

ABPro: Desarrollo Completo del Sistema del Hospital con JavaScript Avanzado

Contexto:

El objetivo de este **Aprendizaje Basado en Proyectos** es que los estudiantes integren todos los conceptos avanzados de **JavaScript** para mejorar y completar el **sitio web del hospital**. Deberán demostrar habilidades en la manipulación de datos, el uso de algoritmos y estructuras de datos, y la aplicación de programación funcional, asíncrona y orientada a objetos.

Duración: 4 horas

Requisitos:

1. Manipulación de Datos con JSON y Simulación de API REST (1.5 puntos)

- Utiliza **JSON** para manejar la información de **doctores**, **servicios médicos**, y **citas**.
 - Simula la obtención de datos desde una **API REST** y carga dinámicamente la información en la interfaz.
 - Realiza operaciones de **clonación**, **fusión (merge)**, y **recorrido** de los datos JSON.

2. Implementación de Algoritmos y Estructuras de Datos (1.5 puntos)

- Implementa **algoritmos de búsqueda** y **ordenamiento** para gestionar los datos de los doctores.
 - Ejemplo: búsqueda por especialidad y ordenamiento por años de experiencia.
 - Utiliza **estructuras de datos** como **listas**, **pilas** o **colas** para gestionar las citas de los pacientes, implementando operaciones de agregar, eliminar y gestionar citas.

3. Programación Funcional (1 punto)

- Implementa funciones avanzadas de **JavaScript funcional**, como **currying**, **composición de funciones**, y **recursión** para el manejo de datos de los doctores y pacientes.

- Ejemplo: uso de currying para calcular costos de servicios médicos basados en cantidad de consultas.

4. Programación Orientada a Objetos (2 puntos)

- Utiliza **POO** para estructurar el proyecto:
 - Crea una clase **Doctor** con subclases como **Cirujano** o **Pediatra**, aplicando **encapsulación, herencia, y polimorfismo**.
 - Implementa métodos para calcular costos de consulta, gestionar disponibilidad, etc.

5. Programación Asíncrona y Eventos (1 punto)

- Integra **programación asíncrona** utilizando **async/await** y **promesas** para la simulación de un registro de citas.
- Implementa **event listeners** para capturar eventos del usuario, como la confirmación de una cita o la solicitud de información de un doctor.

Herramientas a Utilizar:

- **JavaScript avanzado** para manipular JSON, implementar algoritmos y estructuras de datos, aplicar programación funcional, asíncrona y orientada a objetos.
- **README**: Actualiza el archivo **README** con:
 - Descripción de los datos manipulados con JSON y cómo se cargan en la interfaz.
 - Explicación de los algoritmos y estructuras de datos utilizados.
 - Explicación de las funciones y clases creadas, y el manejo de eventos.
 - Instrucciones para ejecutar y probar el proyecto.

Entrega:

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