el bloque de código se ejecute al menos una vez, ya que la condición se evalúa después de la ejecución del bloque.

```
do {
    // Código a ejecutar al menos una vez
} while (condición);
```



IMPLEMENTACION

Asegura que el juego se ejecute al menos una vez, independientemente de la condición. El usuario intenta adivinar el número, recibiendo retroalimentación hasta que lo adivina correctamente.

```
let numeroCorrecto = 7;
let intentoUsuario;
let intentos = 0;

do {
    intentoUsuario = parseInt(prompt("Adivina el número (entre 1 y 10):"));
    intentos++;

    if (intentoUsuario !== numeroCorrecto) {
        console.log("Incorrecto. ¡Intenta de nuevo!");
    }
} while (intentoUsuario !== numeroCorrecto);

console.log(¡Correcto! Adivinaste el número en ${intentos} intentos.);
```

CONTROL DE BUCLES Y OPTIMIZACION

El control de bucles es fundamental en la programación para dirigir la ejecución de código de manera efectiva. En JavaScript, las instrucciones break y continue son herramientas esenciales que permiten modificar el flujo de un bucle.

BREAK Y CONTINUE

- Uso de break para Salir de un Bucle:
  - break se utiliza para salir inmediatamente de un bucle, independientemente de si la condición del bucle aún se cumple.
- Uso de continue para Saltar a la Siguiente Iteración:
  - continue se utiliza para saltar a la siguiente iteración del bucle, omitiendo el resto del código dentro del bucle en la iteración actual.

```
const limite = 10;

for (let i = 1; i <= limite; i++) {
    // Utilizando break para salir del bucle si se alcanza un número espec
    if (i === 5) {
        console.log("Se alcanzó el número 5. Saliendo del bucle con break.")
        break;
    }

    // Utilizando continue para omitir la impresión de números impares
    if (i % 2 !== 0) {
        continue;
    }

    console.log(Número par: ${i});
}
```

SWITCH

La estructura switch en JavaScript se utiliza para realizar múltiples comparaciones sobre el valor de una expresión y ejecutar bloques de código según el caso coincidente.


```
switch (expresion) {
    case valor1:
        // Código a ejecutar si expresion es igual a valor1
        break;
    case valor2:
        // Código a ejecutar si expresion es igual a valor2
        break;
    // ... más casos ...
    default:
        // Código a ejecutar si ninguno de los casos coincide
}
```