



Entrenamiento – Módulo 3 Semana 1

Hoja de trabajo

El objetivo de esta hoja de trabajo es construir una página web interactiva que te permitirá aplicar tus conocimientos sobre el DOM y las herramientas de almacenamiento en el navegador. Crearás un sistema que:

- Manipule dinámicamente los elementos de una página utilizando JavaScript y el DOM
- Almacene y recupere información del Local Storage y Session Storage
- Valide los datos ingresados por el usuario y proporcione retroalimentación en tiempo real

Problema a resolver: Interfaz Interactiva con Persistencia de Datos

Tu tarea es implementar las siguientes funcionalidades:

1. Crear un formulario HTML que capture el nombre y la edad del usuario.
2. Almacenar esta información en el Local Storage al hacer clic en un botón.
3. Mostrar los datos almacenados en la página cada vez que esta se recargue.
4. Implementar un contador de interacciones del usuario utilizando Session Storage.
5. Incluir un botón para limpiar los datos almacenados en el Local Storage.

Pasos sugeridos para solucionar el problema:

Paso 1: Configuración del proyecto

1. Crea un archivo HTML con un archivo JavaScript asociado.
2. Configura un entorno básico con las siguientes etiquetas en HTML:
 - Formulario (<form>): Campos para nombre y edad.
 - Botones: "Guardar Datos" y "Limpiar Datos".

```
<form id="userForm">
  <label for="name">Nombre:</label>
  <input type="text" id="name" name="name" required>
  <label for="age">Edad:</label>
  <input type="number" id="age" name="age" required>
  <button type="button" id="saveButton">Guardar Datos</button>
  <button type="button" id="clearButton">Limpiar Datos</button>
</form>
<div id="output"></div>
```

Sebastián Agudelo. (2024) – Captura de pantalla del código, paso 1 entrenamiento – módulo 3 – semana 4



Paso 2: Captura y almacenamiento de datos

1. Agrega un evento click al botón "Guardar Datos". Este evento debe:
 - Capturar el nombre y la edad del formulario
 - Validar que ambos campos contengan datos válidos
 - Guardar los datos en Local Storage

```
// Evento para guardar datos en el Local Storage
document.getElementById('saveButton').addEventListener('click', () => {
  const nameInput = document.getElementById('name');
  const ageInput = document.getElementById('age');

  if (!nameInput || !ageInput) {
    console.error('Los elementos del formulario no existen.');
```

```
    return;
  }

  const name = nameInput.value.trim();
  const age = parseInt(ageInput.value);

  if (name && !isNaN(age)) {
    localStorage.setItem('userName', name);
    localStorage.setItem('userAge', age);
    displayData();
  } else {
    alert('Por favor, ingresa un nombre válido y una edad numérica.');
```

```
  }
});
```

Sebastián Agudelo. (2024) – Captura de pantalla del código, paso 2 entrenamiento – módulo 3 – semana 4

Paso 3: Recuperación y visualización de datos

1. Al cargar la página, recupera los datos de Local Storage y muéstralos en el elemento `<div id="output">`.
2. Si no hay datos en Local Storage, muestra un mensaje indicando que no hay información almacenada.

```
// Función para mostrar los datos almacenados
function displayData() {
  const name = localStorage.getItem('userName');
  const age = localStorage.getItem('userAge');
  const outputDiv = document.getElementById('output');
  if (name && age) {
    outputDiv.textContent = `Hola ${name}, tienes ${age} años.`;
  } else {
    outputDiv.textContent = 'No hay datos almacenados.';
  }
}

// Al cargar la página, mostrar los datos almacenados
window.onload = displayData;
```

Sebastián Agudelo. (2024) – Captura de pantalla del código, paso 3 entrenamiento – módulo 3 – semana 4



Paso 4: Contador con Session Storage

1. Implementa un contador que registre cuántas veces el usuario ha interactuado con la página en la sesión actual.
2. Muestra el número de interacciones en tiempo real.

```
// Inicializar contador de interacciones en Session Storage
if (!sessionStorage.getItem('interactionCount')) {
  sessionStorage.setItem('interactionCount', 0);
}

// Actualizar contador de interacciones
function updateInteractionCount() {
  let count = parseInt(sessionStorage.getItem('interactionCount'));
  count++;
  sessionStorage.setItem('interactionCount', count);
  console.log(`Interacciones en esta sesión: ${count}`);
}

// Asignar eventos al contador
document.getElementById('saveButton').addEventListener('click', updateInteractionCount);
document.getElementById('clearButton').addEventListener('click', updateInteractionCount);
```

Sebastián Agudelo. (2024) – Captura de pantalla del código, paso 4 entrenamiento – módulo 3 – semana 4

Paso 5: Limpieza de datos

1. Al hacer clic en "Limpiar Datos", elimina toda la información del Local Storage y actualiza el contenido de la página.

```
// Evento para limpiar los datos del Local Storage
document.getElementById('clearButton').addEventListener('click', () => {
  localStorage.clear();
  displayData();
});
```

Sebastián Agudelo. (2024) – Captura de pantalla del código, paso 5 entrenamiento – módulo 3 – semana 4

Resultado esperado:

1. El usuario puede guardar su nombre y edad, y estos datos se muestran al recargar la página.
2. El contador de interacciones muestra cuántas veces el usuario ha interactuado en la sesión.
3. Un botón limpia toda la información almacenada en el Local Storage.

Recomendaciones finales:

- **Documenta tu código:** usa comentarios para describir cada sección.
- **Sigue buenas prácticas:** usa nombres de variables descriptivos y el formato camelCase.
- **Explora más:** Intenta mejorar el programa con validaciones adicionales o mensajes personalizados.