UNIVERSIDAD NACIONAL DE SAN AGUSTÍN FACULTAD DE INGENIERIA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS



CURSO: PROGRAMACION WEB 2

DOCENTE:CORRALES DELGADO CARLOS JOSE

TEMA: SQLite

PRESENTADO POR:SANTA CRUZ VILLA DIEGO SEBASTIAN

GIT HUB:

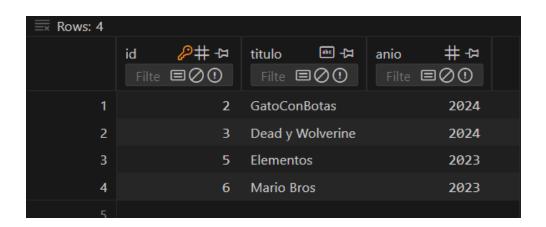
https://github.com/DiegoSantaC/LaboratoriosPW2/tree/TareasTeoriaPW2

RESULTADOS: Tabla de Peliculas dinamica

Películas



- GatoConBotas (2024)
- Dead y Wolverine (2024)
- Elementos (2023) X
- Mario Bros (2023)



ERRORES:

1) Falla de nombres

```
Expression
GET <a href="http://localhost:3000/script.js">http://localhost:3000/script.js</a> net::ERR_ABORTED <a href="localhost/:18">localhost/:18</a> (NOT FOUND)
```

Esto es debido a una falla de nombres de los archivos al conectar con el servidos python (script-> scrips)

```
20 @app.route("/script.js")
21 def js():
22     return send_from_directory('.', 'scrips.js')
```

2) Error de conexion

Se crea la conexión fuera de la función lo cual causa error en los hilos.

CODIGO:

a)Index.html

b) Script.js(AJAX)

```
function cargarPeliculas() {
         fetch("/peliculas")
             .then(res => res.json())
             .then(data => {
                 const lista = document.getElementById("lista");
                 lista.innerHTML = "";
6
                 data.forEach(p => {
                      const li = document.createElement("li");
                      li.textContent = `${p.titulo} (${p.anio}) `;
                      const btnEliminar = document.createElement("button");
                      btnEliminar.textContent = "X";
                      btnEliminar.onclick = () => eliminarPelicula(p.id);
                      li.appendChild(btnEliminar);
                      lista.appendChild(li);
                  });
             });
     function agregarPelicula(evento) {
         evento.preventDefault();
         const titulo = document.getElementById("titulo").value;
         const anio = document.getElementById("anio").value;
         fetch("/peliculas", {
             method: "POST",
             headers: { "Content-Type": "application/json" },
             body: JSON.stringify({ titulo, anio })
         .then(() => {
             document.getElementById("formulario").reset();
             cargarPeliculas();
     function eliminarPelicula(id) {
         fetch(`/peliculas/${id}`, { method: "DELETE" })
             .then(() => cargarPeliculas());
     document.getElementById("formulario").addEventListener("submit", agregarPelicula);
     window.onload = cargarPeliculas;
42
```

c) Servidor.py

```
@app.route("/peliculas", methods=["GET"])
     def obtener_peliculas():
        with sqlite3.connect("imdb.db") as conn:
        peliculas = conn.execute("SELECT * FROM Pelicula").fetchall()
return jsonify([{"id": p[0], "titulo": p[1], "anio": p[2]} for p in peliculas])
    @app.route("/peliculas", methods=["POST"])
    def agregar_pelicula():
        datos = request.get_json()
        titulo = datos.get("titulo")
        anio = datos.get("anio")
        with sqlite3.connect("imdb.db") as conn:
            conn.execute("INSERT INTO Pelicula (titulo, anio) VALUES (?, ?)", (titulo, anio))
        return jsonify({"mensaje": "Película agregada correctamente"})
38
    @app.route("/peliculas/<int:id>", methods=["DELETE"])
    def eliminar_pelicula(id):
        with sqlite3.connect("imdb.db") as conn:
            conn.execute("DELETE FROM Pelicula WHERE id = ?", (id,))
        return jsonify({"mensaje": "Película eliminada"})
    if __name__ == "__main__":
        inicializar_bd()
        app.run(debug=True, host="localhost", port=3000)
     from flask import Flask, request, jsonify, send_from_directory
     import sqlite3
     app = Flask( name )
     def inicializar bd():
          with sqlite3.connect("imdb.db") as conn:
               conn.execute("""
                    CREATE TABLE IF NOT EXISTS Pelicula (
                          id INTEGER PRIMARY KEY AUTOINCREMENT,
                          titulo TEXT NOT NULL,
                          anio INTEGER NOT NULL
     @app.route("/")
     def index():
          return send from directory('.', 'index.html')
     @app.route("/script.js")
     def js():
          return send from directory('.', 'script.js')
```

CONCLUSIONES:

Este proyecto me sirvio para enseñarme a cómo crear una página web que conecta con un programa en Python para mostrar información de una base de datos. Aprendi a guardar los datos en una base pequeña llamada SQLite y a hacer que el servidor entregue esos datos cuando el usuario los pide. También pude ver cómo el navegador puede pedir esos datos usando JavaScript y mostrarlos en la página. Durante el desarrollo, entendi la importancia de organizar bien el código, abrir y cerrar la base de datos correctamente, y manejar errores comunes para que todo funcione sin problemas. Así, el proyecto sirve para entender cómo trabajan juntos el servidor, la base de datos y la página web para que el usuario vea la información que quiere.