

# Alkemy – Challenge Node JS

( 03-03-2022 )

( aplicación para aceleración de Abril 2022)

## Estructura y contexto del proyecto

El escenario planteado en el trabajo comienza considerando un usuario que se registra y loguea.

Se ha realizado el FrontEnd de todo el proyecto con React JS, con validaciones, botones desplegables y mensajes al usuario (con sweetalert2).

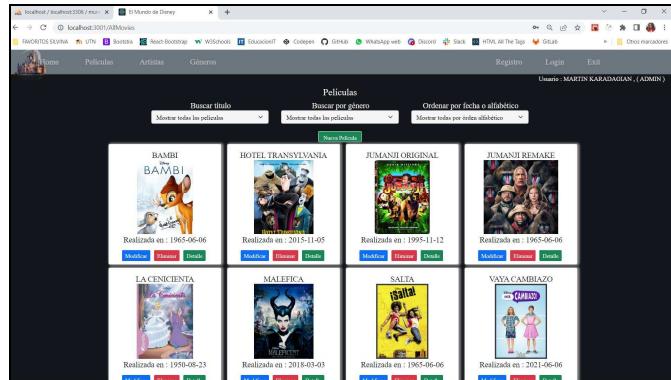
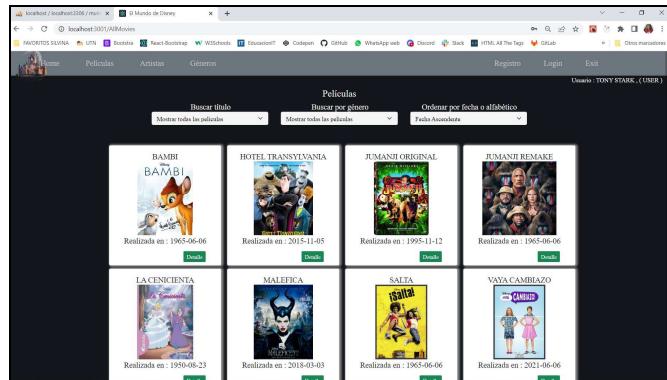
En cuanto al BackEnd , se ha trabajado con Node JS y Sequelize.

Como base de datos se ha utilizado MySQL.

Se han desarrollado tests con Mocha y Chai, y comprobaciones de funcionalidad con Postman.

Además de los servicios solicitados en el proyecto para Movies y Characters, implementación de base de datos de usuarios, registro y logueo; el BackEnd realiza los servicios relacionados a deslogueo de Users , altas, bajas y modificaciones de Genders.

Cada usuario tiene permisos relacionados a su clave personal, un token y para el acceso a las rutas de servicio además, se han implementado roles para Administrador y Usuario común, mostrando opciones diferentes para cada rol. Los usuarios comunes solo pueden consultar , mientras que el administrador además puede modificar, eliminar y dar de alta.

Admin	User
	

## Aclaraciones

Dado los enunciados del challenge, se ha tomado a los “Characters”, como actores y actrices.

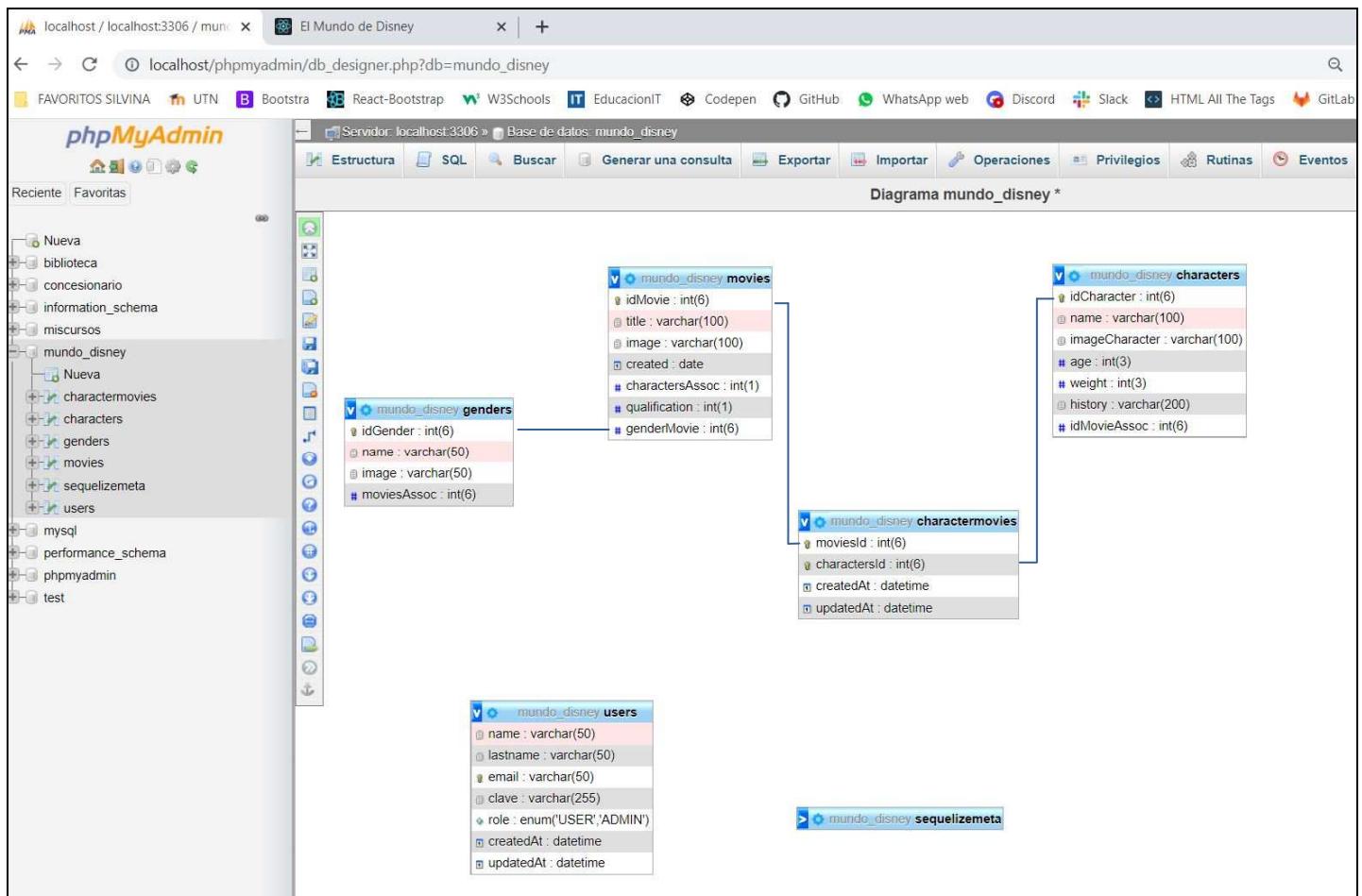
En cuanto a los campos idMoviesAssoc ( tabla Characters ) y charactersAssoc ( tabla Movies ) , no se han utilizado en la base de datos, ya que la relación entre ambas tablas es de Muchos a Muchos, por lo que se generó una tabla intermedia adicional para manejar dicha relación.

El campo moviesAssoc ( tabla Genders ) tampoco se ha utilizado en la base de datos, ya que la relación con la tabla Movies, está dada como Muchos a Uno, por lo que la información de las películas por género, se podría obtener directamente desde la tabla Movies mediante el campo genderMovie.

## Servicios para Genders

GET	Obtener todos los géneros
GET/:id	Obtener un género por su id
POST	Incorporar un género
PUT/:id	Modificar un género por su id
DELETE/:id	Eliminar un género mediante su id

## Base de Datos : Tablas y Relaciones



## Tests y comprobaciones

Se realizaron 19 tests utilizando Mocha y Chai, con resultados correctos

The screenshot shows the Visual Studio Code interface with two code files open:

- testCharacters.js**:

```
1 const chai=require('chai');
2 const chaiHttp=require('chai-http');
3 const expect=require('chai').expect;
4 require('dotenv').config();
5
6 chai.use(chaiHttp);
7
8 const url=process.env.CRED_URL;
9
10 const email=process.env.CRED_EMAIL;
11 const clave=process.env.CRED_CLAVE;
12 let user={};
13
14 describe('Testing POST /auth/login', function(){
15     it('char, respond with token', function(done){
16         data={
17             "email":email,
18             "clave":clave
```
- testMovies.js**:

```
30
31
32 describe('Testing GET /movies', function(){
33     it('respond with array all movies', function(done){
34         chai.request(url)
35             .get("/movies")
36             .set("Authorization","Bearer "+ user.token)
37             .end(function(err,res){
38                 expect(res.body).to.be.an('array');
39             })
40         });
41     });
42
43 Test Regex...
44 describe('testing GET /movies/:id', function(){
45     it('respond with single movie', function(done){
46         chai.request(url)
47             .get("/movies/1")
```

At the bottom, the terminal window displays test results:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
+ ^ x
testing PUT /movies/edit:id
✓ verify qualification
✓ verify created (54ms)

19 passing (718ms)
```

## Pruebas con Postman :

## Pantalla de inicio



## Registro de Usuarios

The screenshot shows a dark-themed web application interface. At the top, there is a navigation bar with links for 'Home', 'Películas', 'Artistas', and 'Géneros'. On the right side of the navigation bar are links for 'Registro', 'Login', and 'Exit'. The main content area features a white card titled 'Formulario de Registro'. Inside the card, there are four input fields labeled 'Nombre' (Name), 'Apellido' (Last Name), 'Email', and 'Contraseña' (Password). Below these fields is a blue 'Enviar' (Send) button.

## Email de bienvenida posterior al registro de Usuarios

The screenshot displays two windows side-by-side. On the left is the Postman application, which is used for API testing. It shows a POST request to 'localhost:3000/auth/register' with a JSON body containing user registration data:

```
1 {  
2   "id":1,  
3   "nombre":"Tony",  
4   "apellido":"Stark",  
5   "email":"sronzoni@yahoo.com",  
6   "clave":"jarvis"
```

The response from the API is:

```
1 {  
2   "msg": "Usuario ingresado exitosamente"  
3 }
```

On the right is a screenshot of a Yahoo Mail inbox. A new email from 'sronzoni@gmail.com' is visible, with the subject 'Alkemy Challenge' and the body content:

Hola TONY STARK !!!  
Bienvenid@ a Mundo Disney  
estamos muy felices de contarte entre nuestros usuarios

## Logueo y uso de Token

The top half of the image shows a screenshot of the Postman application. A POST request is being made to `localhost:3000/auth/login`. The 'Body' tab is selected, showing a JSON payload:

```

1 {
2   "msg": "Usuario MARTIN KARADAGIAN logueado !",
3   "user": {
4     "name": "MARTIN",
5     "lastName": "KARADAGIAN",
6     "email": "SLRONZUNI@YAHOO.COM",
7     "clave": "52d105NTQRuASf+u012P67drkT0wdbG1HK1/sA21/EhHzzcAZwfklVdgXX",
8     "role": "ADMIN",
9     "createdAt": null,
10    "updatedAt": null
11  },
12  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7InShbwI0I3NQV3USU4ILC3sYXNobmFtZSI6IkFpbC16I1NMUk90Wk90SUBQUhPTy50T081Lcjb0F22S16I10yYiQxMCRKSVFsdUFTZnUv"
13

```

The response status is 200 OK, time 134 ms, size 918 B. The bottom half of the image shows a screenshot of a web application for movie characters. It has a header with 'Home', 'Películas', 'Artistas', and 'Géneros'. Below the header is a 'Login' form with fields for 'Email' and 'Contraseña'. The main content area displays a grid of character cards. One card for 'ANGELINA JOLIE' is visible, along with other cards for 'CORBIN BLEU', 'DWAYNE JHONSON', 'JACOB BERTRAND', 'PERSONAJES ANIMADOS', and 'PEYTON ROI'. Each card includes a thumbnail image and three buttons: 'Detalle', 'Modificar', and 'Borrar'.

## Ruta para Characters : GET

The top half of the image shows a screenshot of the Postman application. A GET request is being made to `localhost:3000/characters`. The 'Body' tab is selected, showing the response data:

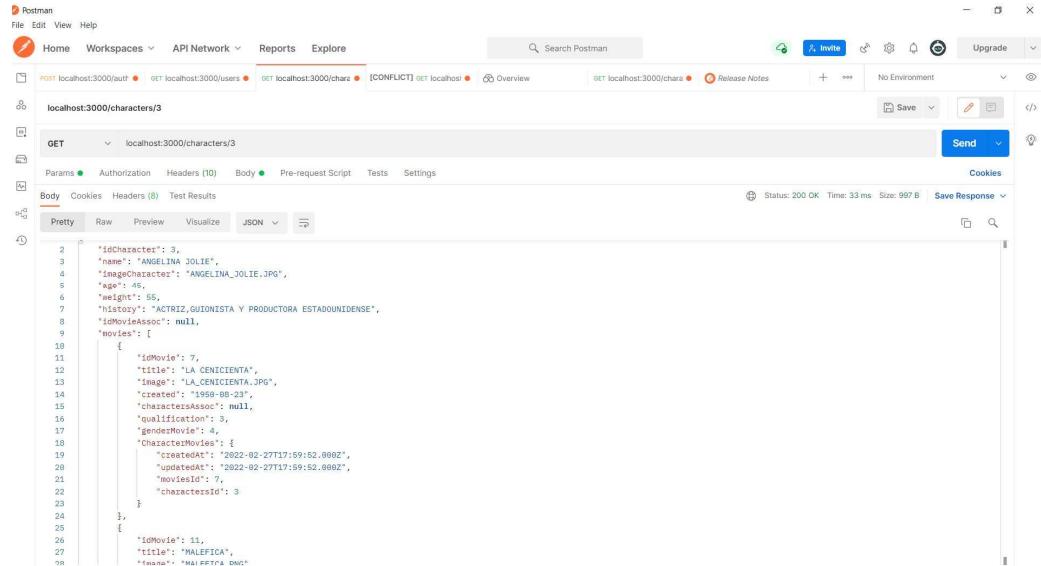
```

1 [
2   {
3     "name": "ROBIN WILLIAMS",
4     "imageCharacter": "ROBIN_WILLIAMS.JPG"
5   },
6   {
7     "name": "DWAYNE JHONSON",
8     "imageCharacter": "DWAYNE_JHONSON.JPG"
9   },
10  {
11    "name": "ANGELINA JOLIE",
12    "imageCharacter": "ANGELINA_JOLIE.JPG"
13  },
14  {
15    "name": "JACOB BERTRAND",
16    "imageCharacter": "JACOB_BERTRAND.PNG"
17  },
18  {
19    "name": "PERSONAJES ANIMADOS",
20    "imageCharacter": "DIBUJOSANIMADOS.JPG"
21  },
22  {
23    "name": "CORBIN BLEU",
24    "imageCharacter": "CORBIN_BLEU.JPG"
25  },
26  {
27    "name": "PEYTON ROI",
28

```

The response status is 200 OK, time 63 ms, size 710 B. The bottom half of the image shows a screenshot of the same web application for movie characters. It has a header with 'Home', 'Películas', 'Artistas', and 'Géneros'. Below the header is a search bar and a 'Nuevo artista' button. The main content area displays a grid of character cards. The cards for 'ANGELINA JOLIE', 'CORBIN BLEU', 'DWAYNE JHONSON', and 'JACOB BERTRAND' are visible, along with other cards for 'PERSONAJES ANIMADOS', 'PEYTON ROI', and 'ROBIN WILLIAMS'. Each card includes a thumbnail image and three buttons: 'Detalle', 'Modificar', and 'Borrar'.

## Ruta para Characters : GET\_id

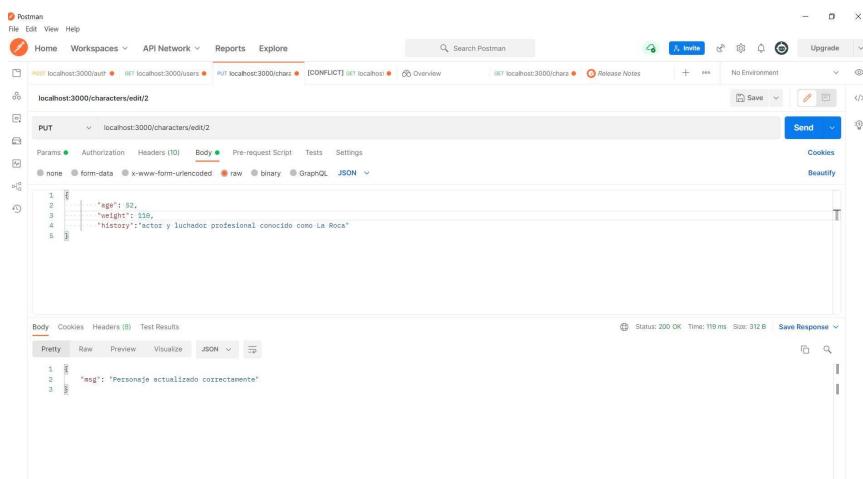


Postman screenshot showing a successful GET request to `localhost:3000/characters/3`. The response body contains the following JSON data:

```
2 "idCharacter": 3,
3 "name": "ANGELINA JOLIE",
4 "imageCharacter": "ANGELINA_JOLIE.JPG",
5 "age": 46,
6 "weight": 55,
7 "history": "ACTRIZ, GUIONISTA Y PRODUCTORA ESTADOUNIDENSE",
8 "idMovieAssociate": null,
9 "movies": [
10   {
11     "idMovie": 7,
12     "title": "LA CENICIENTA",
13     "img": "LA_CENICIENTA.JPG",
14     "created": "1995-08-23",
15     "characterAssociate": null,
16     "qualification": 3,
17     "genderMovie": 4,
18     "CharacterMovies": {
19       "createdAt": "2022-02-27T17:59:52.000Z",
20       "updatedAt": "2022-02-27T17:59:52.000Z",
21       "movieId": 7,
22       "characterId": 3
23     }
24   },
25   {
26     "idMovie": 11,
27     "title": "MALEFICA",
28     "img": "MALEFICA.JPG"
29   }
30 ]
```

Below the Postman screenshot is a screenshot of a web application showing the details of the selected artist (Angelina Jolie). The application interface includes tabs for Home, Películas, Artistas, and Géneros. The selected artist's profile shows her photo, name, age (45), weight (55), and her history as an actress, screenwriter, and producer. A table titled "Películas en las que participó" lists the movies she has been in, including "LA CENICIENTA" and "MALEFICA".

## Ruta para Characters : PUT



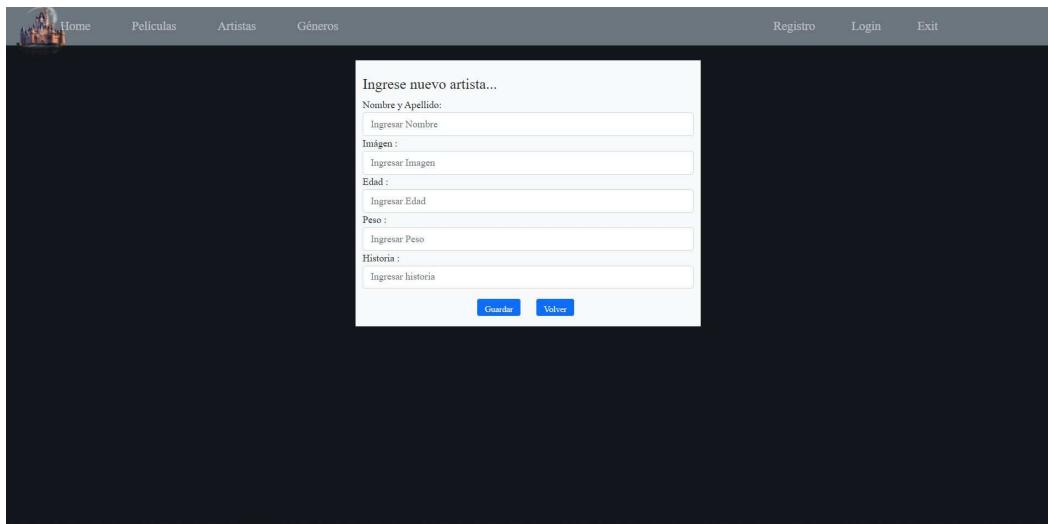
Postman screenshot showing a successful PUT request to `localhost:3000/characters/edit/2`. The request body contains the following JSON data:

```
1 {
2   "name": "DWAYNE JOHNSON",
3   "age": 52,
4   "weight": 110,
5   "history": "Actor y luchador profesional conocido como La Roca"
```

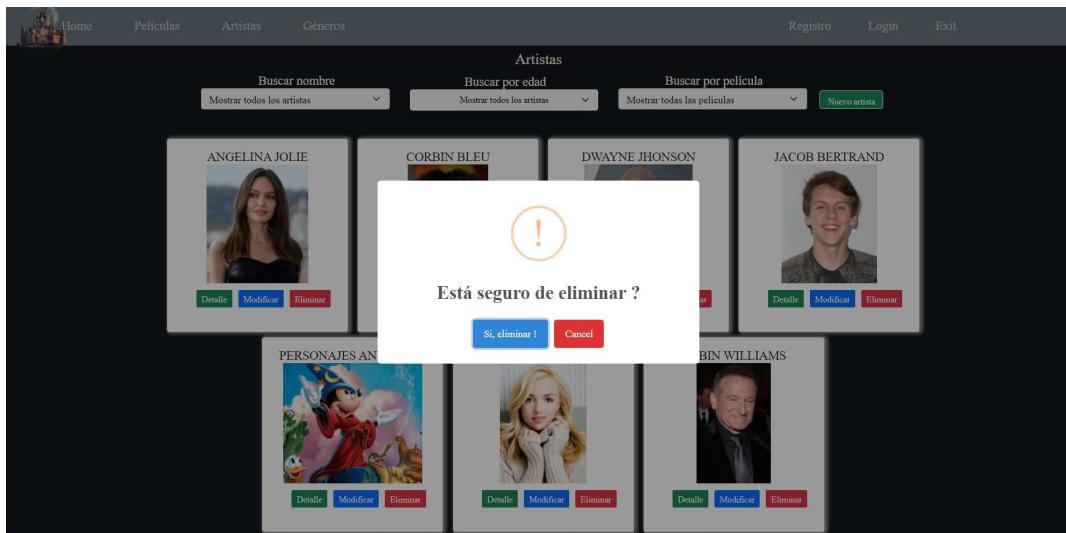
The response body indicates the update was successful: "msg: 'Personaje actualizado correctamente'".

Below the Postman screenshot is a screenshot of a web application showing the update of an artist's information. The application interface includes tabs for Home, Películas, Artistas, and Géneros. The selected artist's profile shows his photo, name (Dwayne Johnson), age (52), and weight (110). A text input field contains his history: "ACTOR Y LUCHADOR PROFESIONAL CONOCIDO COMO LA ROCA".

## Ruta para Characters :POST



## Ruta para Characters : DELETE



## Ruta para Characters : DELETE cuando existen películas asociadas

```
1 "msg": "No se puede eliminar. El artista tiene 2 película/s relacionada/s. PRIMERO DEBE QUITAR LA RELACION",
2 "data": {
3     "id": "3",
4     "name": "Jacob Bertrand",
5     "lastName": "Bertrand",
6     "image": "https://res.cloudinary.com/dqjwvzq3t/image/upload/v1610823750/characters/3.jpg",
7     "age": 22,
8     "weight": 75,
9     "history": "Jacob Bertrand es un actor norteamericano conocido por su papel en la serie de televisión 'The Walking Dead'. Tiene una hermosa sonrisa y ojos azules.",
10     "movies": [
11         {
12             "id": "2",
13             "name": "Personajes animados",
14             "year": "2022-02-27T19:14:24.000Z",
15             "director": "Pedro Gómez"
16         },
17         {
18             "id": "3",
19             "name": "Personajes animados",
20             "year": "2022-02-27T19:14:24.000Z",
21             "director": "Pedro Gómez"
22         }
23     ]
24 }
```

## Ruta de filtros para Characters :

( para todos los filtros, se han realizado botones desplegables )

### Filtro por Nombre

```
1
2
3     "idCharacter": 3,
4     "name": "ANGELINA JOLIE",
5     "imageCharacter": "ANGELINA_JOLIE.JPG",
6     "age": 40,
7     "weight": 85,
8     "history": "ACTRIZ, GUIONISTA Y PRODUCTORA ESTADOUNIDENSE",
9     "idMovieAssoc": null,
10    "movies": [
11      {
12        "idMovie": 7,
13        "title": "LA CENICIENTA",
14        "CharacterMovie": {
15          "createdAt": "2022-02-27T17:59:52.000Z",
16          "updatedAt": "2022-02-27T17:59:52.000Z",
17          "moviesId": 7,
18          "characterId": 3
19        }
20      }
21    ],
22    "idMovie": 11,
23    "title": "MALEFICA",
24    "CharacterMovie": {
25      "createdAt": "2022-02-27T19:14:24.000Z",
26      "updatedAt": "2022-02-27T19:14:24.000Z",
27      "moviesId": 11,
28      "characterId": 3
29    }
30  }
```

### Filtro por Edad

```
1
2
3     "idCharacter": 8,
4     "name": "PEYTON ROI",
5     "imageCharacter": "PEYTON_ROI.JPG",
6     "age": 22,
7     "weight": 60,
8     "history": "ACTRIZ NORTEAMERICANA",
9     "idMovieAssoc": null,
10    "movies": [
11      {
12        "idMovie": 5,
13        "title": "VIVA GAMBAZO",
14        "CharacterMovie": {
15          "createdAt": "2022-02-28T20:04:39.000Z",
16          "updatedAt": "2022-02-28T20:04:39.000Z",
17          "moviesId": 5,
18          "characterId": 8
19        }
20      }
21    ],
22    "idMovie": 22,
23    "title": "VIVA GAMBAZO",
24    "CharacterMovie": {
25      "createdAt": "2022-02-28T20:04:39.000Z",
26      "updatedAt": "2022-02-28T20:04:39.000Z",
27      "moviesId": 22,
28      "characterId": 8
29    }
30  }
```

## Filtro por Película

Postman screenshot showing a successful GET request to `localhost:3000/characters?movies=vaya_cambazo`. The response body is a JSON array containing two character objects:

```

1  [
2    {
3      "id": 1,
4      "name": "JACOB BERTRAND",
5      "image": "JACOB_BERTRAND.JPG",
6      "age": 22,
7      "weight": 75,
8      "characterRole": "ACTOR ESTADOUNIDENSE, CONOCIDO POR SU PAPEL EL COBRA KAI",
9      "info": null,
10     "movies": [
11       {
12         "idMovie": 9,
13         "title": "JUMANJI: EL REINICIO",
14         "CharacterRole": {
15           "created": "2022-02-20T20:04:39.000Z",
16           "updated": "2022-02-20T20:04:39.000Z",
17           "version": 1
18         },
19         "Character": 4
20       }
21     ]
22   },
23   {
24     "idCharacter": 5,
25     "name": "PEYTON ROI",
26     "image": "PEYTON_ROI.JPG",
27     "age": 22,
28     "weight": 68,
29   }
30 ]

```

The UI below shows the results of the search. It displays two cards for the characters JACOB BERTRAND and PEYTON ROI, each with a 'Detalles' button.

## Ruta para Movies : GET

Postman screenshot showing a successful GET request to `localhost:3000/movies`. The response body is a JSON array containing five movie objects:

```

1  [
2    {
3      "id": 1,
4      "title": "JUMANJI ORIGINAL",
5      "image": "JUMANJI_ORIGINAL.JPG",
6      "created": "1996-11-12"
7    },
8    {
9      "id": 2,
10     "title": "JUMANJI REMAKE",
11     "image": "JUMANJI_REMAKE.JPG",
12     "created": "1965-06-06"
13   },
14   {
15     "id": 3,
16     "title": "HOTEL TRANSYLVANIA",
17     "image": "HOTEL_TRANSYLVANIA.JPG",
18     "created": "2023-11-05"
19   },
20   {
21     "id": 4,
22     "title": "BAMBI",
23     "image": "BAMBI.JPG",
24     "created": "1965-06-06"
25   },
26   {
27     "id": 5,
28     "title": "LA CENICIENTA",
29     "image": "LA_CENICIENTA.JPG",
30     "created": "1990-08-23"
31   }
32 ]

```

The UI below shows the results of the search. It displays five movie cards: BAMBI, HOTEL TRANSYLVANIA, JUMANJI ORIGINAL, JUMANJI REMAKE, and LA CENICIENTA, each with a 'Detalles' button.

## Ruta para Movies : GET\_id

Postman screenshot showing a successful GET request to localhost:3000/movies/1, returning a JSON response with movie details and cast information.

```

1 {
2   "id": 1,
3   "title": "JUMANJI ORIGINAL",
4   "image": "JUMANJI_ORIGINAL.JPG",
5   "year": 1995,
6   "charactersAsActor": null,
7   "qualification": 1,
8   "genusMovie": 1,
9   "characters": [
10     {
11       "idCharacter": 1,
12       "name": "ROBIN WILLIAMS",
13       "imageCharacter": "ROBIN_WILLIAMS.JPG",
14       "weight": 68,
15       "history": "Actor y comediante estadounidense",
16       "characterRole": null,
17       "characterMovie": 1,
18       "createdAt": "2022-02-27T12:58:12.890Z",
19       "updatedAt": "2022-02-27T12:58:12.890Z",
20       "modifiedAt": "2022-02-27T12:58:12.890Z",
21       "roles": null,
22       "characterId": 1
23     }
24   ],
25   "idCharacter": 2,
26   "name": "DWAYNE JOHNSON",
27   "imageCharacter": "DWAYNE_JOHNSON.JPG",
28 }

```

The response includes the movie title, year, and its characters (Robin Williams and Dwayne Johnson) with their respective details.

Below the Postman screenshot is a screenshot of the application interface showing the movie details page for Jumanji Original.

**Movie Details:**

- Título: JUMANJI ORIGINAL
- Imagen:
- Created: 1995-11-12
- Género: 1 - AVENTURAS
- Calificación: 1
- Ver más

**Artistas que participaron:**

Nombre	Imagen	Edad	Peso	Historia
ROBIN WILLIAMS		60	80	actor y comediante estadounidense
DWAYNE JOHNSON		52	110	actor y luchador profesional conocido como la roca

## Ruta para Movies : PUT

Postman screenshot showing a successful PUT request to localhost:3000/movies/4, updating the movie Maleficent.

```

1 {
2   "casters": "BD BD-2018",
3   "quintessential": 1,
4   "generoMovie": 5,
5   "yearMovie": 2015,
6   "sheet": "",
7   "sheet": "",
8   "sheet": "",
9   "sheet": "",
10  "sheet": ""
}

```

The response indicates the update was successful with a message: "Datos actualizados correctamente".

The application interface shows the movie Maleficent for modification.

**Movie Details:**

- Título: MALEFICA
- Imagen:
- Created: 2015-03-03
- Calificación: Actual: 1
- Género: Actual: 5

**Actores:**

Actual: 3 (Obligatoria)	Actual: 1 (Opcional)	Actual: 1 (Opcional)	Actual: 1 (Opcional)
Selección... ▾	Selección... ▾	Selección... ▾	Selección... ▾

**Personajes:** Seleccione los artistas que participan

Actual: 3 (Obligatoria)	Actual: 1 (Opcional)	Actual: 1 (Opcional)	Actual: 1 (Opcional)
Selección... ▾	Selección... ▾	Selección... ▾	Selección... ▾

Actualizar Guardar

## Ruta para Movies :POST

POST localhost:3000/movies

Params • Authorization Headers (10) Body • Pre-request Scr

none form-data x-www-form-urlencoded raw binary

```

1
2   "title": "spider man",
3   "image": "spider_man.JPG",
4   "created": "2021-06-06",
5   "qualification": 1,
6   "genderMovie": 1,
7   "charactersAssoc":2,
8   "char2":1,
9   "char3":"",
10  "char4":"",
11  "char5": ""
12

```

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```

1
2   "msg": "Película ingresada correctamente",
3   "createMovie": {
4     "idMovie": 9,
5     "title": "SPIDER MAN",
6     "image": "SPIDER_MAN.JPG",
7     "created": "2021-06-06",
8     "qualification": 1,
9     "genderMovie": 1
10  }
11

```

Ingrese nueva película...

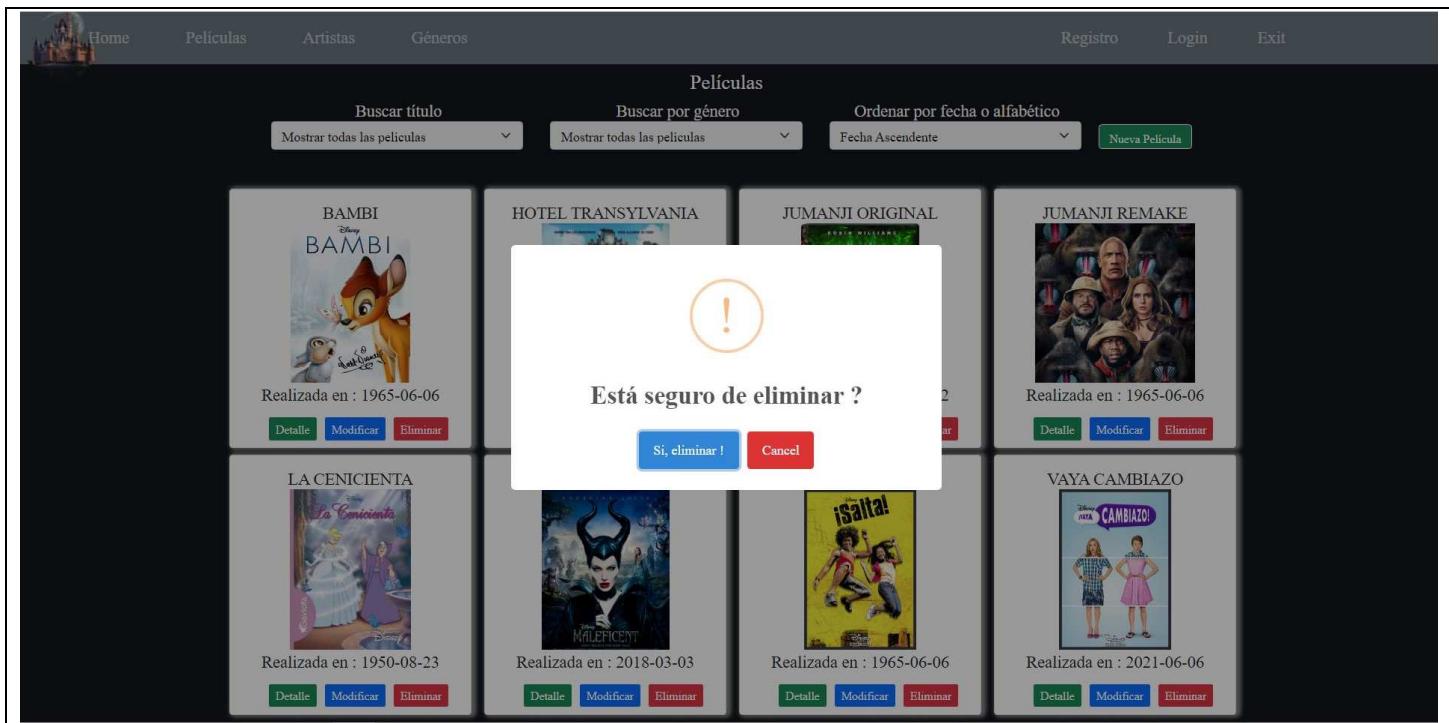
(Para ingresar una nueva película, debe existir al menos un artista para relacionarlo)

Título  
Imagen  
Creada  
Calificación  
Género  
Personajes

Selección... (obligatorio)  
Selección... (opcional)  
Selección... (opcional)  
Selección... (opcional)  
Selección... (opcional)

Guardar Volver Nuevo artista

## Ruta para Movies :DELETE



## Ruta de filtros paraMovies :

( para todos los filtros, se han realizado botones desplegables )

### Filtro por fecha ascendente

```

GET /localhost:3000/movies/order=ASC
{
  "name": "aventuras"
}
  
```

The screenshot shows the Postman interface with a successful response (Status: 200 OK). Below it, a browser window displays the application's home page with a grid of movie cards. The cards are ordered by release date, with the earliest release at the top.

Movie	Release Date
LA CENICIENTA	1950-08-23
JUMANJI REMAKE	1965-06-06
BAMBI	1965-06-06
SALTA	1965-06-06
JUMANJI ORIGINAL	1995-11-12
HOTEL TRANSYLVANIA	2015-11-05
MALEFICA	2018-03-03
VAYA CAMBIAZO	2021-06-06

### Filtro por fecha descendente

```

GET /localhost:3000/movies/order=DESC
{
  "name": "aventuras"
}
  
```

The screenshot shows the Postman interface with a successful response (Status: 200 OK). Below it, a browser window displays the application's home page with a grid of movie cards. The cards are ordered by release date, with the latest release at the top.

Movie	Release Date
VAYA CAMBIAZO	2021-06-06
MALEFICA	2018-03-03
HOTEL TRANSYLVANIA	2015-11-05
JUMANJI ORIGINAL	1995-11-12
JUMANJI REMAKE	1965-06-06
BAMBI	1965-06-06
SALTA	1965-06-06
LA CENICIENTA	1950-08-23

## Filtro por género

Postman

localhost:3000/movies?genre=aventuras

GET

localhost:3000/movies [CONFLICT] localhost:3000/characters [CONFLICT] localhost:3000/actors [CONFLICT] localhost:3000/shared [CONFLICT] Release Notes

Params Authorization Headers (10) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

1 `[`  
2  `{`  
3  `"idMovie": 2,  
4  "title": "JUMANJI ORIGINAL",  
5  "image": "JUMANJI_ORIGINAL.JPG",  
6  "realizada_en": "1995-11-12",  
7  "genero": "AVENTURA",  
8  "calificacion": 5,  
9  "personajes": 1,  
10  "actores": 1,  
11  "idGenero": 1,  
12  "name": "AVVENTURA"  
13  }  
14 ,  
15  {  
16  "idMovie": 3,  
17  "title": "JUMANJI REMAKE",  
18  "image": "JUMANJI_REMAKE.JPG",  
19  "realizada_en": "2021-06-06",  
20  "genero": "AVENTURA",  
21  "calificacion": 5,  
22  "personajes": 1,  
23  "actores": 1,  
24  "idGenero": 1,  
25  "name": "AVVENTURA"  
26  }  
27 ,  
28  {  
29  "idMovie": 8,  
30  "title": "SALTA",  
31  "image": "SALTA.JPG",  
32  "realizada_en": "1965-06-06",  
33  "genero": "AVENTURA",  
34  "calificacion": 5,  
35  "personajes": 1,  
36  "actores": 1,  
37  "idGenero": 1,  
38  "name": "AVVENTURA"  
39  }  
40 ]`

Status: 200 OK Time: 17 ms Size: 629 B Save Response

Pretty Raw Preview Visualize JSON

Body Cookies Headers (8) Test Results

Películas

Buscar título Buscar por género Ordenar por fecha o alfabético

Mostrar todas las películas Filtra Aventuras Fecha Ascendente Nueva Película

JUMANJI ORIGINAL JUMANJI REMAKE SALTA

Realizada en : 1995-11-12 Realizada en : 2021-06-06 Realizada en : 1965-06-06

Detalle Modificar Eliminar

Home Películas Artistas Géneros Registro Login Exit

## Filtro por título

Postman

localhost:3000/movies

GET

localhost:3000/movies [CONFLICT] localhost:3000/characters [CONFLICT] localhost:3000/actors [CONFLICT] localhost:3000/shared [CONFLICT] Release Notes

Params Authorization Headers (10) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

1 `[`  
2  `{`  
3  `"idMovie": 9,  
4  "title": "VAYA CAMBIAZO",  
5  "image": "VAYA_CAMBIAZO.JPG",  
6  "realizada_en": "2021-06-06",  
7  "genero": "FANTASIA",  
8  "calificacion": 5,  
9  "personajes": 1,  
10  "actores": 1,  
11  "idGenero": 3,  
12  "name": "FANTASIA"  
13  }  
14 ]`

Status: 200 OK Time: 17 ms Size: 457 B Save Response

Pretty Raw Preview Visualize JSON

Body Cookies Headers (8) Test Results

Películas

Buscar título Buscar por género Ordenar por fecha o alfabético

Mostrar todas las películas Fecha Ascendente Nueva Película

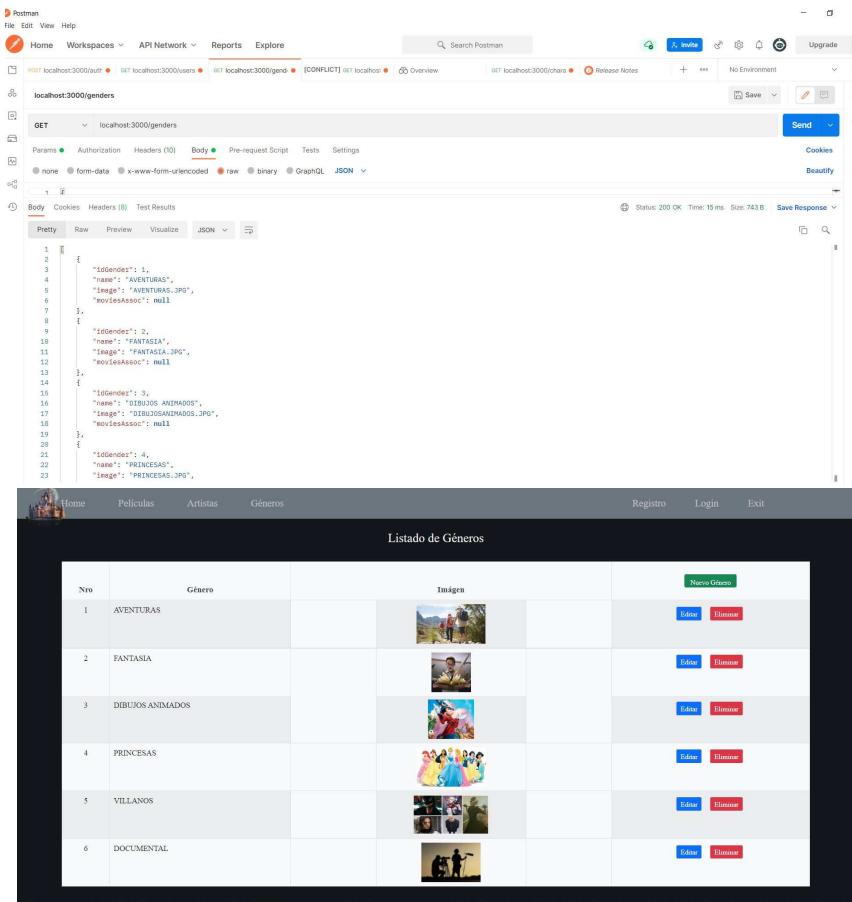
VAYA CAMBIAZO

Realizada en : 2021-06-06

Detalle Modificar Eliminar

Home Películas Artistas Géneros Registro Login Exit

## Ruta para Gender : GET



The screenshot shows the Postman interface with a successful GET request to `localhost:3000/genders`. The response body is a JSON array containing six genre objects:

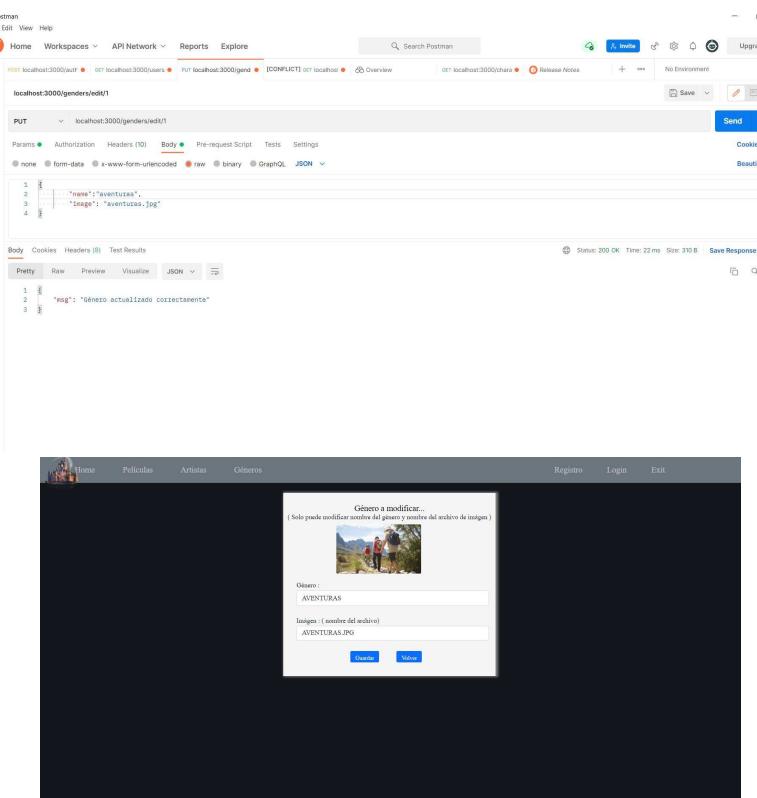
```

1 [
2   {
3     "idGenre": 1,
4     "name": "AVENTURAS",
5     "image": "AVENTURAS.JPG",
6     "moviesAsset": null
7   },
8   {
9     "idGenre": 2,
10    "name": "FANTASIA",
11    "image": "FANTASIA.JPG",
12    "moviesAsset": null
13  },
14  {
15    "idGenre": 3,
16    "name": "DIBUJOS ANIMADOS",
17    "image": "DIBUJOSANIMADOS.JPG",
18    "moviesAsset": null
19  },
20  {
21    "idGenre": 4,
22    "name": "PRINCESAS",
23    "image": "PRINCESAS.JPG",
24  }
25 ]

```

Below the API call, a screenshot of the application's genres list page is shown. The page title is "Listado de Géneros". It displays a table with columns: Nro, Género, Imagen, and actions (Nuevo Género, Editar, Eliminar). The data matches the JSON response from Postman.

## Ruta para Gender : PUT



The screenshot shows the Postman interface with a successful PUT request to `localhost:3000/genders/edit/1`. The request body contains the updated genre information:

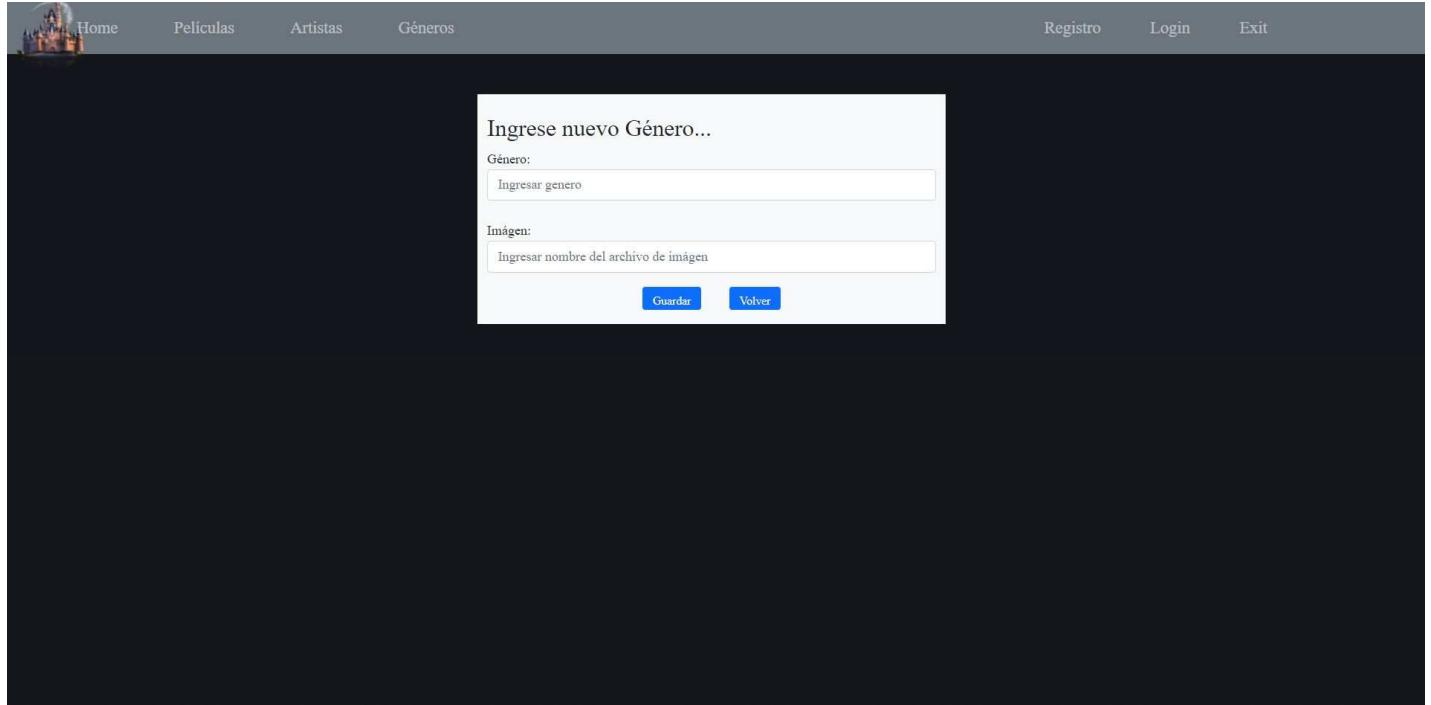
```

1 {
2   "name": "aventuras",
3   "image": "aventuras.jpg"
4 }

```

Below the API call, a screenshot of the application's genre edit page is shown. The page title is "Género a modificar...". It displays a form with fields: Género (set to AVENTURAS) and Imagen (set to AVENTURAS.JPG). The status message "Género actualizado correctamente" is visible.

## Ruta para Gender:POST



## Ruta para Gender : DELETE cuando existen películas asociadas

The screenshot shows the Postman application interface. A DELETE request is being made to 'localhost:3000/genders/3'. The response status is 400 Bad Request, with the message: 'Error : No se puede eliminar. El género tiene 2 película/s asociada/s'. The response body is a JSON object containing the genre's ID and the IDs of the two associated movies.

```
1 {
2   "msg": "No se puede eliminar. El género tiene 2 película/s asociada/s",
3   "deleteGenreWithMovie": [
4     {
5       "idMovie": 3,
6       "title": "HOTEL TRANSYLVANIA",
7       "image": "HOTELTRANSYLVANIA.JPG",
8       "created": "2025-11-09T00:00:00Z",
9       "characterassoc": null,
10      "qualification": 2,
11      "genderMovie": 3
12    },
13    {
14      "idMovie": 6,
15      "title": "BAMBI",
16      "image": "BAMBI.JPG",
17      "created": "2025-08-06T00:00:00Z",
18      "characterassoc": null,
19      "qualification": 2,
20      "genderMovie": 3
21    }
22  ]
23 }
```

The screenshot also shows a screenshot of the web application's 'Listado de Géneros' page. It displays a table with columns 'Nro' and 'Género'. The third row, which corresponds to the genre being deleted, has a red 'X' icon over it. A modal window is overlaid on the table with the message 'Error : No se puede eliminar' and the sub-message 'No se puede eliminar. El género tiene 2 película/s asociada/s'. There is an 'OK' button at the bottom of the modal.

## Ruta para Gender : DELETE

Home Películas Artistas Géneros Registro Login Exit

Listado de Géneros

Nro	Género	Imagen	Nuevo Género
1	AVENTURAS		<a href="#">Editar</a> <a href="#">Eliminar</a>
2	FANTASIA		<a href="#">Editar</a> <a href="#">Eliminar</a>
3	DIBUJOS ANIMADOS		<a href="#">Editar</a> <a href="#">Eliminar</a>
4	PRINCESAS		<a href="#">Editar</a> <a href="#">Eliminar</a>
5	VILLANOS		<a href="#">Editar</a> <a href="#">Eliminar</a>
6	DOCUMENTAL		<a href="#">Editar</a> <a href="#">Eliminar</a>

**Está seguro de eliminar ?**

[Si, eliminar !](#) [Cancel](#)

Silvina L. Ronzoni