

# Guía Interactiva de Museos - Ejercitación de Desarrollo de Software

## Objetivo

El objetivo de esta ejercitación es continuar con el desarrollo de la “**Guía Interactiva de Museos**” y que permita el agregado, eliminación y actualización de un museo. La aplicación mostrará una lista de museos con detalles como ubicación, exposiciones actuales, horarios de visita y precios de entrada. El repositorio original está en:

[https://labsys.frc.utn.edu.ar/gitlab/desarrollo-de-software1/proyectos2024/3k4/repos-guias/frontend\\_backend\\_museos](https://labsys.frc.utn.edu.ar/gitlab/desarrollo-de-software1/proyectos2024/3k4/repos-guias/frontend_backend_museos)

## Descripción de la Ejercitación

Los estudiantes deberán continuar la ejercitación y desarrollar los componentes de frontend y backend. Deberán completar ciertas funciones para que la aplicación funcione correctamente. El frontend debe usar HTML y JavaScript para mostrar una lista de museos. El backend, desarrollado con Node.js, servirá estos museos desde una base de datos en memoria SQLite gestionada por Sequelize.

### Requisitos del Backend

- Completar la implementación de la ruta **POST** en el backend para ingresar un nuevo museo a la base de datos SQLite utilizando Sequelize.
- Completar la implementación de la ruta **DELETE** en el backend para eliminar museos desde la base de datos SQLite utilizando Sequelize.

### Requisitos del Frontend

- Completar las funciones **eliminarMuseos**, **insertarMuseos** en JavaScript para obtener los museos desde el backend usando la API Fetch y renderizarlos dinámicamente en la página web.

## Instrucciones

### 1. Configuración del Backend:

- Se proporcionará el código base para la aplicación Node.js.
- Los estudiantes deben completar la implementación de las rutas:

- i. `app.post('/museos', async (req, res) => {...})` para que permita ingresar museos a la base de datos.
- ii. `app.delete('/museos', async (req, res) => {...})` para que permita eliminar museos a la base de datos.

## 2. Configuración del Frontend:

- Se proporcionará un archivo HTML básico y el código JavaScript inicial.
- Los estudiantes deben completar la función `eliminarMuseos`, `agregar` en el código JavaScript para realizar una solicitud fetch al backend, procesar la respuesta.

## 3. Pruebas:

- Verificar que las funciones `eliminarMuseos`, `agregar` obtiene y muestra correctamente los museos del backend.
- Asegurarse de que las rutas del backend se ejecute correctamente que maneje los posibles errores.

# Criterios de Evaluación

- Funcionalidad: La aplicación funciona como se describe, completando correctamente las partes del código requeridas.
- Calidad del Código: El código es limpio, bien organizado y adecuadamente comentado.
- Diseño: El frontend muestra los museos de manera clara y organizada.

# Pruebas y Ejecución:

Para probar y ejecutar este servidor ejecuta los siguientes comandos en tu terminal para instalar las dependencias y arrancar el servidor:

```
npm install express sequelize sqlite3 cors
node app.js
```

Esto iniciará el servidor en `http://localhost:3000`, y podrás acceder a los datos de los museos accediendo a `http://localhost:3000/museos` desde un navegador o utilizando herramientas como Postman.\