GUÍA TEÓRICO-PRÁCTICA:

UNIDAD I

SEMANA 4: BUCLES Y ARRAYS

TEMA 4.1: BUCLES - for, while

Teoría

Bucles (Loops): Permiten repetir un bloque de código múltiples veces.

for:

Se usa cuando se conoce el número de iteraciones.

```javascript
for (inicializacion; condicion; actualizacion) {
    // Código a repetir
}

for (let i = 0; i < 5; i++) {
    console.log("Iteración:", i);
}
```

while:

Se usa cuando no se conoce el número de iteraciones.

```javascript
while (condicion) {
    // Código a repetir
}

let i = 0;
while (i < 5) {
    console.log("Iteración:", i);
    i++;
}
```

do-while:

Ejecuta al menos una vez.

```javascript
do {
    // Código
} while (condicion);
```

Pregunta de reflexión 13: ¿Cuándo es mejor usar while en lugar de for?

Ejemplo

```javascript
// Ejemplo for
console.log("=== Contar del 1 al 5 ===");
for (let i = 1; i <= 5; i++) {
    console.log(i);
}

// For decremental
console.log("=== Cuenta regresiva ===");
for (let i = 5; i >= 1; i--) {
    console.log(i);
}
console.log("¡Despegue!");

// For con array
let frutas = ["manzana", "banana", "naranja"];
console.log("=== Frutas ===");
for (let i = 0; i < frutas.length; i++) {
    console.log(frutas[i]);
}

// While
console.log("=== While ===");
let contador = 0;
while (contador < 3) {
    console.log("Contador:", contador);
    contador++;
}

```

```javascript
// Do-while
console.log("=== Do-while ===");
let numero;
do {
    numero = Math.floor(Math.random() * 10);
    console.log("Número:", numero);
} while (numero !== 5);
console.log("¡Encontré el 5!");

// Break y continue

console.log("=== Break ===");
for (let i = 0; i < 10; i++) {
    if (i === 5) break;
    console.log(i);
}

console.log("=== Continue ===");
for (let i = 0; i < 5; i++) {
    if (i === 2) continue;
    console.log(i);
}
```

Ejercicio 4.1

Crea programas que:

1. Use for para mostrar la tabla de multiplicar del 7
2. Use while para sumar números hasta que la suma sea mayor a 100
3. Use for anidado para crear un triángulo de asteriscos
4. Use break para salir de un bucle cuando encuentre un número negativo

```javascript
// Tu código aquí:
```

TEMA 4.2: ARRAYS BÁSICOS

Teoría

Métodos de arrays:

Agregar/eliminar:

- push(): Agrega al final
- pop(): Elimina del final
- unshift(): Agrega al inicio
- shift(): Elimina del inicio
- splice(): Agrega/elimina en cualquier posición

Buscar:

- indexOf(): Encuentra la posición de un elemento
- includes(): Verifica si existe un elemento
- find(): Encuentra el primer elemento que cumpla una condición
- filter(): Filtra elementos que cumplan una condición

Transformar:

- map(): Transforma cada elemento
- slice(): Copia una porción del array
- concat(): Une arrays
- join(): Convierte array a string
- reverse(): Invierte el array
- sort(): Ordena el array

Pregunta de reflexión 14: ¿Cuál es la diferencia entre map() y forEach()?

Ejemplo

javascript

```javascript
let numeros = [1, 2, 3, 4, 5];
let frutas = ["manzana", "banana", "naranja"];

// Agregar elementos
frutas.push("uva");           // ["manzana", "banana", "naranja", "uva"]
frutas.unshift("pera");       // ["pera", "manzana", "banana", "naranja", "uva"]

// Eliminar elementos
let ultima = frutas.pop();    // Elimina "uva"
let primera = frutas.shift(); // Elimina "pera"

// splice(posición, cantidad a eliminar, elementos a agregar)
frutas.splice(1, 0, "kiwi");  // Agrega "kiwi" en posición 1

// Buscar
console.log(frutas.indexOf("banana"));  // Posición
console.log(frutas.includes("manzana")); // true

// Filter
let pares = numeros.filter(n => n % 2 === 0);
console.log(pares); // [2, 4]

// Map
let cuadrados = numeros.map(n => n * n);
console.log(cuadrados); // [1, 4, 9, 16, 25]

// Find
let mayorQue3 = numeros.find(n => n > 3);
console.log(mayorQue3); // 4

// Join
let texto = frutas.join(", ");
console.log(texto); // "manzana, kiwi, banana, naranja"

// Sort
let desordenado = [3, 1, 4, 1, 5, 9];
desordenado.sort((a, b) => a - b);
console.log(desordenado); // [1, 1, 3, 4, 5, 9]

// Slice
let copia = numeros.slice(1, 4);
console.log(copia); // [2, 3, 4]
```

Ejercicio 4.2

Dado el array productos = [{nombre: "Laptop", precio: 1000}, {nombre: "Mouse", precio: 25}, {nombre: "Teclado", precio: 50}]:

1. Usa filter() para obtener productos menores a $100
2. Usa map() para crear un array solo con los nombres
3. Usa find() para encontrar el producto "Mouse"
4. Usa reduce() para sumar todos los precios
5. Agrega un nuevo producto con push()

javascript

```javascript
let productos = [
    {nombre: "Laptop", precio: 1000},
    {nombre: "Mouse", precio: 25},
    {nombre: "Teclado", precio: 50}
];

// Tu código aquí:
```

TEMA 4.3: INTEGRACIÓN DE OBJETOS EN FORMULARIOS

Teoría

Objetos: Colección de propiedades (pares clave-valor) .

javascript

```javascript
let persona = {
    nombre: "Juan",
    edad: 30,
    ciudad: "Madrid"
};

// Acceder a propiedades
console.log(persona.nombre);  // "Juan"
console.log(persona["edad"]); // 30

// Modificar
persona.edad = 31;

// Agregar
persona.profesion = "Ingeniero";

// Eliminar
delete persona.ciudad;
```

Formularios y objetos: Capturar datos de formularios y convertirlos en objetos.

Pregunta de reflexión 15: ¿Por qué es útil almacenar datos de formularios en objetos?

Ejemplo

html

```html
<!DOCTYPE html>
<html>
<body>
    <form id="miFormulario">
        <input type="text" id="nombre" placeholder="Nombre" required>
        <input type="email" id="email" placeholder="Email" required>
        <input type="number" id="edad" placeholder="Edad" required>
        <select id="ciudad">
            <option value="">Selecciona ciudad</option>
            <option value="madrid">Madrid</option>
            <option value="barcelona">Barcelona</option>
        </select>
        <button type="submit">Guardar</button>
    </form>

    <div id="resultado"></div>

    <script>
        let usuarios = [];

        document.getElementById("miFormulario").addEventListener("submit",
            function(e) {
            e.preventDefault();

            // Crear objeto desde formulario
            let usuario = {
                nombre: document.getElementById("nombre").value,
```

```
            edad: parseInt(document.getElementById("edad").value),
            ciudad: document.getElementById("ciudad").value
        };

        // Validar
        if (usuario.edad < 18) {
            alert("Debes ser mayor de edad");
            return;
        }

        // Guardar
        usuarios.push(usuario);

        // Mostrar
        mostrarUsuarios();

        // Limpiar formulario
        this.reset();
    });

    function mostrarUsuarios() {
        let html = "<h2>Usuarios registrados:</h2><ul>";
        usuarios.forEach(u => {
            html += `<li>${u.nombre} - ${u.email} - ${u.ciudad}</li>`;
        });
        html += "</ul>";
        document.getElementById("resultado").innerHTML = html;
    }
</script>
</body>
</html>
```

Ejercicio 4.3

Crea un formulario de registro de libros que:

1. Capture: título, autor, año, género
2. Valide que todos los campos estén completos
3. Guarde cada libro como un objeto en un array
4. Muestre la lista de libros registrados
5. Calcule y muestre cuántos libros hay por género

```html
<form id="formLibro">
    <input type="text" id="titulo" placeholder="Título" required>
    <input type="text" id="autor" placeholder="Autor" required>
    <input type="number" id="anio" placeholder="Año" required>
    <select id="genero">
        <option value="ficcion">Ficción</option>
        <option value="ciencia">Ciencia</option>
        <option value="historia">Historia</option>
    </select>
    <button type="submit">Agregar Libro</button>
</form>
<div id="listaLibros"></div>

<script>
    let libros = [];

    document.getElementById("formLibro").addEventListener("submit", function(e) {
        e.preventDefault();

        // Tu código aquí:


    });
</script>
```

TEMA 4.4: PROYECTO - FORMULARIO CON VALIDACIÓN

Teoría

Validación de formularios:

- Validación del lado del cliente (JavaScript)
- Expresiones regulares para emails
- Validación de campos requeridos
- Mensajes de error personalizados

Pregunta de reflexión 16: ¿Por qué es importante validar tanto en el cliente como en el servidor?

Ejemplo

```javascript
function validarFormulario(datos) {
    let errores = [];

    // Validar nombre
    if (datos.nombre.trim().length < 3) {
        errores.push("El nombre debe tener al menos 3 caracteres");
    }

    // Validar email con regex
    let emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    if (!emailRegex.test(datos.email)) {
        errores.push("Email inválido");
    }

    // Validar contraseña
    if (datos.password.length < 8) {
        errores.push("La contraseña debe tener al menos 8 caracteres");
    }

    // Validar que las contraseñas coincidan
    if (datos.password !== datos.confirmarPassword) {
        errores.push("Las contraseñas no coinciden");
    }

    return errores;
}
```

Ejercicio 4.4

Crea un formulario de registro completo con:

1. Campos: nombre, email, teléfono, contraseña, confirmar contraseña
2. Validaciones:
    - Nombre: mínimo 3 caracteres
    - Email: formato válido
    - Teléfono: 10 dígitos
    - Contraseña: mínimo 8 caracteres, 1 mayúscula, 1 número

- Confirmar contraseña: debe coincidir

3. Mostrar mensajes de error debajo de cada campo
4. Si todo es válido, mostrar mensaje de éxito

javascript

```javascript
// Tu código aquí:
```