

Laboratorio Virtual: Integración de JSON, Estructuras de Datos y Algoritmos en el Sitio Web del Hospital

Contexto:

En este laboratorio, los estudiantes implementarán **JSON** para manipular y transferir datos en el sitio web del hospital, además de trabajar con **estructuras de datos** avanzadas y crear **algoritmos** para optimizar las operaciones con estos datos. Los estudiantes también practicarán la clonación, fusión y recorrido de objetos, y la implementación de algoritmos como búsqueda y ordenamiento en JavaScript.

Duración: **2 horas**

Requisitos:

1. Manejo de Objetos JSON (2 puntos)

- Implementa un **objeto JSON** que represente el **listado de doctores** en el hospital (nombre, especialidad, años de experiencia, disponibilidad).
 - Usa **objetos anidados** para organizar información adicional de cada doctor, como horarios disponibles y contacto.
 - Utiliza **destructuring** para acceder a las propiedades de estos objetos y mostrar la información de un doctor específico en la consola y en la interfaz web.

2. Operaciones con JSON (1.5 puntos)

- Realiza las siguientes operaciones con los objetos JSON:
 - **Clonación**: Crea una copia de un objeto JSON y modifícalo sin afectar el original.
 - **Merge**: Fusiona dos objetos JSON. Por ejemplo, puedes fusionar la información de doctores con la lista de servicios médicos disponibles.
 - **Recorrido y stringify**: Recorre los objetos JSON para mostrar en el navegador la lista de doctores disponibles. Convierte el objeto a una cadena JSON usando **JSON.stringify()** y muestra el resultado en la consola.

3. Implementación de Estructuras de Datos (2 puntos)

- Implementa **estructuras de datos** en el sitio web del hospital para gestionar mejor la información de los doctores:
 - **Arreglos**: Utiliza un arreglo para almacenar la lista de doctores. Implementa operaciones como agregar, eliminar y buscar doctores dentro del arreglo.
 - **Pilas**: Implementa una pila para gestionar las citas de los pacientes (última cita agendada, próxima cita a atender, etc.).
 - **Colas**: Crea una cola para simular el orden de llegada de los pacientes en la página de contacto.

4. Programación de Algoritmos (1.5 puntos)

- Implementa un **algoritmo de búsqueda** que permita encontrar un doctor específico dentro del arreglo de doctores.
 - Implementa un **algoritmo de ordenamiento** para organizar la lista de doctores por años de experiencia.
 - Explica en el archivo README la **complejidad de los algoritmos** utilizados, aplicando conceptos de **Big-O** y **complejidad ciclomática**.
-

Herramientas a Utilizar:

- **JavaScript** para manipular objetos JSON, implementar estructuras de datos y desarrollar algoritmos.
 - **README**: Actualiza el archivo README con:
 - Explicación de cómo se implementaron los objetos JSON y las operaciones realizadas (clonación, merge, recorrido).
 - Explicación de las estructuras de datos implementadas (arreglos, pilas, colas) y su utilidad en el proyecto.
 - Descripción de los algoritmos implementados y su complejidad.
-

Entrega:

- **Formato de entrega**:
 - Opción 1: Enviar un **enlace al repositorio de GitHub** con el proyecto actualizado, incluyendo los archivos HTML, CSS, JS y README.
 - Opción 2: Entregar un archivo **ZIP comprimido** con el proyecto completo (HTML, CSS, JS, README, etc.).