

# BRIEF de API's

Nombre de la API: API movies

## Descripción

La API de Películas proporciona acceso a una base de datos de películas. La API permite a los usuarios obtener información sobre todas las películas disponibles en la base de datos, buscar películas por id, insertar una nueva película, actualizar una película existente y eliminar una película existente.

## Endpoints

Obtener todas las películas

Este endpoint devuelve todas las películas disponibles en la base de datos.

GET /movies

## Response

Content-Type: application/json

```
[
  {
    "id": 1,
    "name": "Nombre de la película",
    "genre": "Género de la película",
    "summary": "Resumen de la película",
    "director": "Director de la película",
    "cover": "URL de la portada de la película"
  },
  {
    "id": 2,
    "name": "Nombre de la película",
    "genre": "Género de la película",
    "summary": "Resumen de la película",
    "director": "Director de la película",
    "cover": "URL de la portada de la película"
  }
  //... y así sucesivamente
]
```

Obtener una película por id

Este endpoint devuelve una película específica según el id proporcionado.

## Request

GET /movies/{id}

Parámetros

id: ID de la película a buscar.

Content-Type: application/json

```
{  
  "id": 1,  
  "name": "Nombre de la película",  
  "genre": "Género de la película",  
  "summary": "Resumen de la película",  
  "director": "Director de la película",  
  "cover": "URL de la portada de la película"  
}
```

Insertar una nueva película

Este endpoint inserta una nueva película en la base de datos.

Request

POST /movies

Body

name: Nombre de la película.  
genre: Género de la película.  
summary: Resumen de la película.  
director: Director de la película.  
cover: URL de la portada de la película.

Ejemplo:

json

```
{  
  "name": "Nombre de la película",  
  "genre": "Género de la película",  
  "summary": "Resumen de la película",  
  "director": "Director de la película",  
  "cover": "URL de la portada de la película"  
}
```

Content-Type: application/json

```
{
  "id": 1,
  "name": "Nombre de la película",
  "genre": "Género de la película",
  "summary": "Resumen de la película",
  "director": "Director de la película",
  "cover": "URL de la portada de la película"
}
```

## Actualizar una película

Este endpoint actualiza una película existente en la base de datos según el id proporcionado.

Request

PUT /movies/{id}

## Parámetros

id: ID de la película a actualizar.

## Body

name: Nombre de la película.  
 genre: Género de la película.  
 summary: Resumen de la película.  
 director: Director de la película.  
 cover: URL de la portada de la película.

## Ejemplo:

json

```
{
  "name": "Nuevo nombre de la película",
  "genre": "Género de la película actualizado",
  "summary": "Resumen actualizado de la película",
  "director": "Director actualizado de la película",
  "cover": "URL actualizada de la portada de la película"
}
```

Content-Type: application/json

```
{
  "id": 1,
  "name": "Nuevo nombre de la película",
  "genre": "Género de la película actualizado",
  "summary": "Resumen actualizado de la película",
  "director": "Director actualizado de la película",
  "cover": "URL actualizada de la portada de la película"
}
```

```
"director": "Director actualizado de la película",  
"cover": "URL actualizada de la portada de la película"  
}
```

Eliminar una película

Este endpoint elimina una película existente en la base de datos según el id proporcionado.  
Request

DELETE /movies/{id}

Parámetros

id: ID de la película a eliminar.

204 No Content-Type: N/A

Esta respuesta indica que la película se eliminó correctamente de la base de datos y no se devuelve ningún contenido en el cuerpo de la respuesta. Si la película con el id proporcionado no existe, la respuesta será un código de estado 404 Not Found.

## BRIEF de Aplicación

La aplicación web es un catálogo de películas en línea, que permite al usuario ver información sobre las películas y agregar nuevas películas a la base de datos. La aplicación está construida utilizando una arquitectura cliente-servidor, donde el cliente es un sitio web y el servidor es una API RESTful.

El sitio web del cliente tiene tres páginas principales: una página de inicio, una página de lista de películas y una página de detalle de película. En la página de inicio, se muestra una lista de las películas más populares. La página de lista de películas muestra una lista de todas las películas disponibles en la base de datos. En la página de detalle de película, se muestra información detallada sobre una película seleccionada, como su nombre, género, resumen y director.

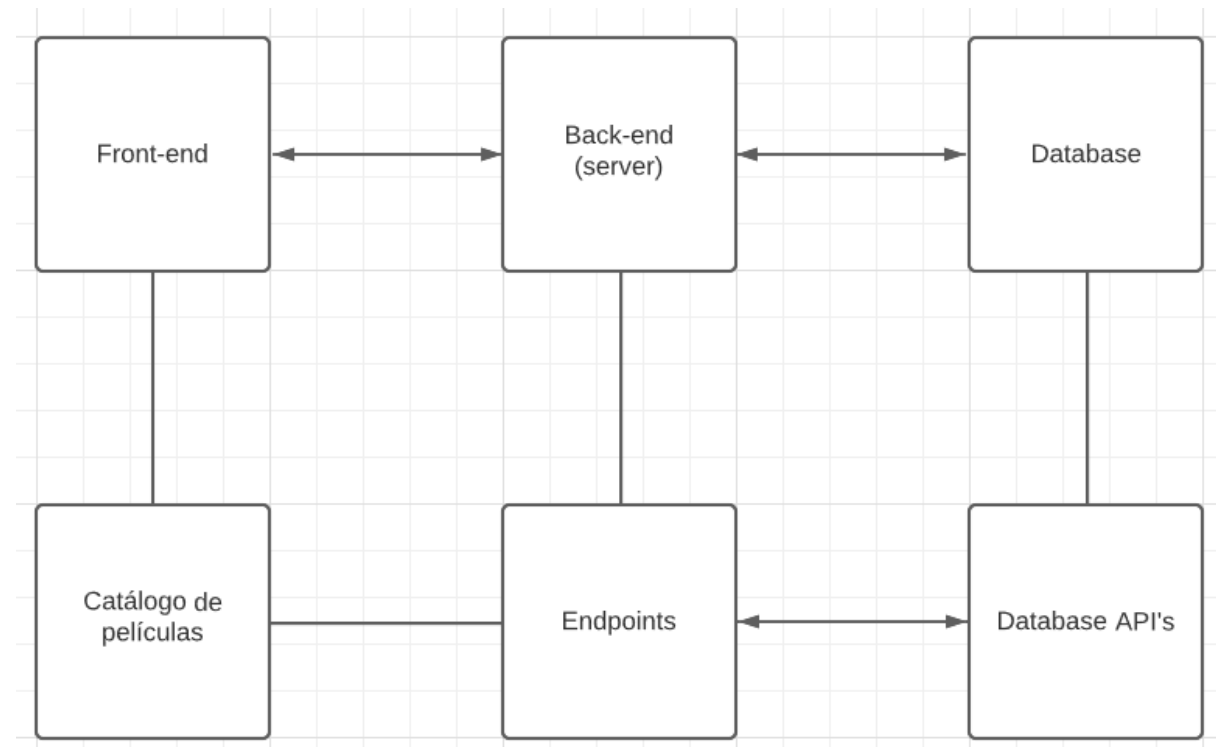
El servidor API RESTful tiene endpoints que permiten al cliente realizar operaciones en la base de datos de películas. Los endpoints disponibles son:

- Obtener todas las películas
- Obtener una película específica por ID
- Agregar una nueva película

La aplicación utiliza una base de datos MySQL para almacenar la información de las películas. La base de datos tiene una tabla de películas con las siguientes columnas: ID, nombre, género, portada, resumen y director.

En general, la aplicación permite a los usuarios buscar y ver información sobre películas, así como agregar nuevas películas a la base de datos.

## Diagrama de componentes



El diagrama se conforma de tres componentes principales:

- Front-end
- Back-end
- Base de datos

El front-end es responsable de mostrar la interfaz de usuario al usuario, mientras que el back-end se encarga de procesar las solicitudes del usuario y proporcionar las respuestas adecuadas. La base de datos almacena la información de las películas y proporciona acceso a ella a través de su API.

El front-end y el back-end están conectados a través de una conexión HTTP. El back-end se comunica con la base de datos a través de su API de base de datos por medio de rutas.

El componente de Catálogo de películas se encuentra en el front-end y es responsable de mostrar las películas disponibles para el usuario y de procesar las solicitudes del usuario para obtener información detallada de la película. Los endpoints se encuentran en el back-end y son responsables de procesar las solicitudes HTTP enviadas desde el front-end y devolver las respuestas apropiadas.

## Repositorio de GitHub

<https://github.com/HeberM69/mini-reto>