**Proyecto Final**

**Desarrollo Backend NodeJS**

**ECOMMERCE**

**Nancy Barea**

**09/08/2022**

**Contenido**

[Notas antes de empezar 3](#_Toc110940030)

[Github 3](#_Toc110940031)

[Heroku 3](#_Toc110940032)

[**Menú** 4](#_Toc110940033)

[**Chat** 5](#_Toc110940034)

[**Subir Archivo (registro)** 5](#_Toc110940035)

[**Información Servidor** 8](#_Toc110940036)

[Swagger 8](#_Toc110940037)

[Graphl 9](#_Toc110940038)

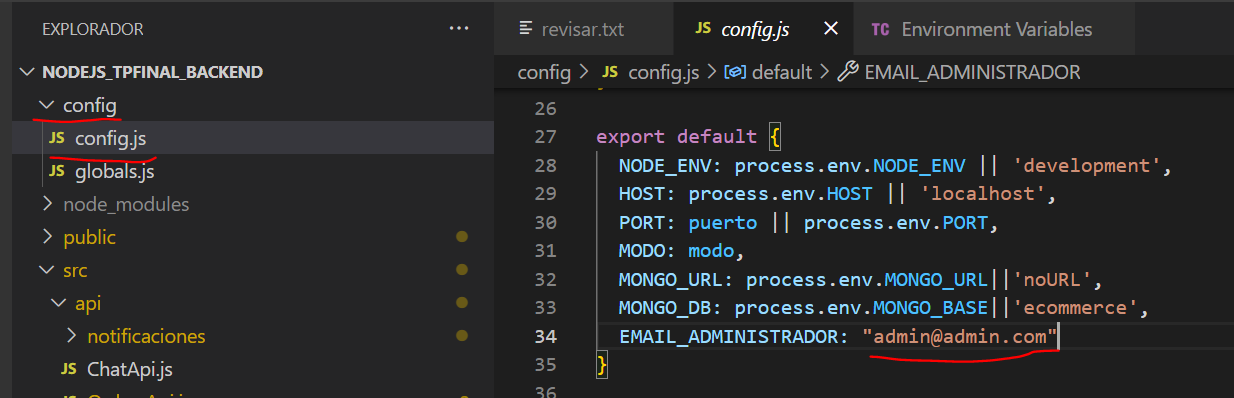
[Testing 9](#_Toc110940039)

# Notas antes de empezar

**Ejecutar el programa**

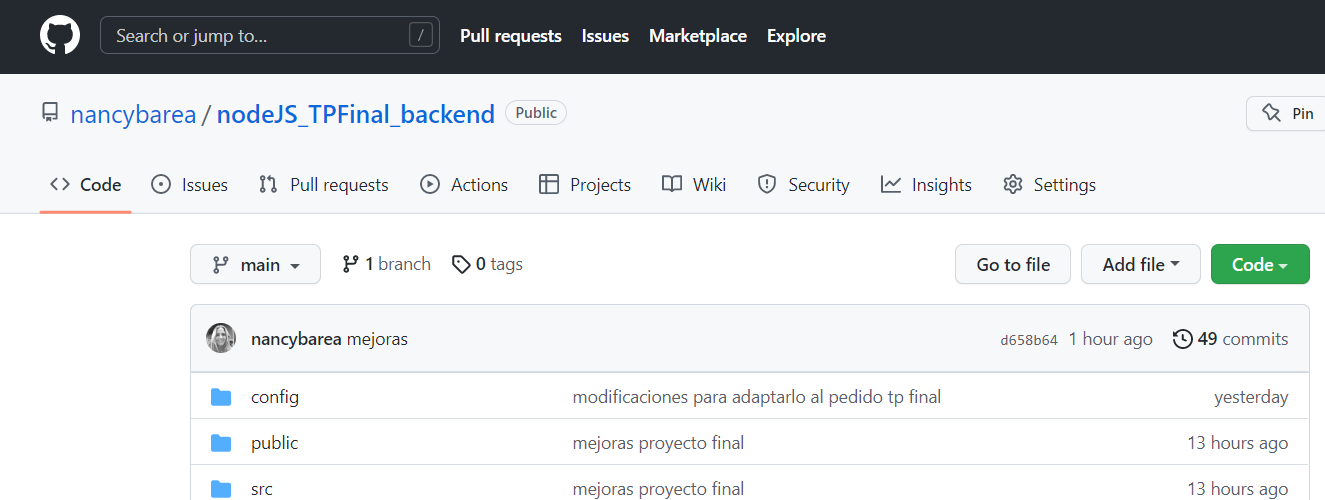


**Configuración usuario administrador**



# Github

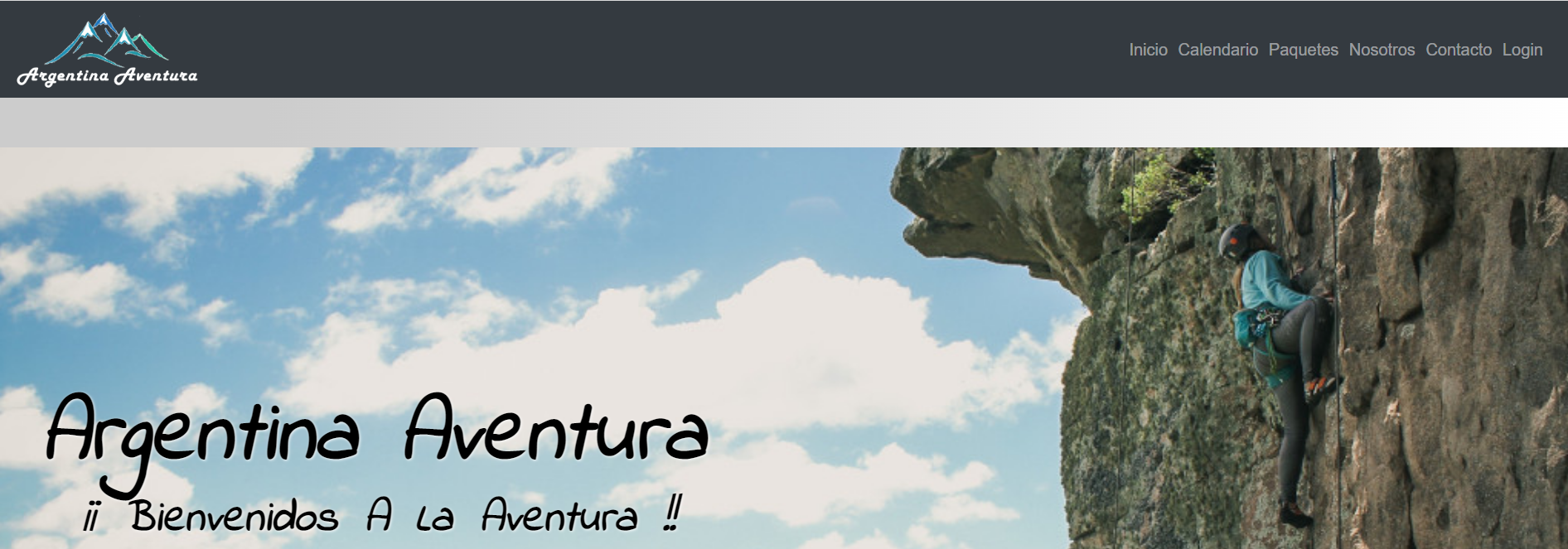
<https://github.com/nancybarea/nodeJS_TPFinal_backend>



# Heroku

<https://ecommerce-bac.herokuapp.com/web/>

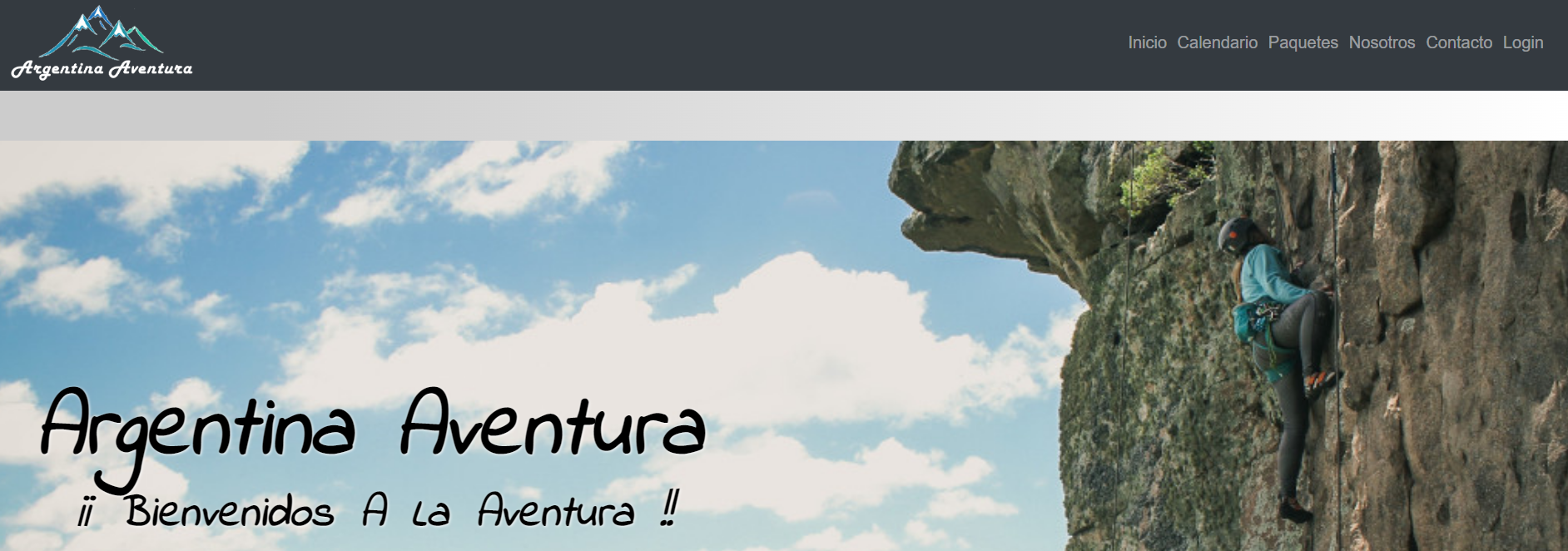
(está conectado con el repositorio de github)



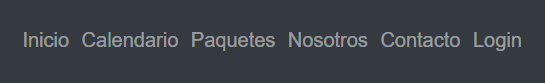
A continuación se detalla solo las funcionalidades más relevantes para este proyecto

* Menú: como se modifica el menú según el rol del usuario logueado.
* Chat: se solicito realizar un chat con socket donde pueden hablar todos los usuarios.
* Subir Archivo : subir imagen con multer y lo guarde en carpeta public
* Información servidor: solicitada en el proyecto con el uso de plantillas.

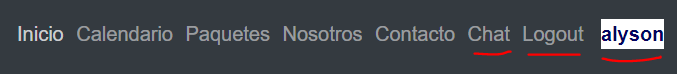
## **Menú**



* **Menu Sin usuario logueado**

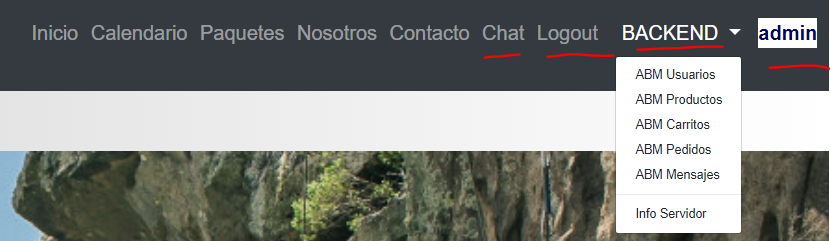


* **Menu con usuario logueado y rol “usuario”**



Siendo “alyson” el nombre del usuario logueado

* **Menu con usuario logueado y rol “admin”**

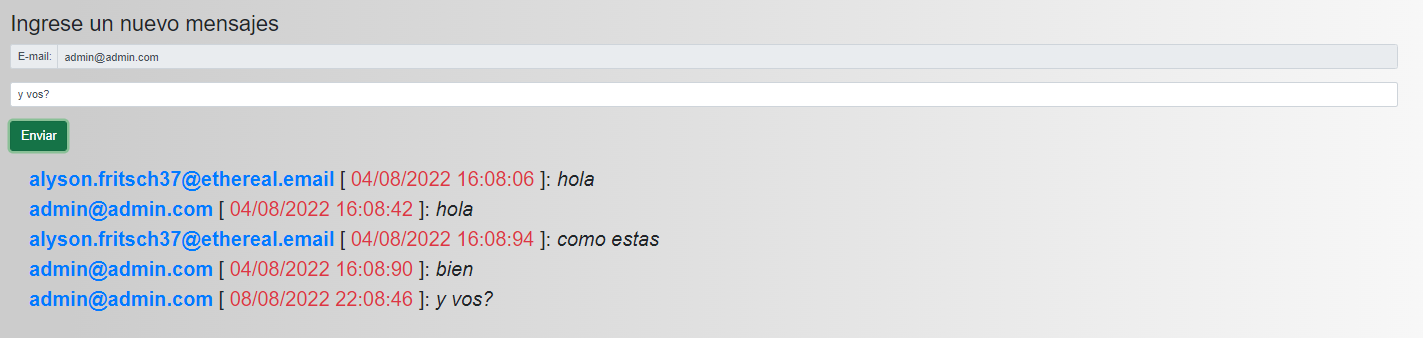


Siendo “admin” el nombre del usuario logueado

## **Chat**

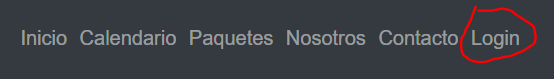
En el cuadro de email aparecerá sin poder editar, el email del usuario logueado.

El cuadro de abajo es para escribir los mensajes

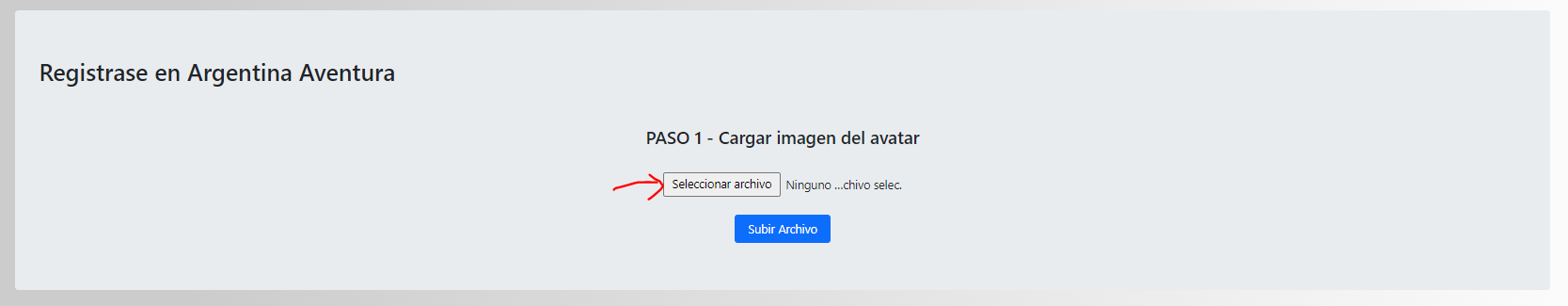


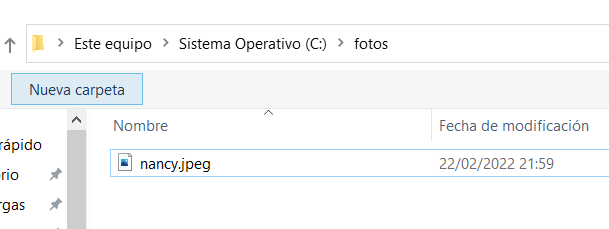
## **Subir Archivo (registro)**

El ejemplo para subir archivo se va a encontrar en el registro del nuevo usuario



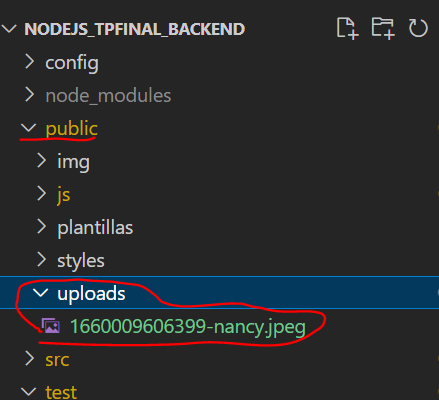




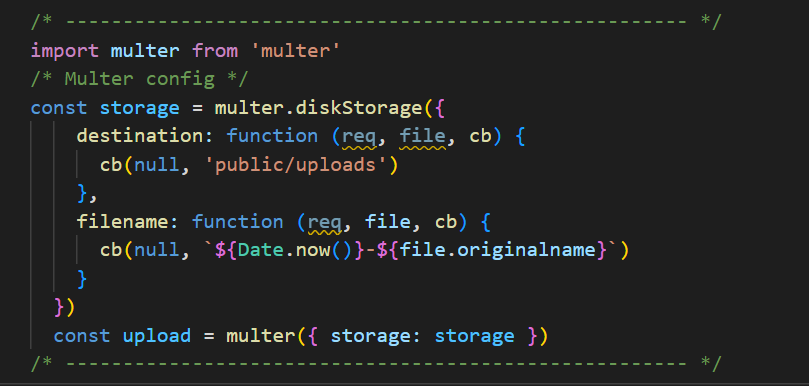




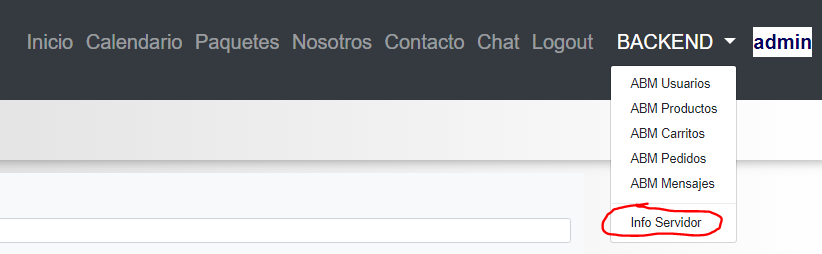




Se observa en el código como se usa multer



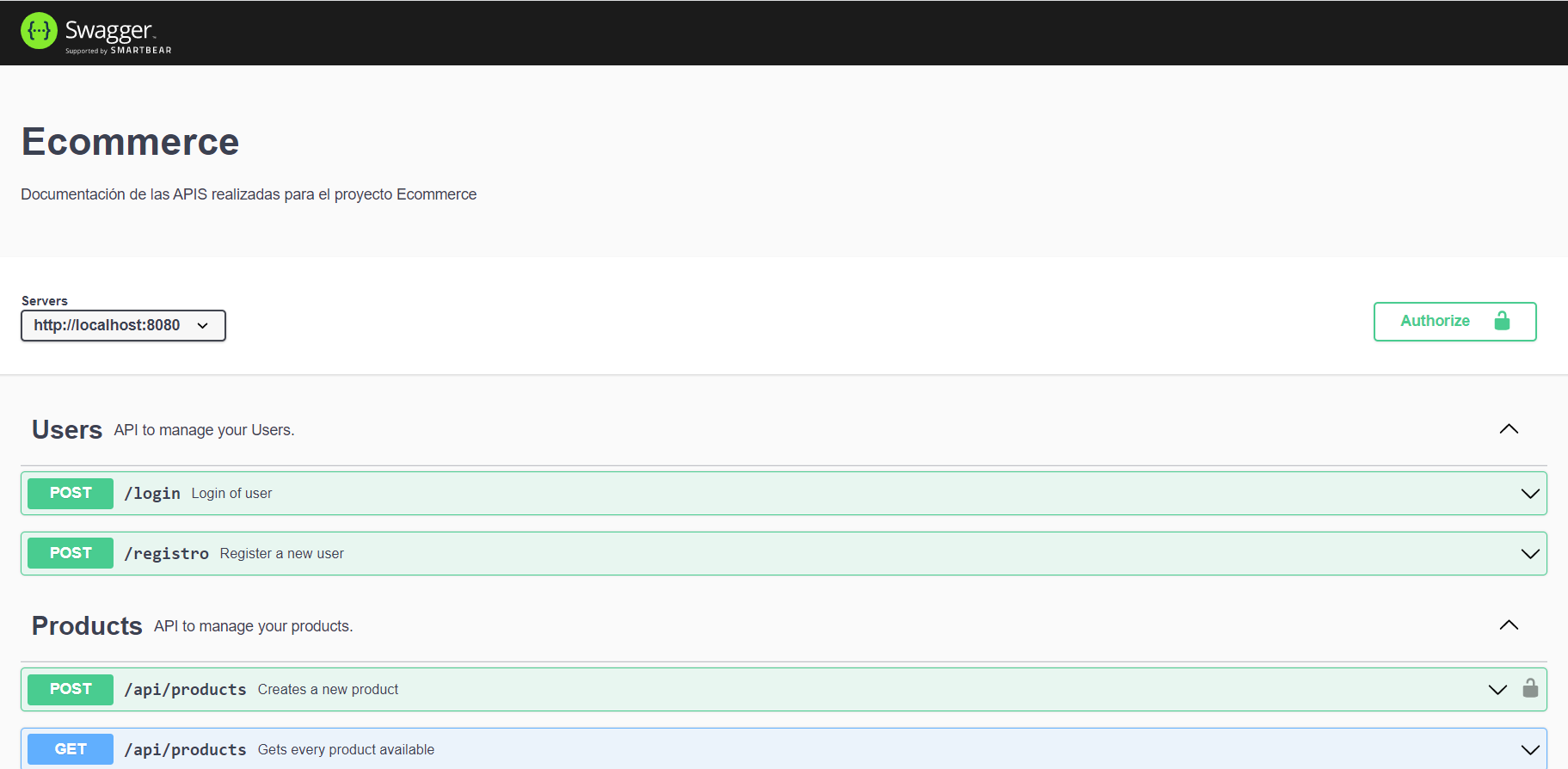
## **Información Servidor**

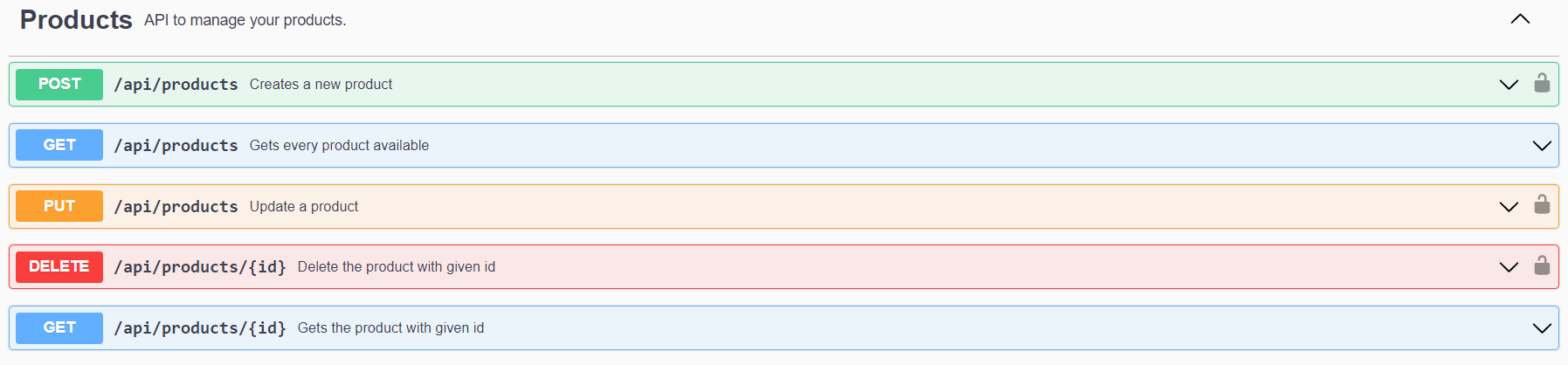


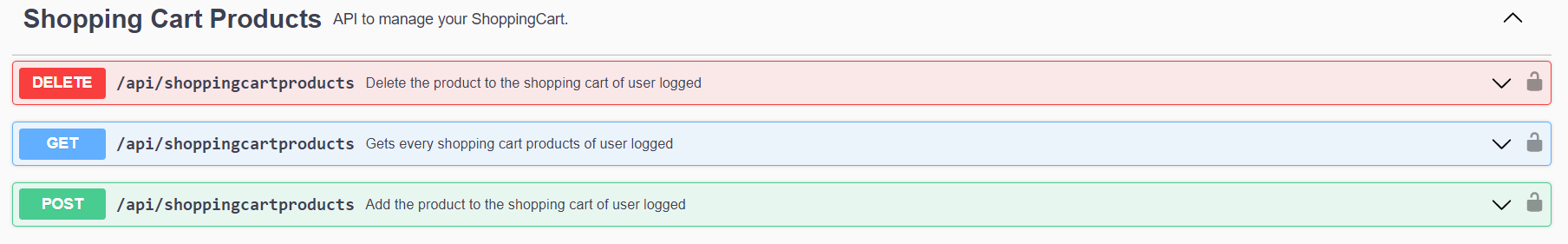


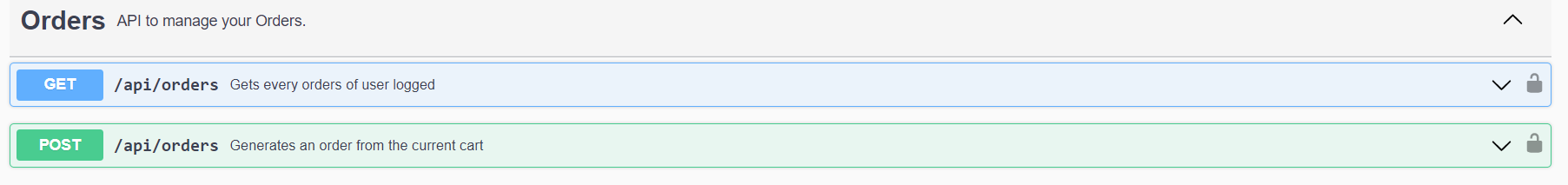
# Swagger

<http://localhost:8080/api/docs/>







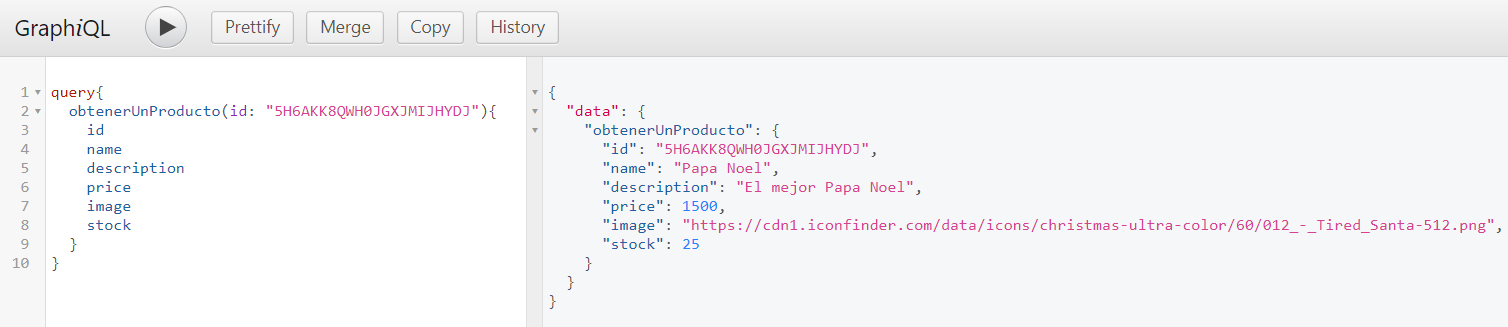




# Graphl

<http://localhost:8080/graphql>

Ejemplo:



# Testing



