**Registro de pruebas**

*Sergio Groppa y Marcos Vázquez*

05/05/2023

Índice

[Introducción 3](#_Toc135396599)

[Desarrollo 3](#_Toc135396600)

[Prueba 1: 3](#_Toc135396601)

[Prueba 2: 5](#_Toc135396602)

[Prueba 3: 7](#_Toc135396603)

[Bibliografía 16](#_Toc135396604)

## Introducción

En este documento redactamos todas las pruebas realizadas en entorno de desarrollo.

## Desarrollo

#### Prueba 1:

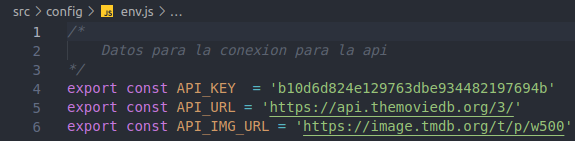
(02/04/2023)

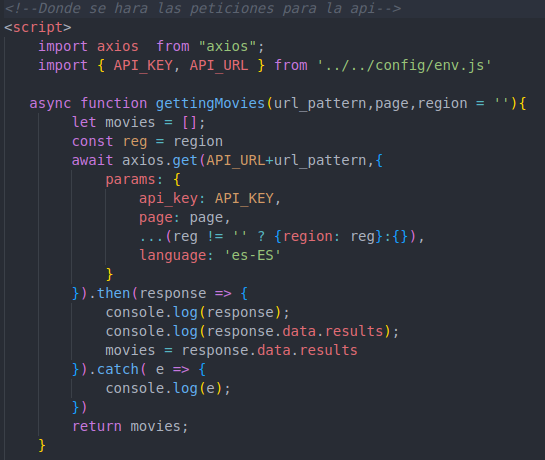
Escenario inicial:

Creado el proyecto Vue, realizar una conexión a la API de terceros (TheMovieDB).

Proceso:

Crear una cuenta en “TheMovieDB.org”, guardar la API key y la ruta inicial de la página web en el archivo config/env.js. Instalar el paquete axios para las peticiones a APIs. Crear un archivo para las peticiones de la API.



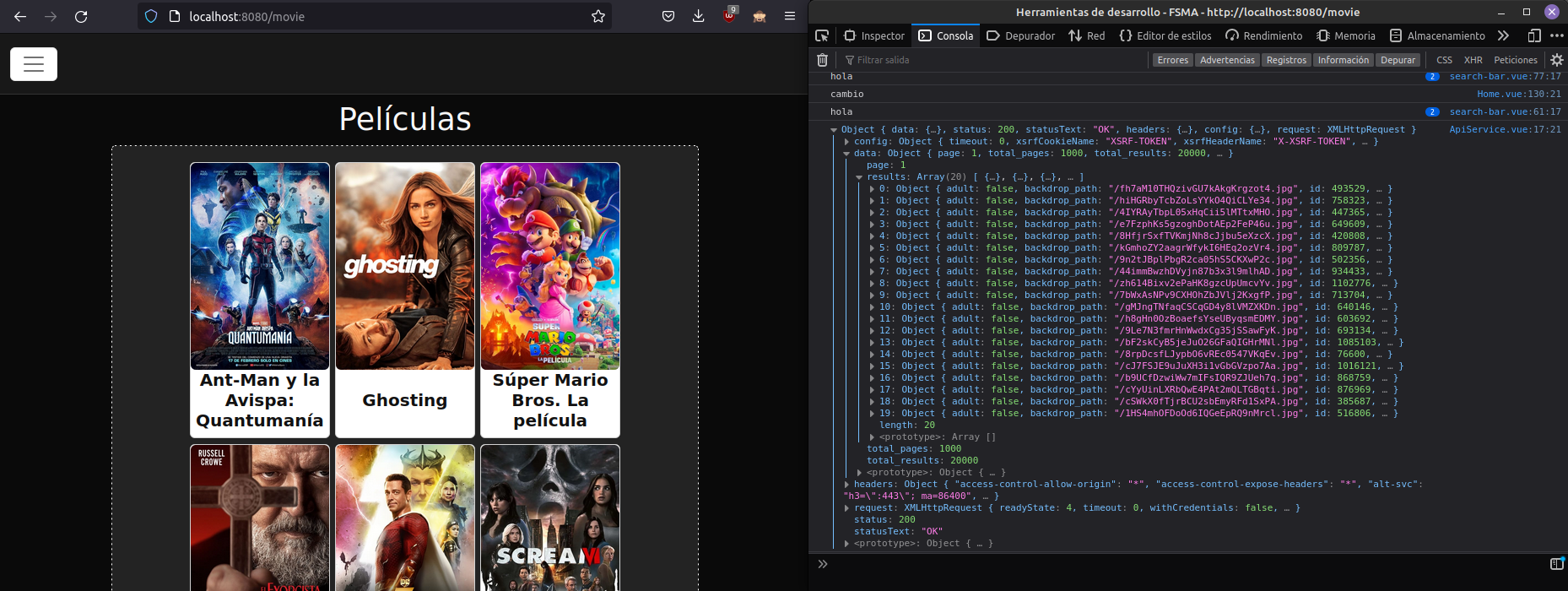


Resultado esperado:

La conexión se realice correctamente.

Resultado final:

La conexión se realiza correctamente.



#### Prueba 2:

(20/04/2023)

Escenario inicial:

Tenemos varios componentes, pero aún no podemos navegar entre ellos (home.vue y movie.vue).

Proceso:

Utilizaremos el paquete vue-router para hacer la navegación y el enrutamiento de la página. Crear un archivo router.js donde iremos definiendo que componentes se visualizan y en que rutas se muestran.



Después de configurar el archivo router.js lo importamos en main.js y en la aplicación indicarle que lo use.



Luego en la aplicación inicial (app.vue) hacemos una configuración de lo que se visualizará, el apartado de arriba un navegador (header.vue), en el medio dependiendo de la ruta el componente acorde a ello, y el pie de página (footer.vue).



Lo último, definir los enlaces para redirigir correctamente.

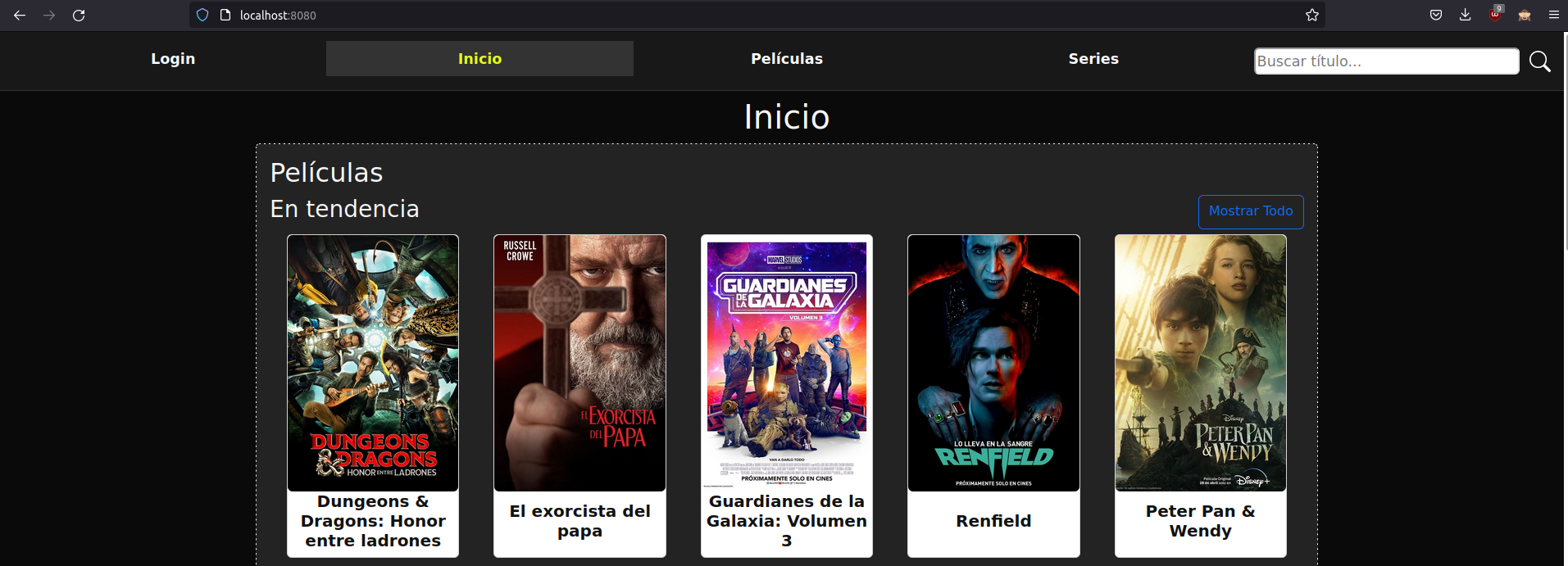


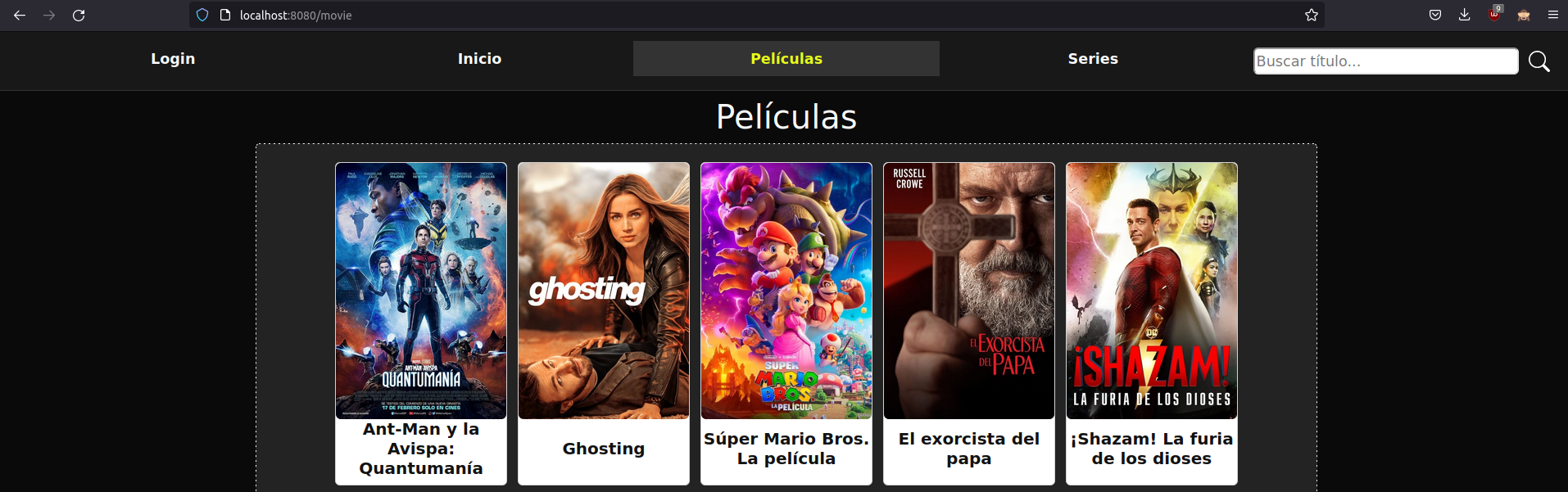
Resultado esperado:

Al clicar uno de los enlaces del navegador se redirija al componente acorde.

Resultado final:

Los enlaces del navegador se redirige correctamente.





#### Prueba 3:

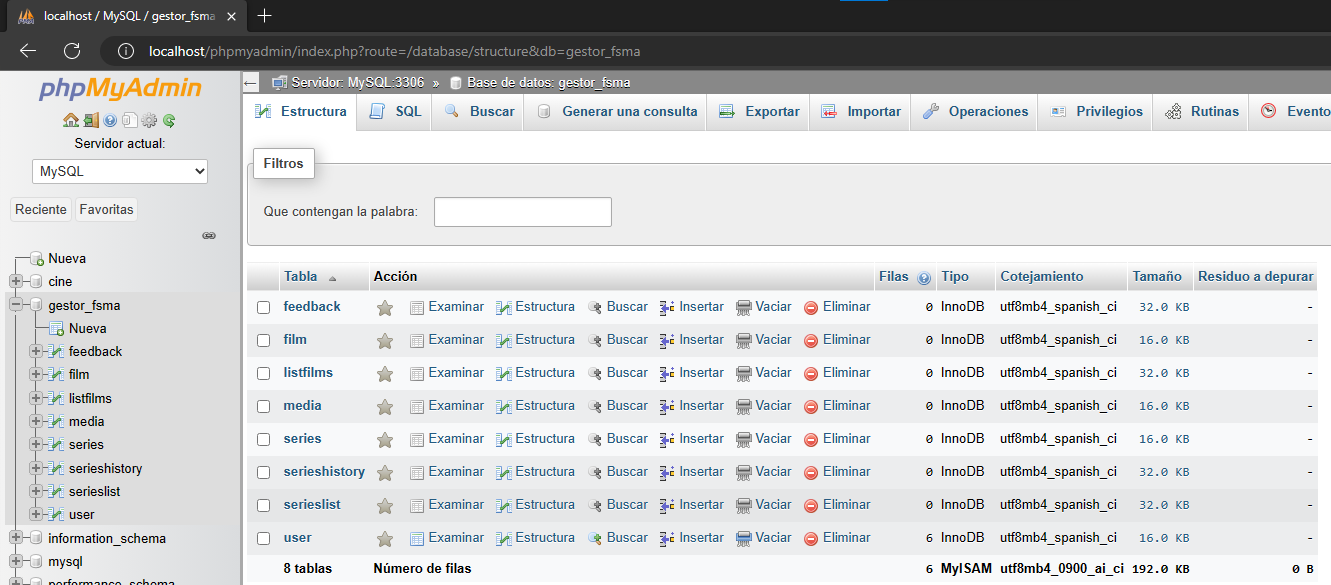
(23/04/2023)

Escenario inicial:

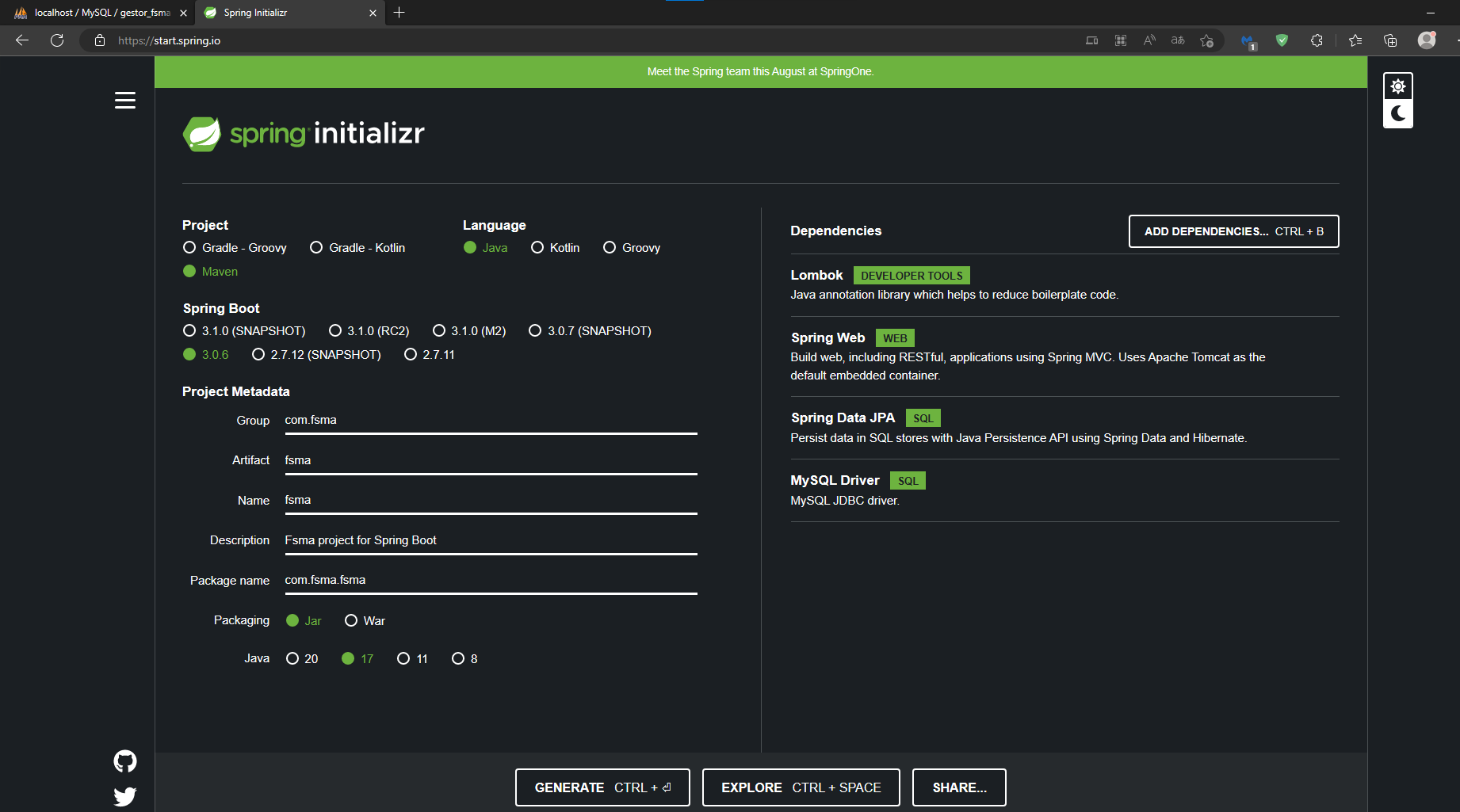
Tener el apartado Login y Register pero falta las peticiones de base de datos:

Proceso:

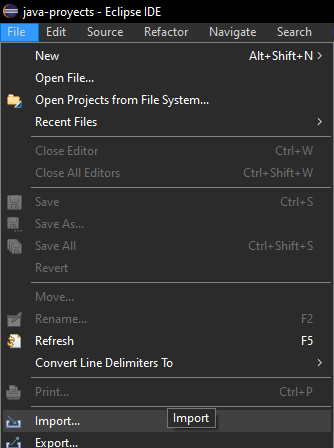
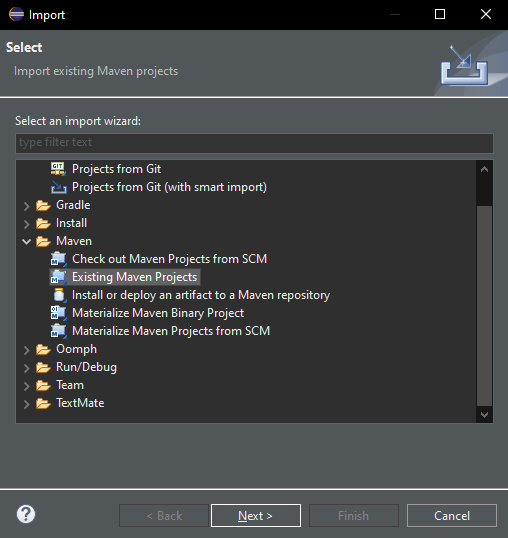
Como necesitamos peticiones a la base de datos. Primero tendríamos que tener la base de datos creada. En este caso la base de datos está dentro de WAMP definida en phpmyadmin.

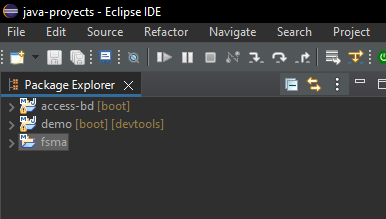
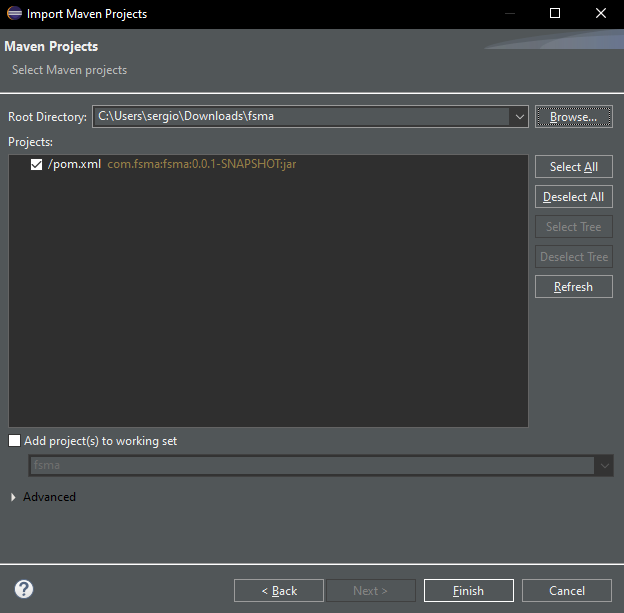


Como la utilización de base de datos se encarga el backend, en nuestro caso utilizaremos Java con el Framework Spring Boot. Tendremos que tener instalado Java en nuestro caso (Eclipse java Developers 2023-03). Crearemos una plantilla con la ayuda [Spring Initializr](https://start.spring.io/) que nos creamos un proyecto de Maven con las dependencias base.



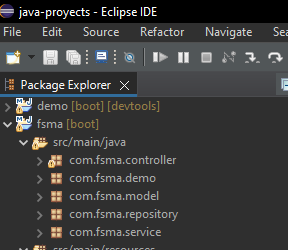
Descomprimimos el archivo y desde el propio Eclipse importamos el proyecto generado.



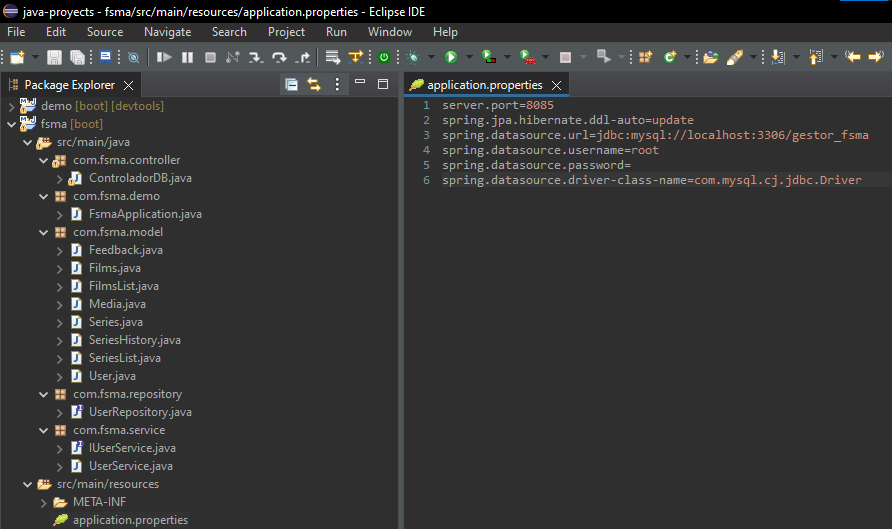
Seleccionamos el proyecto donde lo descomprimimos y al finalizar se visualiza en los proyectos del workspace.

Ahora tocaría la configuración del propio backend en nuestro caso se utilizará para petición de base de datos.

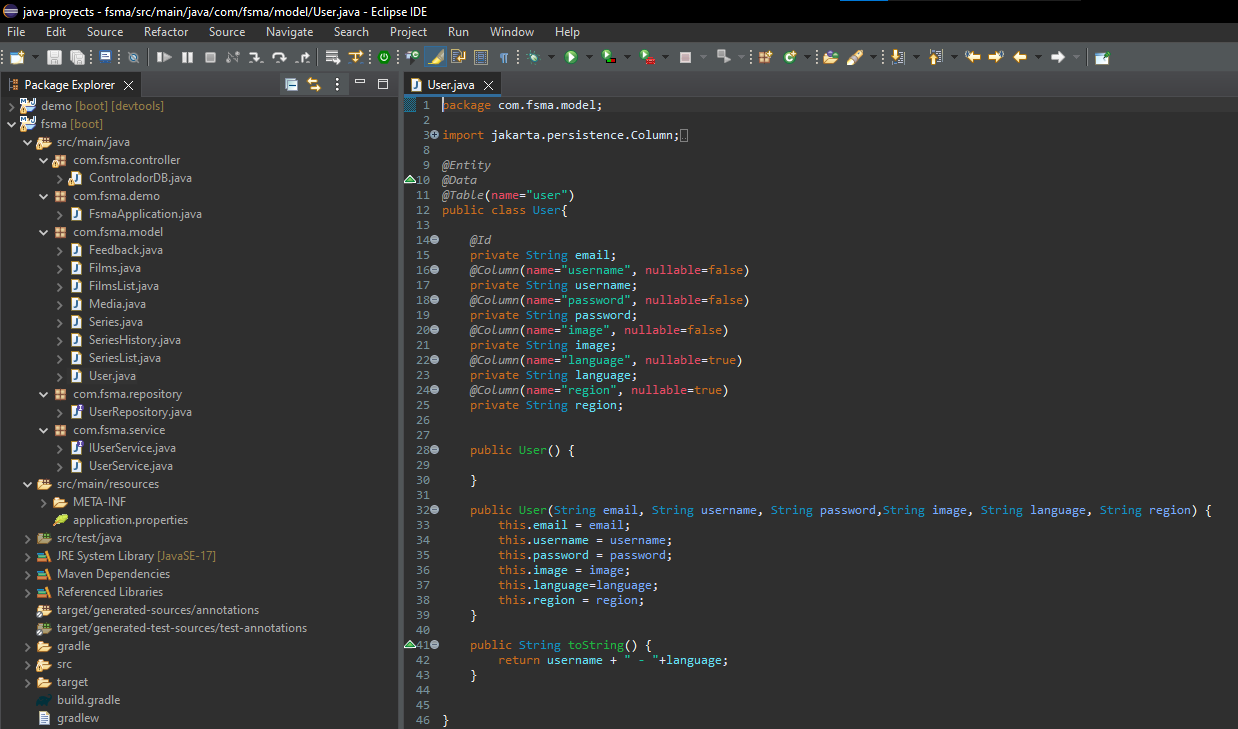
Primero creamos una estructura de carpeta así:



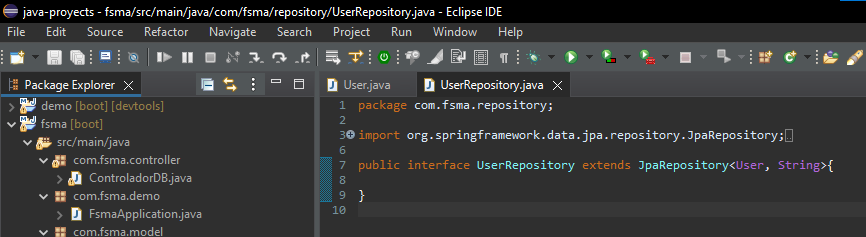
Después definimos las propiedades de la aplicación:



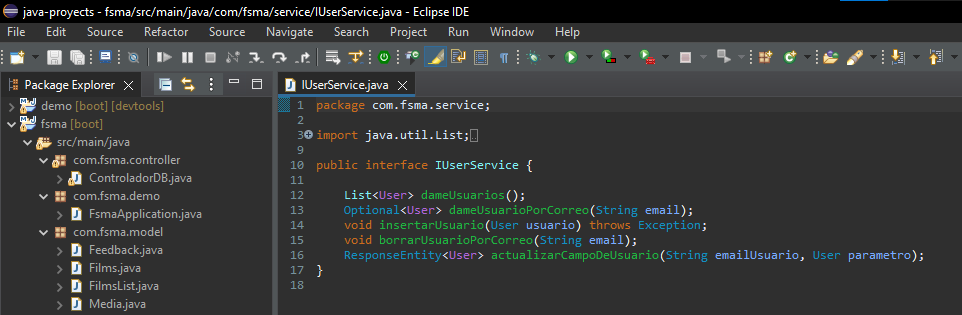
En paquete model definimos cada tabla de la base de datos:

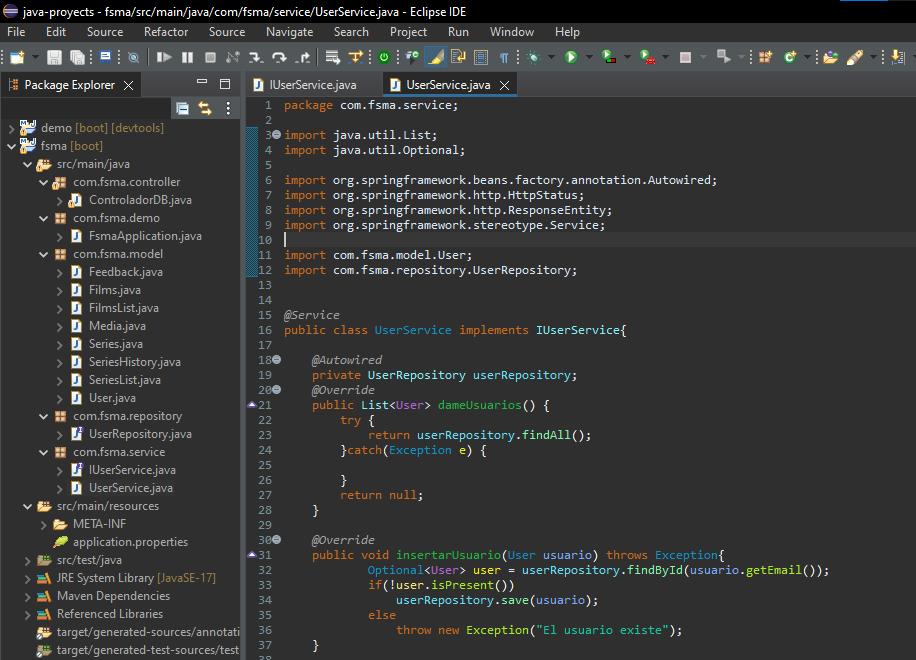


En paquete repositorio definimos una interfaz que se utiliza para sacar información de la tabla defina en el apartado de model.

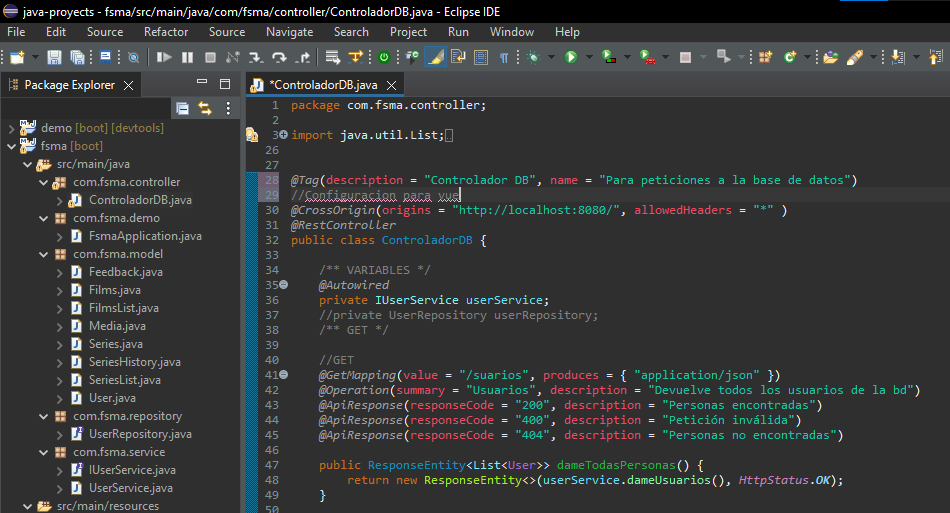


En el paquete de service creamos una interfaz y clase para las funcionalidades que se harán en esa tabla:

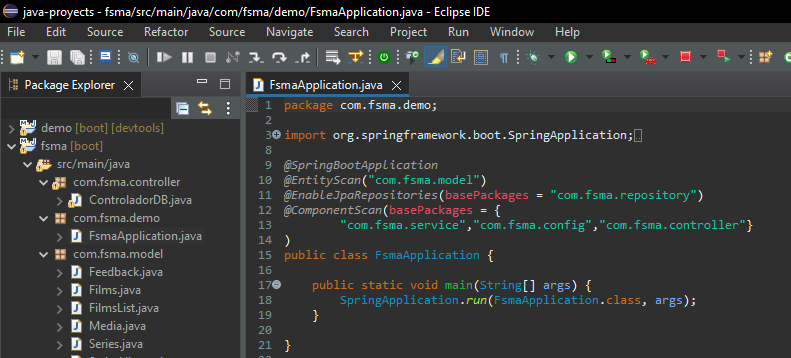




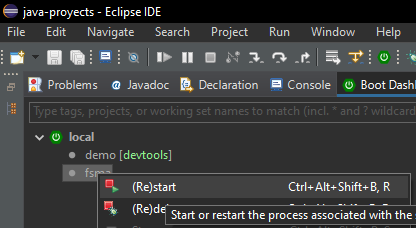
En el paquete controller se define el api rest, el mapeado para conseguir la información en que método se hace (POST, GET, PUT, DELETE).



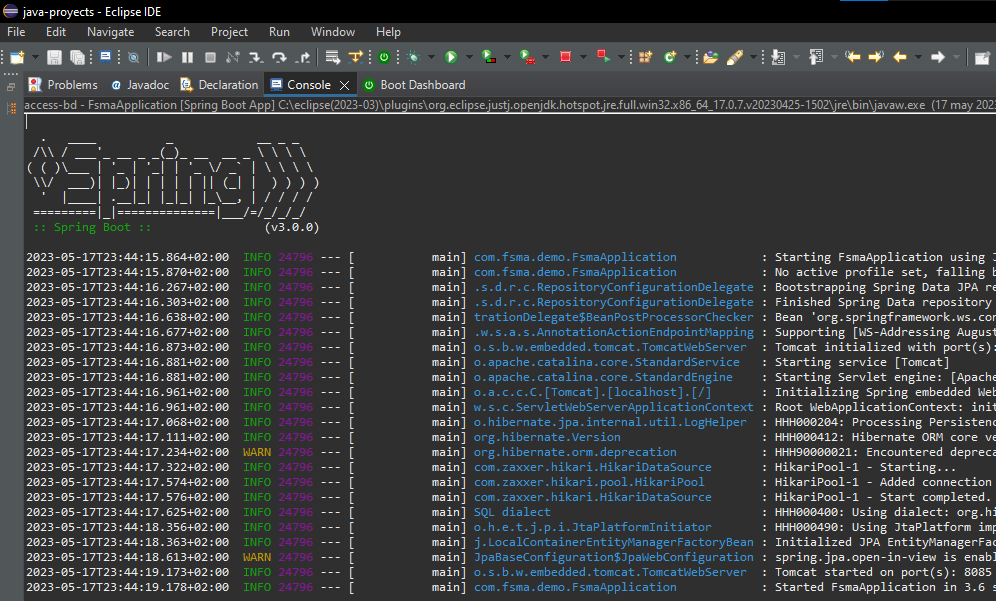
En el paquete demo se define el arranque del proyecto:



Con esto están definidas las peticiones que se pueden hacer, falta ejecutar el proyecto Spring boot se podrán hacer peticiones:

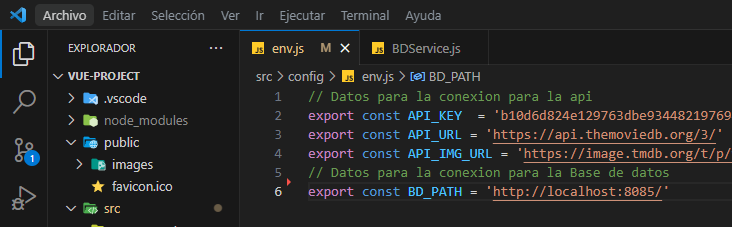


Y en la consola se visualiza que se ha iniciado:

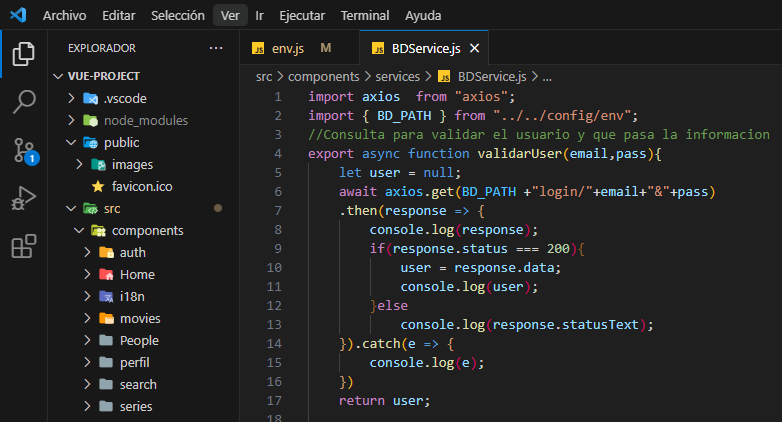


Ahora tocaría definir el proceso para hacer petición en vue.

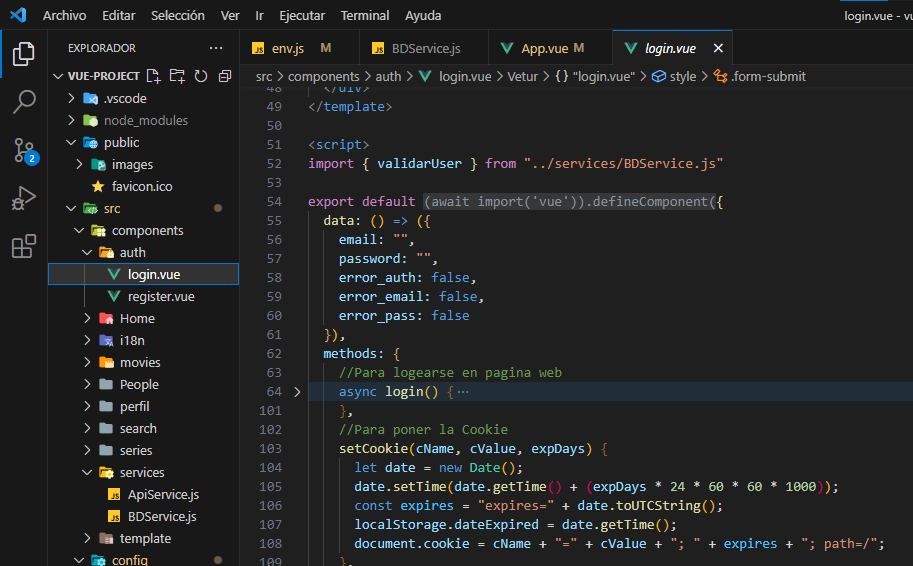
En el proyecto vue definimos en el fichero env.js la ruta para hacer la petición



Y en el apartado de servicios hacemos la petición:



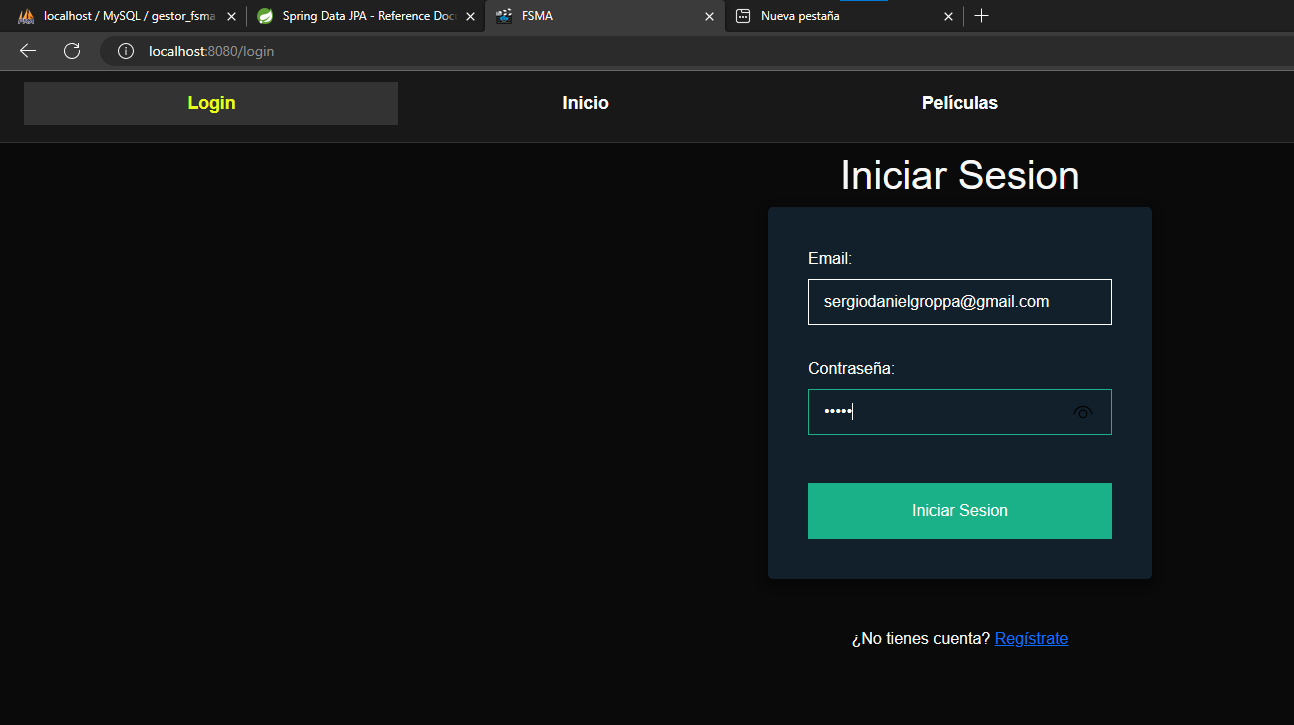
En el componente del login lo configuramos:



Resultado esperado:

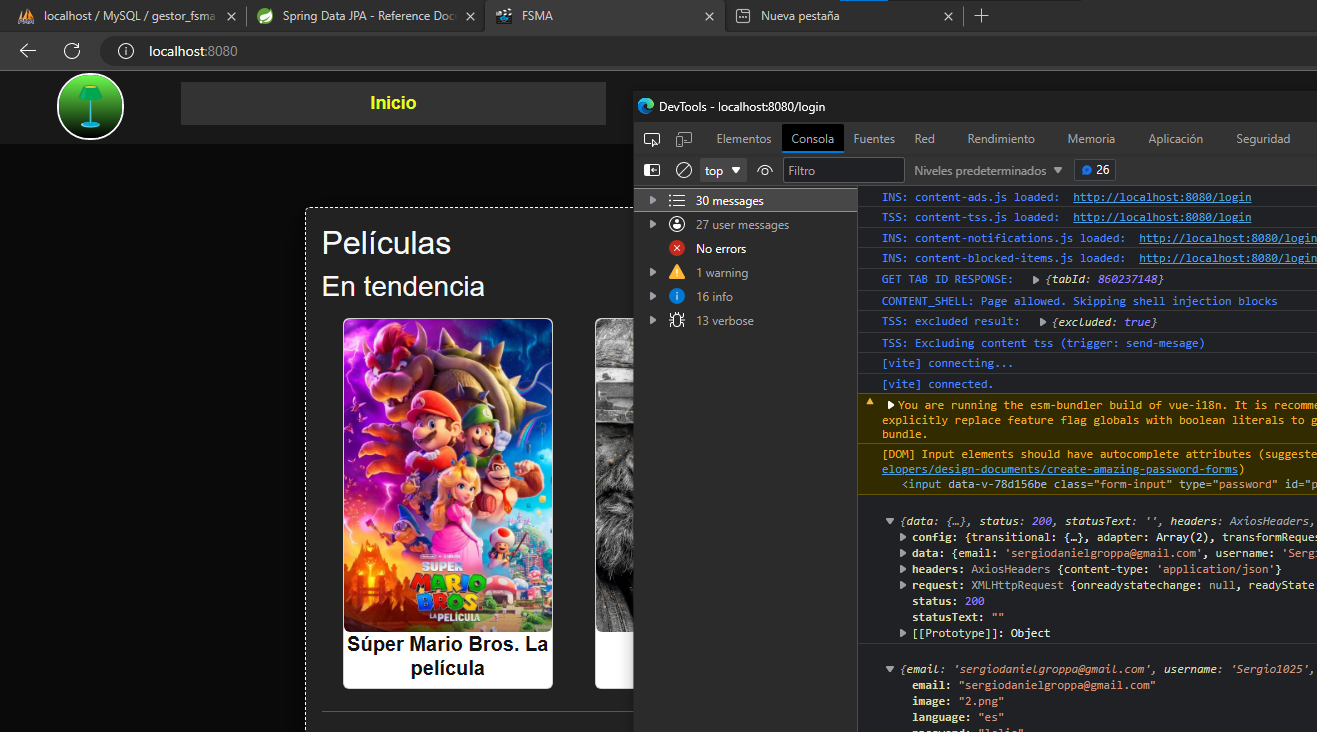
Que al logearte se inicie sesión en usuario

En el login:



Resultado final:

Consigue logearse



#### Prueba 4:

La prueba de 5 estadísticas globales, no se ha realizado porque no ha habido tiempo para crear el apartado.

#### Prueba 5:

La prueba de 4 gráficos, no se ha realizado porque no ha habido tiempo para crear el apartado.

#### Prueba 6:

La prueba de Opiniones personales, no se ha realizado porque no ha habido tiempo para crear el apartado.

#### Prueba 7:

La prueba de Notificaciones de interés, no se ha realizado porque no ha habido tiempo para crear el apartado.

#### Prueba 8:

La prueba de Recomendaciones, no se ha realizado porque no ha habido tiempo para crear el apartado.

## Bibliografía

Documentación de Vue:

https://vuejs.org/

Documentación oficial de los enrutados para Vue:

https://router.vuejs.org/

Documentación oficial de para peticiones:

https://axios-http.com/es/docs/intro

Documentación oficial para Spring Boot:

<https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>

Documentación utilización de Spring Boot con base de datos:

[Spring Data JPA - Reference Documentation](https://docs.spring.io/spring-data/jpa/docs/current/reference/html/)