Javascript avanzado

Módulo 1 - Unidad 3

Formulario interactivo con validación en clases

Objetivos

Practicar:

- Uso de setTimeout o Promise para simular asincronía.
- Creación de **clases** con propiedades y métodos.
- Manipulación de datos con **map**, **filter** y **find**.

Consigna

1. **Definir la clase**
    - Crear una clase Tarea con:
        - Propiedades: id, titulo, completada (booleano).
        - Método toggleEstado() que cambie completada a su valor contrario.

- Crear una clase GestorTareas que:
  - Contenga un array de tareas.
  - Método agregarTarea(titulo) → crea una nueva tarea y la agrega al array.
  - Método listarTareas() → usa forEach para mostrar todas las tareas en consola.
  - Método buscarPorTitulo(titulo) → usa find para devolver la tarea que coincida.
  - Método listarCompletadas() → usa filter para devolver las tareas con completada en true.

2. **Simulación asíncrona**
   - Crear una función cargarTareas() que devuelva una **promesa** y simule la carga inicial de datos con setTimeout (2 segundos).
   - Dentro del resolve, devolver un array con 3 tareas iniciales.
3. **Flujo del programa**
   - Usar async/await para:
     - Esperar a cargarTareas() y asignar los resultados al GestorTareas.
     - Mostrar un mensaje en consola: "Tareas cargadas correctamente".

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

        - Listar las tareas.

    - Agregar una nueva tarea y luego mostrar la lista actualizada.

    - Filtrar y mostrar las tareas completadas.

4. **Extra (opcional)**

    - Usar map para crear un nuevo array con los títulos de las tareas y mostrarlo en consola.

    - Aplicar Promise.all para simular la carga de tareas y otra operación en paralelo (ej. cargar usuarios).

Formato de presentación

La entrega debe hacerse en un **repositorio en GitHub** que incluya:

- **Archivo JavaScript principal** con:

    - Clase Tarea y clase GestorTareas.

    - Función cargarTareas() con promesa y setTimeout.

    - Uso de async/await para flujo asincrónico.

    - Ejemplo de agregar, listar y filtrar tareas.

    - (Opcional) Uso de map y Promise.all.

- **README.md** con:
    - Breve descripción y objetivos.
    - Instrucciones de instalación/ejecución.
    - Capturas de pantalla de la consola.
    - Créditos del autor.
    - Citación de fuentes.

Criterios de evaluación

- Implementación correcta de **clases y métodos** en JavaScript.
- Uso adecuado de **propiedades y métodos de array** (forEach, find, filter, map).
- Simulación de asincronía con **Promise y setTimeout**.
- Uso de **async/await** para manejar flujos asincrónicos.
- Ejecución correcta del flujo de tareas: carga inicial, listado, agregado, filtrado.
- (Opcional) Uso de Promise.all para manejar múltiples promesas en paralelo.
- Organización clara y prolija del código en el repositorio.
- README con instrucciones claras y bibliografía.

Bibliografía utilizada y sugerida

Libros y otros manuscritos

Flanagan, D. *JavaScript: The Definitive Guide. Master the World's Most-Used Programming Language*. 7ª ed. Estados Unidos: O'Reilly Media; 2020. https://share.google/NWtI0mj2jOuHpwbuu

Freeman, E. y Robson E. *Head First. JavaScript Programming*. 1ª ed. Estados Unidos: O'Reilly Media; 2014. https://nibmehub.com/opac-service/pdf/read/Head%20First%20JavaScript%20Programming%20_%20a%20learner's%20guide%20to%20JavaScript%20programming-compressed.pdf

Artículos y documentación en línea

MDN Web Docs. (s.f.-a). *Classes*. Mozilla Corporation. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Classes

MDN Web Docs. (s.f.-b). *Promise*. Mozilla Corporation. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise

MDN Web Docs. (s.f.-c). *Array methods*. Mozilla Corporation. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array