

Trabajo Práctico Integrador.

IA 3.5 REDES DE DATOS

Tecnicatura Universitaria en Inteligencia Artificial.



López Ceratto, Julieta: L-3311/1.

Slepoy, David: S-5782/7.

24/06/2024.

INTRODUCCIÓN.....	2
DESCRIPCIÓN DEL ENTORNO DE TRABAJO DEL SERVIDOR.....	2
DESCRIPCIÓN DEL ENTORNO DE TRABAJO DEL DEL CLIENTE.....	2
DESCRIPCIÓN DE LA BASE DE DATOS ELEGIDA.....	2
PROCEDIMIENTO.....	3
INSTRUCCIONES DE USO DEL CLIENTE.....	5
INSTRUCCIONES DE EJECUCIÓN DEL SERVIDOR.....	5
HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO DEL TRABAJO.....	6
REFERENCIAS.....	6
ANEXO A: REPOSITORIO GITHUB.....	7
APÉNDICE A: Código servidor.....	8
APÉNDICE B: CÓDIGO CLIENTE.....	15
JavaScript.....	15

INTRODUCCIÓN.

Este trabajo desarrolla una API con parte cliente y servidor. Se eligió la base de datos "Películas Americanas", la cual posee un registro de películas norteamericanas con diversos datos de las mismas.

Se trabajó en un entorno físico, con sistema operativo Windows.

DESCRIPCIÓN DEL ENTORNO DE TRABAJO DEL SERVIDOR.

El entorno de trabajo fue desarrollado mediante Visual Studio Code, utilizando el lenguaje de python; fue levantado con Uvicorn.

Se configuraron los accesos para que permita accesos de todos los hosts, es decir, no se utilizaron credenciales de acceso.

Los problemas encontrados fueron en relación a la carga y modificación del archivo movies.json, lo cual se solucionó abriendo el archivo dentro de cada método y cerrándolo antes de realizar cualquier return en el mismo; por otro lado, aquellos métodos que realizan modificaciones en los archivos, se solucionó el error agregando también seek(0) que va al inicio del archivo y truncate() que trunca el tamaño del archivo para que sea compatible con las modificaciones hechas.

DESCRIPCIÓN DEL ENTORNO DE TRABAJO DEL DEL CLIENTE.

El entorno del cliente fue desarrollado a través Visual Studio Code, utilizando JavaScript para realizar las consultas a la api. Dicho JavaScript está vinculado a un archivo html, el cual representa un sitio web a través del cual interactúa el usuario. La página web tiene un formato dado a través de un css.

Un inconveniente presentado fue en cuanto a los permisos de acceso al servidor, lo cual se solucionó configurando los permisos al mismo en el código del servidor.

DESCRIPCIÓN DE LA BASE DE DATOS ELEGIDA.

La base de datos elegida tiene formato .json, "Películas Americanas", contiene más de 20.000 películas de origen norteamericano. Cada película tiene como máximo la siguiente información:

- title: str.
- year: int.
- cast: list (str).
- genres: list (str).
- href: str.
- extract: str.
- thumbnail: link (str).
- thumbnail_width: int.
- thumbnail_height: int.

Como mínimo, una película tiene su nombre y año de lanzamiento.

PROCEDIMIENTO.

Para comunicarse con la API hay 4 métodos.

- Buscar película (GET).
- Agregar película (POST).
- Modificar película (PUT)
- Eliminar película (DELETE).

El usuario realiza la consulta a través de un sitio web mediante formularios.

1. Buscar película (GET).

El usuario debe completar el criterio que tenga para buscar una película. El método permite buscar película(s) por:

- a. Nombre de la película: devuelve todas las películas con dicho nombre.
- b. Año de lanzamiento: devuelve todas las películas con dicho año de lanzamiento.
- c. Nombre y año de lanzamiento: devuelve todas las películas con dicho nombre y año de lanzamiento.

Obs: siempre 1 es obligatorio, no se puede enviar el formulario vacío. El nombre es sensible a minúsculas y mayúsculas.

2. Agregar película (POST).

Agrega una película nueva. El usuario debe completar como mínimo nombre de la película y año de lanzamiento. Luego, tiene otras opciones tales como:

- Cast (reparto).
- Genres (géneros).
- Extract (resumen).
- Href (link de referencia).
- Thumbnail.
- Thumbnail width.
- Thumbnail height.

3. Modificar película (PUT).

El usuario puede realizar modificaciones a películas ya existentes en la base de datos. Debe completar como obligatorio: nombre de película a modificar y año de lanzamiento de dicha película a modificar. Luego, debe completar los campos que desea modificar. Si en el campo no se escribe nada, lo deja tal

cual está. Si escribe la palabra "borrar" en los datos no numéricos o 0 en los datos numéricos, borra la información existente.

4. Eliminar película (DELETE).

El usuario puede eliminar una película existente en la base de datos. Debe completar como obligatorio los dos campos del formulario "nombre de la película" y "año de lanzamiento".

INSTRUCCIONES DE USO DEL CLIENTE.

Los requerimientos necesarios para hacer uso de este sitio web son:

1. Navegador de preferencia.

Se ejecuta abriendo el archivo index.html y luego eligiendo uno de los cuatro métodos que quiera realizar, que abren un formulario que deben completar con las especificaciones aclaradas en el apartado anterior.

Es necesario que el técnico controle la dirección IP dentro del archivo main.js coincida con la dirección IP del servidor. Se estima que esta dirección no cambie por lo que sería un proceso de única vez.

INSTRUCCIONES DE EJECUCIÓN DEL SERVIDOR.

El servidor, como se mencionó, fue levantado en Uvicorn. Es importante para su funcionamiento en host de otros dispositivos dentro de la misma red, que se levante utilizando el siguiente comando:

```
Uvicorn TP_REDES:app --host 0.0.0.0 --reload
```

Siendo el `--reload` no necesario, pero el `--host` imprescindible.

Además, si no funcionan las consultas desde otro host luego del comando anterior, se debe probar desactivando el firewall.

HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO DEL TRABAJO.

- Uvicorn.
- Python.
- Fastapi.
- JavaScript.
- Librerías: fastapi, json, CORSMiddleware.
- Bootstrap.
- Html.
- Css.

En caso de no tener Fastapi, Uvicorn, Python ejecutar previamente su instalación.

```
pip install fastapi  
pip install Uvicorn  
pip install Python
```

Librería Github para descarga y vista del trabajo:

REFERENCIAS.

1. JavaScript: <https://lenguajejs.com/javascript/>
2. Bootstrap: <https://getbootstrap.com/>
3. CSS: <https://developer.mozilla.org/en-US/docs/Web/CSS>
4. HTML: <https://developer.mozilla.org/en-US/docs/Web/HTML>

Universidad Nacional de Rosario.

Facultad de Ciencias Exactas, Ingeniería y Agrimensura.

ANEXO A: REPOSITORIO GITHUB

[julilc/tp_redes: Trabajo practico final de asignatura 'Redes de datos' de la Tec. Universitaria en Inteligencia Artificial. Desarrollo de una API con funciones GET, POST, UPDATE, DELETE \(github.com\)](#)

APÉNDICE A: Código servidor.

```
from fastapi import FastAPI

import json

from fastapi.middleware.cors import CORSMiddleware

app = FastAPI()

ruta = r'movies.json'

# Configurar CORS

app.add_middleware(

    CORSMiddleware,

    allow_origins=["*"], # Permitir todas las solicitudes de origen (en
producción, deberías limitarlo a los orígenes específicos que necesitas)

    allow_credentials=True,

    allow_methods=["GET", "POST", "PUT", "DELETE"],

    allow_headers=["*"],

)

@app.get("/my-first-api")

def buscar(nombre = None, año = None):

    with open(ruta,'r+',encoding='ANSI') as file:

        data = json.load(file)
```

```
peliculas = []

if año is None and nombre is not None:

    for pelicula in data:

        if pelicula['title'] == nombre:

            peliculas.append(pelicula)

    if len(peliculas) == 0:

        file.close()

        return('No hubo coincidencias')

    file.close()

    return peliculas

if nombre is None and año is not None:

    for pelicula in data:

        if pelicula['year'] == int(año):

            peliculas.append(pelicula)

    if len(peliculas) == 0:

        file.close()

        return('No hubo coincidencias')

    file.close()

    return peliculas

if nombre is not None and año is not None:

    for pelicula in data:

        if pelicula['year'] == int(año) and pelicula['title'] ==
nombre:

            peliculas.append(pelicula)
```

```
        if len(peliculas) == 0:

            file.close()

            return('No hubo coincidencias')

        file.close()

        return peliculas

    file.close()

    return 'No se encontró'

@app.post("/my-first-api")

def agregar_peli(nombre:str,año:int,cast=None,genres=None, href=None,
extract=None,thumbnail=None, thumbnail_width=None, thumbnail_height=None
):

    pelicula_agregar = buscar(nombre, año)

    print(type(pelicula_agregar))

    if type(pelicula_agregar) != list:

        with open(ruta,'r+', encoding='ANSI') as file:

            data = json.load(file)

            if cast != None:

                cast = cast.split(sep = ',')

            if genres != None:

                genres = genres.split(sep = ',')

            nueva_pelicula = {

                'title' : nombre,

                'year' : año,
```

```
        'cast' : cast,

        'genres' : genres,

        'href' : href,

        'extract':extract,

        'thumbnail' : thumbnail,

        'thumbnail_width':thumbnail_width,

        'thumbnail_height':thumbnail_height

    }

    #va al inicio del archivo

    file.seek(0)

    data.append(nueva_pelicula)

    json.dump(data, file, indent=4, ensure_ascii=False)

    #trunca el tamaño del archivo

    file.truncate()

    file.close()

    return('Pelicula agregada')

else:

    return('Ya existe la pelicula')
```

```
@app.put("/my-first-api")

def actualizar(nombre: str, año: int, title = None, year =
None, cast=None, genres=None, href=None, extract=None, thumbnail=None,
thumbnail_width=None, thumbnail_height=None ):

    if cast != None:

        cast = cast.split(sep = ',')

    if genres != None:

        genres = genres.split(sep = ',')

    variables_actualizar = {'title':title, 'year':year, 'cast':cast,
'genres':genres, 'href':href, 'extract':extract, 'thumbnail':thumbnail,
'thumbnail_width':thumbnail_width, 'thumbnail_height':thumbnail_height}

    #Usamos la funcion buscar para ver si existe la película

    pelicula_actualizar = buscar(nombre, año)

    if pelicula_actualizar != str:

        #Si no es un str, entonces existe la película, por lo que
realizamos la actualización.

        with open(ruta, 'r+', encoding='ANSI') as file:

            data = json.load(file)

            for pelicula in data:

                if pelicula['title'] == nombre and pelicula['year'] ==
int(año):

                    for nombre_variable, valor_variable in
variables_actualizar.items():
```

```
        #si la variable tiene un valor para actualizar la
actualiza
        if valor_variable != None and valor_variable !=
'borrar' and valor_variable != 0:

            pelicula[nombre_variable] = valor_variable

        #si la variable tiene un valor para borrar la
borra
        elif valor_variable == 'borrar' or valor_variable
== 0:

            if nombre_variable == 'year':

                pelicula[nombre_variable] = 0

            else:

                pelicula[nombre_variable] = ''

        file.seek(0)

        json.dump(data, file, indent=4, ensure_ascii=False)

        #trunca el tamaño del archivo

        file.truncate()

    file.close()

    return 'variable actualizada'

else:

    return 'pelicula no encontrada'
```

```
@app.delete("/my-first-api")

def borrar(nombre, año):

    with open(ruta, 'r+', encoding='ANSI') as file:

        data = json.load(file)

        for pelicula in data:

            if pelicula['title'] == nombre and pelicula['year'] ==
int(año):

                data.remove(pelicula)

                with open(ruta, 'w', encoding='ANSI') as file:

                    json.dump(data, file, indent=4, ensure_ascii=False)

                    file.close()

                return 'Pelicula borrada'

    return 'No se encontro la pelicula'
```

APÉNDICE B: CÓDIGO CLIENTE.

JavaScript

```
// main.js

document.addEventListener('DOMContentLoaded', function() {

    const buscarPeliculaLink =
document.querySelector('a[href="#Buscar_pelicula"]');

    const agregarPeliculaLink =
document.querySelector('a[href="#Agregar_pelicula"]');

    const modificarPeliculaLink =
document.querySelector('a[href="#Modificar_pelicula"]');

    const eliminarPeliculaLink =
document.querySelector('a[href="#Eliminar_pelicula"]');

    const inicioLink = document.querySelector('a[href = "#"]')

    const agregarPeliculaSection =
document.getElementById('agregar-pelicula-section');

    const buscarPeliculaSection =
document.getElementById('buscar-pelicula-section');

    const modificarPeliculaSection =
document.getElementById('modificar-pelicula-section');

    const eliminarPeliculaSection =
document.getElementById('eliminar-pelicula-section');

    inicioLink.addEventListener('click', function(event) {

        event.preventDefault();

        buscarPeliculaSection.style.display = 'none';
```



```
modificarPeliculaSection.style.display = 'none';

agregarPeliculaSection.style.display = 'none';

eliminarPeliculaSection.style.display = 'none';

});

buscarPeliculaLink.addEventListener('click', function(event) {

    event.preventDefault();

    buscarPeliculaSection.style.display = 'block';

    modificarPeliculaSection.style.display = 'none';

    agregarPeliculaSection.style.display = 'none';

    eliminarPeliculaSection.style.display = 'none';

});

agregarPeliculaLink.addEventListener('click', function(event) {

    event.preventDefault();

    buscarPeliculaSection.style.display = 'none';

    modificarPeliculaSection.style.display = 'none';

    agregarPeliculaSection.style.display = 'block';

    eliminarPeliculaSection.style.display = 'none';

});

modificarPeliculaLink.addEventListener('click', function(event) {

    event.preventDefault();
```

```
        buscarPeliculaSection.style.display = 'none';

        modificarPeliculaSection.style.display = 'block';

        agregarPeliculaSection.style.display = 'none';

        eliminarPeliculaSection.style.display = 'none';

    });

    eliminarPeliculaLink.addEventListener('click', function(event) {

        event.preventDefault();

        buscarPeliculaSection.style.display = 'none';

        modificarPeliculaSection.style.display = 'none';

        agregarPeliculaSection.style.display = 'none';

        eliminarPeliculaSection.style.display = 'block';

    });

    // INTERFAZ BUSCAR

    const buscarPeliculaForm =
document.getElementById('buscar-pelicula-form');

    buscarPeliculaForm.addEventListener('submit', function(event) {

        event.preventDefault();

        // Obtén los valores del formulario

        const nombre_buscar =
document.getElementById("nombre-pelicula").value;
```

```
const año = document.getElementById("año-pelicula").value;

let url = ``;

// Verificar si tanto el nombre como el año tienen un valor
distinto de null

if (nombre_buscar !== '' && año !== '') {

    url +=
`http://192.168.84.121:8000/my-first-api?nombre=${nombre_buscar}&año=${año}
`;

} else if (nombre_buscar === '' && año !== '') {

    // Verificar si el nombre es '' pero el año no lo es

    url += `http://192.168.84.121:8000/my-first-api?año=${año}`;

} else if (nombre_buscar !== '' && año === '') {

    // Verificar si el nombre no es '' pero el año lo es

    url +=
`http://192.168.84.121:8000/my-first-api?nombre=${nombre_buscar}`;

}

console.log(url)

// Realiza la solicitud GET utilizando fetch

fetch(url)

.then(response => response.json())

.then(data => {

    // Obtener una referencia al elemento del contenido del cartel

    const infoContent = document.getElementById('info-content');
```

```
// Limpiar cualquier contenido previo del cartel
infoContent.innerHTML = '';

// Iterar sobre cada película en los datos devueltos por la
API

console.log(data)

if (Array.isArray(data)) {
    data.forEach(pelicula => {
        // Crear un nuevo elemento de párrafo para cada
        película

        const peliculaInfo = document.createElement('p');

        // Iterar sobre cada par clave-valor de la película
        for (const [key, value] of Object.entries(pelicula)) {
            // Crear un nodo de texto con la clave y el valor
            de la película

            const keyValueText =
            document.createTextNode(`${key}: ${value}`);

            // Agregar el nodo de texto al elemento de párrafo
            peliculaInfo.appendChild(keyValueText);

            // Agregar un elemento de salto de línea después
            de cada par clave-valor
```

```
peliculaInfo.appendChild(document.createElement('br'));

    }

    // Agregar un elemento de salto de línea adicional
entre cada película

peliculaInfo.appendChild(document.createElement('br'));

    // Agregar el elemento de párrafo al contenido del
cartel

    infoContent.appendChild(peliculaInfo);

    });

}

else {

    infoContent.textContent = data

};

    // Mostrar el cartel

    const infoBox = document.getElementById('info-box');

    infoBox.style.display = 'block';

})

.catch(error => {

    // Manejar cualquier error que pueda ocurrir
```

```
        console.error('Hubo un error:', error);

    });

    // Agregar un event listener al botón de cerrar para ocultar
    el cartel

    document.getElementById('close-button').addEventListener('click',
function() {

        const infoBox = document.getElementById('info-box');

        infoBox.style.display = 'none';

    });

});

//INTERFAZ AGREGAR

const agregarPeliculaForm =
document.getElementById('agregar-pelicula-form');

agregarPeliculaForm.addEventListener('submit', function(event){

    event.preventDefault();

    // Obtén los valores del formulario

    const nombre_agregar =
document.getElementById('nombre-pelicula-agregar').value;

    const año_agregar =
document.getElementById('año-pelicula-agregar').value;

    const cast_agregar =
document.getElementById('Cast-agregar').value;
```

```
const genres_agregar =
document.getElementById('genres-agregar').value;

const href_agregar =
document.getElementById('href-agregar').value;

const extract_agregar =
document.getElementById('extract-agregar').value;

const thumbnail_agregar =
document.getElementById('thumbnail-agregar').value;

const thumbnail_width_agregar =
document.getElementById('thumbnail_width-agregar').value;

const thumbnail_height_agregar =
document.getElementById('thumbnail_height-agregar').value;


const diccionario_agregar = {

  nombre: nombre_agregar,

  año: año_agregar,

  cast: cast_agregar,

  genres: genres_agregar,

  href: href_agregar,

  extract: extract_agregar,

  thumbnail: thumbnail_agregar,

  thumbnail_width: thumbnail_width_agregar,

  thumbnail_height: thumbnail_height_agregar

};
```

```
let url = 'http://192.168.84.121:8000/my-first-api?';

let firstParam = true;

// Verificar los valores que no son null y arma la consulta url a
la api

for (const variable in diccionario_agregar) {

  const valor = diccionario_agregar[variable];

  if (valor) {

    if (!firstParam) {

      url += '&';

    }

    url += `${variable}=${encodeURIComponent(valor)}`;

    firstParam = false;

  }

}

fetch(url, {

  method: 'POST',

  headers: {

    'Content-Type': 'application/json'

  },

  body: JSON.stringify(diccionario_agregar)

})

.then(response => response.json())
```



```
.then(data => {  
  
    // Obtener una referencia al elemento del contenido del cartel  
    const infoContent = document.getElementById('info-content');  
  
    // Limpiar cualquier contenido previo del cartel  
    infoContent.innerHTML = '';  
  
    // Mostrar los datos obtenidos  
    console.log(data);  
  
    infoContent.textContent = JSON.stringify(data, null, 2);  
  
    // Mostrar el cartel  
    const infoBox = document.getElementById('info-box');  
    infoBox.style.display = 'block';  
  
    })  
  
    .catch(error => console.error('Error:', error));  
  
    // Agregar un event listener al botón de cerrar para ocultar el  
cartel  
  
    document.getElementById('close-button').addEventListener('click',  
function() {  
  
        const infoBox = document.getElementById('info-box');  
  
        infoBox.style.display = 'none';  
  
    });
```

```
});  
  
//INTERFAZ MODIFICAR  
  
const modificarPeliculaForm =  
document.getElementById('modificar-pelicula-form');  
  
modificarPeliculaForm.addEventListener('submit', function(event){  
  
    event.preventDefault();  
  
    // Obtén los valores del formulario  
  
    const nombre_pelicula_modificar =  
document.getElementById('nombre-pelicula-a-modificar').value;  
  
    const año_pelicula_modificar =  
document.getElementById('año-pelicula-a-modificar').value;  
  
    const nombre_modificar =  
document.getElementById('nombre-pelicula-modificar').value;  
  
    const año_modificar =  
document.getElementById('año-pelicula-modificar').value;  
  
    const cast_modificar =  
document.getElementById('Cast-modificar').value;  
  
    const genres_modificar =  
document.getElementById('genres-modificar').value;  
  
    const href_modificar =  
document.getElementById('href-modificar').value;  
  
    const extract_modificar =  
document.getElementById('extract-modificar').value;  
  
    const thumbnail_modificar =  
document.getElementById('thumbnail-modificar').value;
```

```
const thumbnail_width_modificar =
document.getElementById('thumbnail_width-modificar').value;

const thumbnail_height_modificar =
document.getElementById('thumbnail_height-modificar').value;

const diccionario_modificar = {

  nombre: nombre_pelicula_modificar,

  año: año_pelicula_modificar,

  title: nombre_modificar,

  year: año_modificar,

  cast: cast_modificar,

  genres: genres_modificar,

  href: href_modificar,

  extract: extract_modificar,

  thumbnail: thumbnail_modificar,

  thumbnail_width: thumbnail_width_modificar,

  thumbnail_height: thumbnail_height_modificar

};

let url = 'http://192.168.84.121:8000/my-first-api?';

url +=
`http://192.168.84.121:8000/my-first-api?nombre=${nombre_pelicula_modifica
r}&año=${año_pelicula_modificar}&`

let firstParam = true;
```

```
// Verificar los valores que no son null y arma la consulta url a
la api

for (const variable in diccionario_modificar) {

    const valor = diccionario_modificar[variable];

    if (valor) {

        if (!firstParam) {

            url += '&';

        }

        url += `${variable}=${encodeURIComponent(valor)}`;

        firstParam = false;

    }

}

fetch(url, {

    method: 'PUT',

    headers: {

        'Content-Type': 'application/json'

    },

    body: JSON.stringify(diccionario_modificar)

})

.then(response => response.json())

.then(data => {

    // Obtener una referencia al elemento del contenido del cartel

    const infoContent = document.getElementById('info-content');
```

```
// Limpiar cualquier contenido previo del cartel
infoContent.innerHTML = '';

// Mostrar los datos obtenidos
console.log(data);

infoContent.textContent = JSON.stringify(data, null, 2);

// Mostrar el cartel
const infoBox = document.getElementById('info-box');
infoBox.style.display = 'block';

})

.catch(error => console.error('Error:', error));

// Agregar un event listener al botón de cerrar para ocultar el
cartel

document.getElementById('close-button').addEventListener('click',
function() {

    const infoBox = document.getElementById('info-box');

    infoBox.style.display = 'none';

});

});

//INTERFAZ ELIMINAR
```

```
const eliminarPeliculaForm =
document.getElementById('eliminar-pelicula-form');

eliminarPeliculaForm.addEventListener('submit', function(event){

    event.preventDefault();

    // Obtén los valores del formulario

    const nombre_pelicula_eliminar =
document.getElementById('nombre-pelicula-eliminar').value;

    const año_pelicula_eliminar =
document.getElementById('año-pelicula-eliminar').value;

    const diccionario_eliminar = {

        nombre: nombre_pelicula_eliminar,

        año: año_pelicula_eliminar,

    };

    // Construir la URL con los parámetros nombre y año

    let url =
`http://192.168.84.121:8000/my-first-api?nombre=${encodeURIComponent(nombre_pelicula_eliminar)}&año=${encodeURIComponent(año_pelicula_eliminar)}`;

    fetch(url, {

        method: 'DELETE',

        headers: {

            'Content-Type': 'application/json'

        },

        body: JSON.stringify(diccionario_eliminar)
```

```
    })

    .then(response => response.json())

    .then(data => {

        // Obtener una referencia al elemento del contenido del cartel

        const infoContent = document.getElementById('info-content');

        // Limpiar cualquier contenido previo del cartel

        infoContent.innerHTML = '';

        // Mostrar los datos obtenidos

        console.log(data);

        infoContent.textContent = JSON.stringify(data, null, 2);

        // Mostrar el cartel

        const infoBox = document.getElementById('info-box');

        infoBox.style.display = 'block';

    })

    .catch(error => console.error('Error:', error));

    // Agregar un event listener al botón de cerrar para ocultar el
    cartel

    document.getElementById('close-button').addEventListener('click',
function() {

        const infoBox = document.getElementById('info-box');

        infoBox.style.display = 'none';
```

```
});
```

```
});
```

```
});
```

html.

```
<!DOCTYPE html>

<html lang="es">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Filmania</title>

  <link rel="stylesheet" href="sitio_web/css/main.css">

  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.
css" rel="stylesheet"

integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2
QvZ6jIW3" crossorigin="anonymous">

  <meta name="description" content="Página de una sola página con
información sobre películas">

  <meta name="keywords" content="películas, cine">

  <meta name="author" content="Julieta Lopez Ceratto & David Slepoy">
```



```
<meta name="copyright" content="Filmania">

<script src="sitio_web/js/main.js"></script>

</head>

<body>

  <div class="container-fluid white-box">

    <header class="mb-3">

      <nav class="navbar navbar-expand-xl nav">

        <div class="container-fluid">

          <a class="navbar-brand" href="#">

            

            <p class="mb-0" href = #Filmania>Filmania</p>

          </a>

          <button class="navbar-toggler" type="button"
data-bs-toggle="collapse" data-bs-target="#navbarNav"

            aria-controls="navbarNav" aria-expanded="false"
aria-label="Toggle navigation">

            <span class="navbar-toggler-icon"></span>

          </button>

          <div class="collapse navbar-collapse" id="navbarNav">

            <ul class="navbar-nav mx-auto">

              <li class="nav-item">
```

```
                <a class="nav-link" aria-current="page"
href="#Buscar_pelicula">Buscar película</a>

            </li>

            <li class="nav-item">

                <a class="nav-link"
href="#Agregar_pelicula">Agregar película</a>

            </li>

            <li class="nav-item">

                <a class="nav-link"
href="#Modificar_pelicula">Modificar película</a>

            </li>

            <li class="nav-item">

                <a class="nav-link"
href="#Eliminar_pelicula">Eliminar película</a>

            </li>

        </ul>

    </div>

</div>

</nav>

</header>

</div>

<!--Buscar pelicula-->

<div id="buscar-pelicula-section" style="display: none;" class = form>

    <div class="card">
```

```
<div class="card-header">

    Buscar Película

</div>

<div class="card-body">

    <form id="buscar-pelicula-form">

        <div class="mb-3">

            <label for="nombre-pelicula"
class="form-label">Nombre de la película</label>

            <input type="text" class="form-control"
id="nombre-pelicula" name="nombre" >

        </div>

        <div class="mb-3">

            <label for="año-pelicula" class="form-label">Año de
lanzamiento</label>

            <input type="number" class="form-control"
id="año-pelicula" name="año" >

        </div>

        <button type="submit" class="btn btn-primary">Buscar
Película</button>

    </form>

</div>

</div>

</div>

<!--agregar pelicula-->
```

```
<div id="agregar-pelicula-section" style="display: none;" class = form>

  <div class="card">

    <div class="card-header">

      Agregar Película

    </div>

    <div class="card-body">

      <form id="agregar-pelicula-form">

        <div class="mb-3">

          <label for="nombre-pelicula"
class="form-label">Nombre de la película</label>

          <input type="text" class="form-control"
id="nombre-pelicula-agregar" name="nombre" required>

        </div>

        <div class="mb-3">

          <label for="año-pelicula" class="form-label">Año de
lanzamiento</label>

          <input type="number" class="form-control"
id="año-pelicula-agregar" name="año" required>

        </div>

        <div class="mb-3">

          <label for="cast" class="form-label">Cast</label>

          <input type="text" class="form-control"
id="Cast-agregar" name="Cast">

        </div>

        <div class="mb-3">
```

```
        <label for="genres"
class="form-label">Generos</label>

        <input type="text" class="form-control"
id="genres-agregar" name="genres">

    </div>

    <div class="mb-3">

        <label for="href" class="form-label">href</label>

        <input type="text" class="form-control"
id="href-agregar" name="href">

    </div>

    <div class="mb-3">

        <label for="extract"
class="form-label">Resumen</label>

        <input type="text" class="form-control"
id="extract-agregar" name="extract">

    </div>

    <div class="mb-3">

        <label for="thumbnail"
class="form-label">thumbnail</label>

        <input type="text" class="form-control"
id="thumbnail-agregar" name="thumbnail">

    </div>

    <div class="mb-3">

        <label for="thumbnail_width"
class="form-label">thumbnail_width</label>

        <input type="text" class="form-control"
id="thumbnail_width-agregar" name="thumbnail_width">
```

```
        </div>

        <div class="mb-3">

            <label for="thumbnail_height"
class="form-label">thumbnail_height</label>

            <input type="text" class="form-control"
id="thumbnail_height-agregar" name="thumbnail_height">

        </div>

        <button type="submit" class="btn btn-primary">Agregar
Película</button>

    </form>

</div>

</div>

</div>

<!--Modificar pelicula-->

<div id="modificar-pelicula-section" style="display: none;" class =
form>

    <div class="card">

        <div class="card-header">

            modificar Película

        </div>

        <div class="card-body">

            <form id="modificar-pelicula-form">

                <div class="mb-3">
```

```
        <label for="nombre-pelicula"
class="form-label">Nombre de la película a modificar</label>

        <input type="text" class="form-control"
id="nombre-pelicula-a-modificar" name="nombre" required>

    </div>

    <div class="mb-3">

        <label for="año-pelicula" class="form-label">Año de
lanzamiento de la película</label>

        <input type="number" class="form-control"
id="año-pelicula-a-modificar" name="año" required>

    </div>

    <div class="mb-3">

        <label for="nombre-pelicula"
class="form-label">Nombre de la película</label>

        <input type="text" class="form-control"
id="nombre-pelicula-modificar" name="nombre">

    </div>

    <div class="mb-3">

        <label for="año-pelicula" class="form-label">Año de
lanzamiento</label>

        <input type="number" class="form-control"
id="año-pelicula-modificar" name="año">

    </div>

    <div class="mb-3">

        <label for="cast" class="form-label">Cast</label>

        <input type="text" class="form-control"
id="Cast-modificar" name="Cast">
```

```
</div>

<div class="mb-3">

    <label for="genres"
class="form-label">Generos</label>

    <input type="text" class="form-control"
id="genres-modificar" name="genres">

</div>

<div class="mb-3">

    <label for="href" class="form-label">href</label>

    <input type="text" class="form-control"
id="href-modificar" name="href">

</div>

<div class="mb-3">

    <label for="extract"
class="form-label">Resumen</label>

    <input type="text" class="form-control"
id="extract-modificar" name="extract">

</div>

<div class="mb-3">

    <label for="thumbnail"
class="form-label">thumbnail</label>

    <input type="text" class="form-control"
id="thumbnail-modificar" name="thumbnail">

</div>

<div class="mb-3">
```



```
        <label for="thumbnail_width"
class="form-label">thumbnail_width</label>

        <input type="text" class="form-control"
id="thumbnail_width-modificar" name="thumbnail_width">

    </div>

    <div class="mb-3">

        <label for="thumbnail_height"
class="form-label">thumbnail_height</label>

        <input type="text" class="form-control"
id="thumbnail_height-modificar" name="thumbnail_height">

    </div>

    <button type="submit" class="btn btn-primary">modificar
Película</button>

    </form>

</div>

</div>

</div>

<!--Eliminar película-->

<div id="eliminar-pelicula-section" style="display: none;" class =
form>

    <div class="card">

        <div class="card-header">

            Eliminar Película

        </div>
```

```
<div class="card-body">

  <form id="eliminar-pelicula-form">

    <div class="mb-3">

      <label for="nombre-pelicula" class="form-label">Nombre
de la película</label>

      <input type="text" class="form-control"
id="nombre-pelicula-eliminar" name="nombre" required>

    </div>

    <div class="mb-3">

      <label for="año-pelicula" class="form-label">Año de
lanzamiento</label>

      <input type="number" class="form-control"
id="año-pelicula-eliminar" name="año" required>

    </div>

    <button type="submit" class="btn btn-primary">Eliminar
Película</button>

  </form>

</div>

</div>

</div>

<!--box info-->

<div id="info-box" style="display: none;" class="info">

  <div id="info-content-wrapper"> <!-- Contenedor adicional -->

    <div id="info-content"></div>

  </div>

</div>
```

```
</div>

<button id="close-button">Cerrar</button>

</div>

    <!-- Bootstrap JS (incluye Popper.js primero) -->

    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/2.11.6/umd/popper.mi
n.js"></script>

    <script
src="https://cdnjs.cloudflare.com/ajax/libs/bootstrap/5.1.3/js/bootstrap.m
in.js"></script>

</body>

</html>
```