Funcionalidades de la API:

Esta API está montada sobre un servidor local XAMP, sobre Laravel trabajada desde Visual Studio Code, y tiene las siguientes funcionalidades:

Se utilizaron las migraciones para generar la BD API\_CHILE y se utilizaron los seeders para generar los datos de pruebas, donde se registran en BD los principales artículos con las direcciones de las imágenes mandadas para la prueba técnica, que están físicamente en la carpeta **images** que está dentro de la carpeta **public**.

La aplicación realizada en Laravel tiene una batería de pruebas de 12 productos 3, roles y 3 usuarios. Para montarla completa hay que hacer lo siguiente:

php artisan migrate:fresh –seed: Corre las migraciones hacia la BD **API\_CHILE** y los seeder que es la batería de pruebas del sistema

php artisan passport:install –force: Es necesario correr el comando para que inserte en la BD las credenciales para poder trabajar con Laravel Passport u OAuTch 2.0 que es la autenticación que utiliza para la API

Autenticación: Se autentica el usuario que gestionará los artículos, las categorías y las solicitudes de compra

Login: <http://127.0.0.1:8000/api/login>

Parámetros:

email: [rsayu@nauta.cu](mailto:rsayu@nauta.cu)

password: password

Genera un token como resultado de la autenticación:

{

    "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiOiIzIiwianRpIjoiMzBmZmI5OTZkOWYyZWIwNjdkOWJlNWFlZDhjNjA4MmViYTQ1MWRhNTAzOGFhYTBjMDc1ZmE1ZTlhZDQwY2MyZTkyMTJjMmIwODJmYzI5ODciLCJpYXQiOjE2ODU0NjI5MTIuMjU1MzUsIm5iZiI6MTY4NTQ2MjkxMi4yNTUzNTMsImV4cCI6MTcxNzA4NTMxMi4yNDgyMTQsInN1YiI6IjEiLCJzY29wZXMiOltdfQ.f9j5sFssW9ADL2dr266TOyadmw8Ao5qYfj8S0-s7PH-A0Hw\_4VOUX3yfqmxWeMdDSYFg7XConDQ3OqveBXbyOF9pgiS-GQEIklc-4kydu1xGrtsCMOyk4FitY0u7EH\_Jz3H2H8bj2rhq0q9n9pHbxkSe7r8fPamWnM5DPZ6493DnrAGW1m7Xfc9f-BYK40OmvvhIPbsETYIvpc9UcHaMvJ\_rLRtzEXBg1UpR6uACfurKSZ\_sO0PwCRosgE4aSM-gVbGt0Wxgb-ucoWr\_nhRwEpvzVoQDeM-CMj0PSNa0slwPGLTwKV7mphIQW7FsCn5wiN4qmnMGNQLLBqvwcBImgpb9qC2uuBALt9S9wGVHj62OMkM4BrMEUH2WrzPGwEg9uxkWtMqzxb046TEkd\_pNGbXkiI98YieenGP4fQrFHw9hQSEFvsL7XEm4e\_VXJxWw07VOO8\_1bRANlerIODlVutzeet1VFW5ne49zsNksOyDROCqXDoc7Mz\_N5GEqsQQBbc0h584ClR94-HkWPuCcNOo3vAg3-wM-kCxeBkYfZ76Io-UAyiKwKIDrBWQ8xIFTTcSg4Hz0jasdrHVOCzmTs9DfI8ZJy-3ygUUbYDLTOg4pAqZ9CcI3R2isJaUSWAsjXYzdGBMxT\_mm6Reab0bv39lwDFtzLSd5-xTIPJ7zd2g"

}

Logout: <http://127.0.0.1:8000/api/logout>

En el postman en Authorization en Type: OAuth 2.0 y en Available Tokens: eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiOiIzIiwianRpIjoiMzBmZmI5OTZkOWYyZWIwNjdkOWJlNWFlZDhjNjA4MmViYTQ1MWRhNTAzOGFhYTBjMDc1ZmE1ZTlhZDQwY2MyZTkyMTJjMmIwODJmYzI5ODciLCJpYXQiOjE2ODU0NjI5MTIuMjU1MzUsIm5iZiI6MTY4NTQ2MjkxMi4yNTUzNTMsImV4cCI6MTcxNzA4NTMxMi4yNDgyMTQsInN1YiI6IjEiLCJzY29wZXMiOltdfQ.f9j5sFssW9ADL2dr266TOyadmw8Ao5qYfj8S0-s7PH-A0Hw\_4VOUX3yfqmxWeMdDSYFg7XConDQ3OqveBXbyOF9pgiS-GQEIklc-4kydu1xGrtsCMOyk4FitY0u7EH\_Jz3H2H8bj2rhq0q9n9pHbxkSe7r8fPamWnM5DPZ6493DnrAGW1m7Xfc9f-BYK40OmvvhIPbsETYIvpc9UcHaMvJ\_rLRtzEXBg1UpR6uACfurKSZ\_sO0PwCRosgE4aSM-gVbGt0Wxgb-ucoWr\_nhRwEpvzVoQDeM-CMj0PSNa0slwPGLTwKV7mphIQW7FsCn5wiN4qmnMGNQLLBqvwcBImgpb9qC2uuBALt9S9wGVHj62OMkM4BrMEUH2WrzPGwEg9uxkWtMqzxb046TEkd\_pNGbXkiI98YieenGP4fQrFHw9hQSEFvsL7XEm4e\_VXJxWw07VOO8\_1bRANlerIODlVutzeet1VFW5ne49zsNksOyDROCqXDoc7Mz\_N5GEqsQQBbc0h584ClR94-HkWPuCcNOo3vAg3-wM-kCxeBkYfZ76Io-UAyiKwKIDrBWQ8xIFTTcSg4Hz0jasdrHVOCzmTs9DfI8ZJy-3ygUUbYDLTOg4pAqZ9CcI3R2isJaUSWAsjXYzdGBMxT\_mm6Reab0bv39lwDFtzLSd5-xTIPJ7zd2g

Para los casos de uso de Usuario, Artículo, Categoría y Solicitud que es con el usuario autenticado:

En el postman en Authorization en Type: OAuth 2.0 y en Available Tokens: Aquí va el token que se genera en la autenticación.

**Caso de Uso Usuario:** Este registro de usuario se realiza con el objetivo de que el usuario es quien podrá gestionar los artículos, las categorías, y las solicitudes de las personas que agreguen al carrito los artículos.

Mostrar usuario (GET): <http://127.0.0.1:8000/api/users>

Registrar usuario (POST): <http://127.0.0.1:8000/api/users>

Parámetros:

name: Imara

email: [imara@nauta.cu](mailto:imara@nauta.cu)

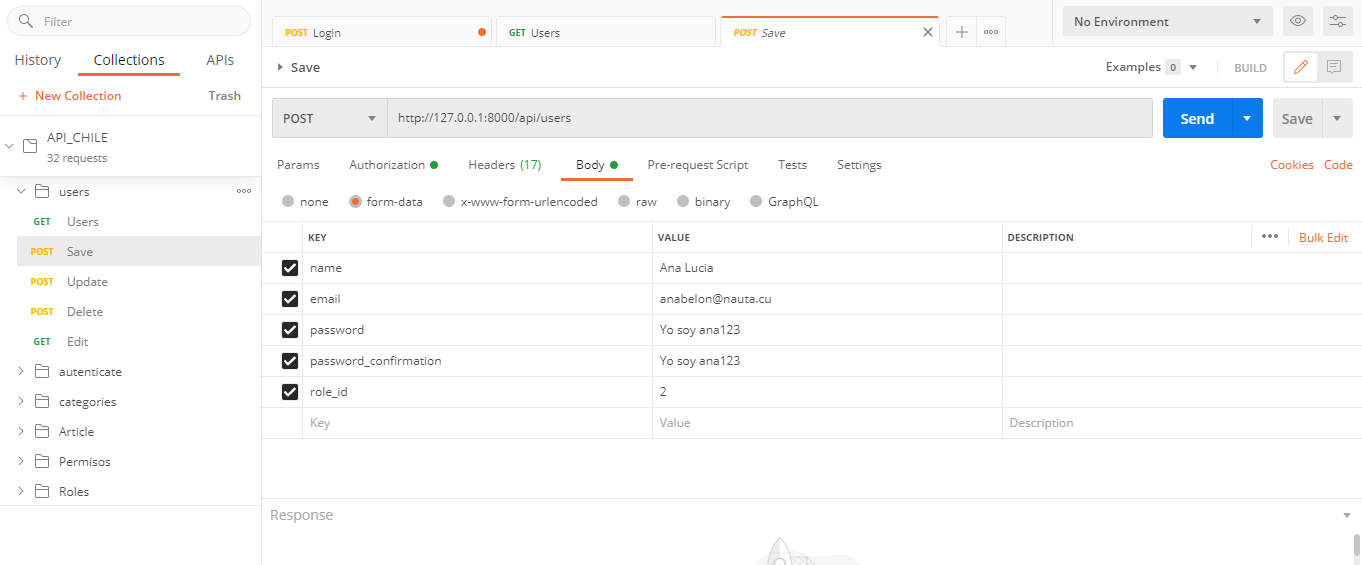
password: Yo soy imara123

password\_confirmation: Yo soy imara123

La contraseña debe tener números, letras, caracteres especiales y mayúsculas y de de haber mas de 8 caracteres.

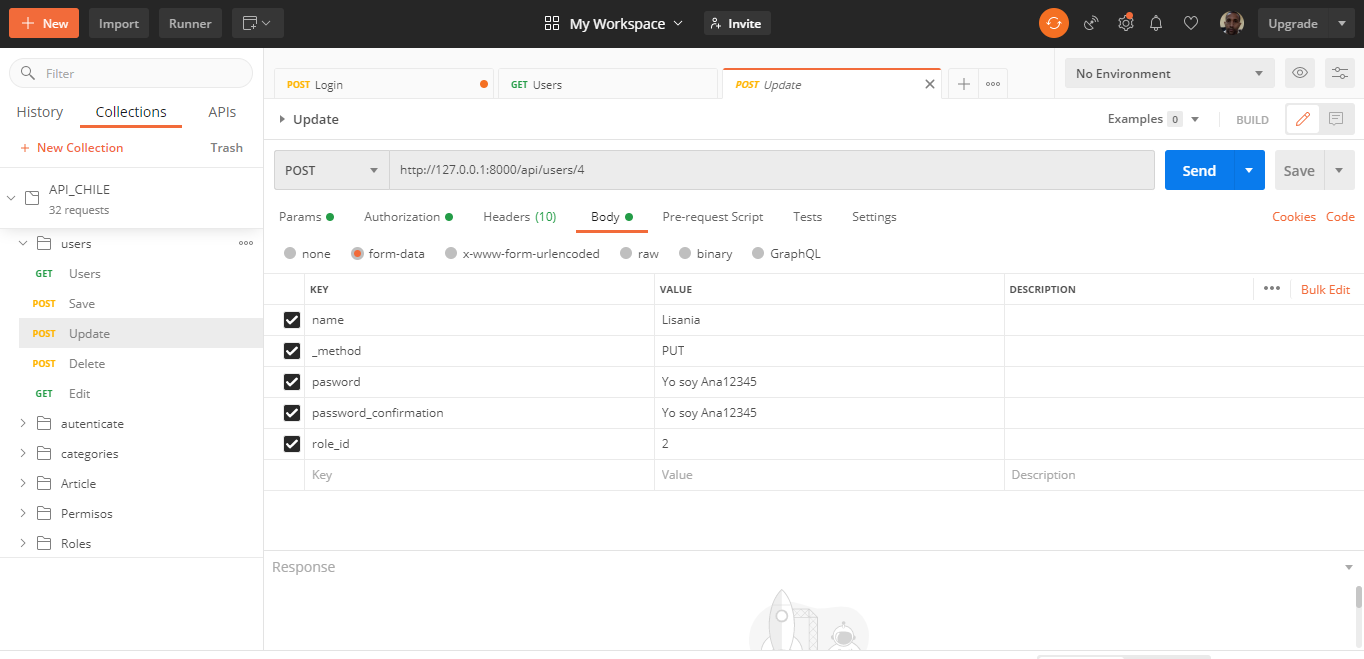
El correo no debe ser el mismo.

Estos errores no salen en el postmas pero si en una aplicación de front donde son tratados a través de mensajes.



Actualizar Usuario (POST a PUT): http://127.0.0.1:8000/api/users/4

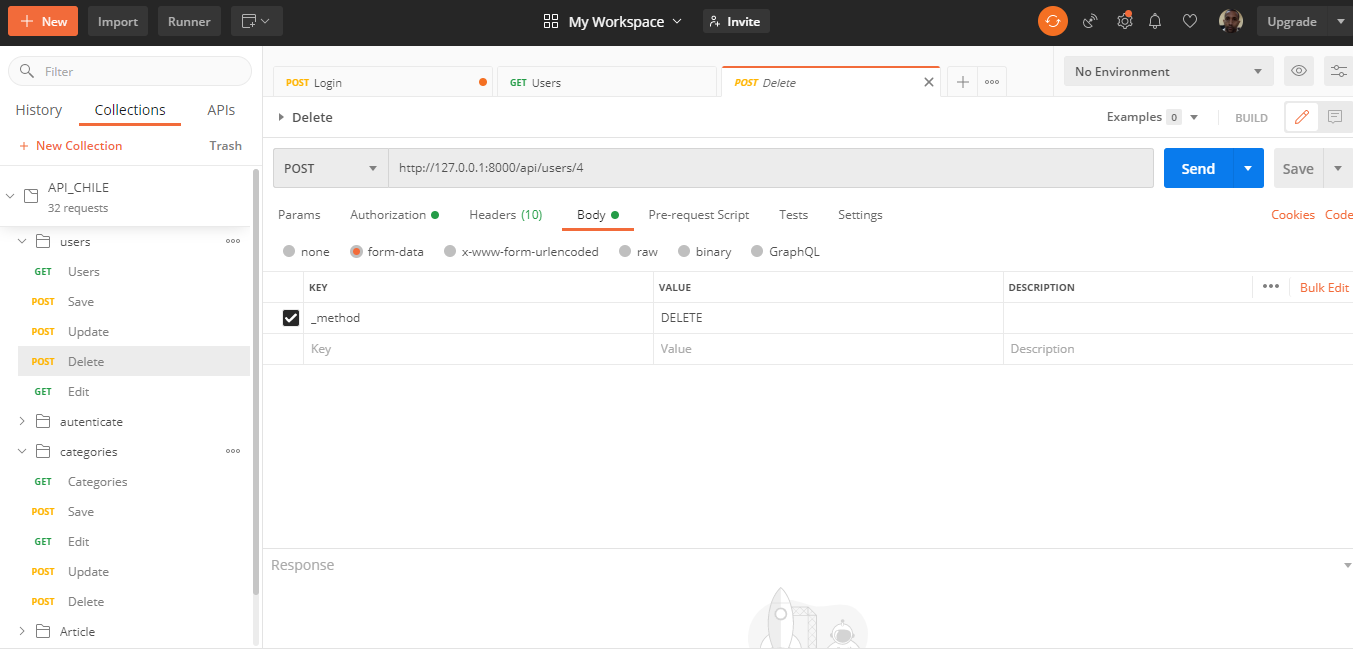
En este caso el usuario puede actualizar la contraseña o el nombre.



Editar Usuario (GET): <http://127.0.0.1:8000/api/users/2/edit>

No lleva parámetros puede ser el mostrar usuario.

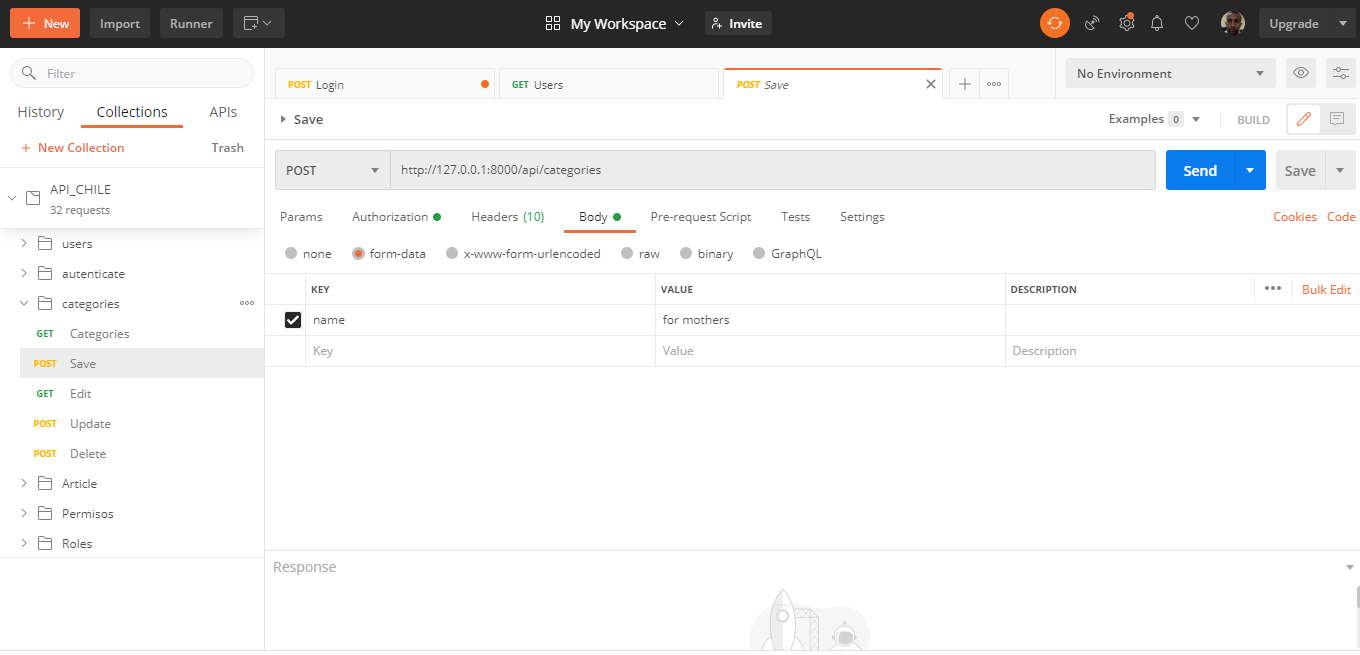
**Eliminar Usuario (POST a DELETE):** <http://127.0.0.1:8000/api/users/4>

****

**CASO DE USO CATEGORIAS:**

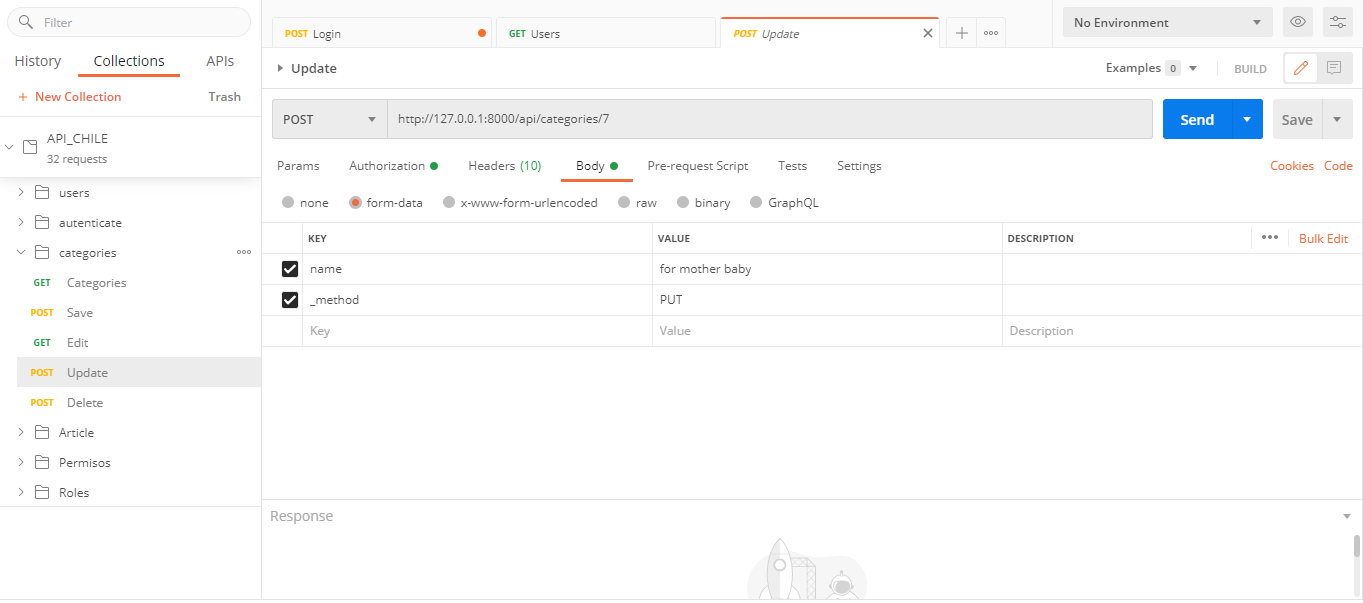
Mostrar Categorías (GET) <http://127.0.0.1:8000/api/categories>

Salvar Categorías (POST): <http://127.0.0.1:8000/api/categories>

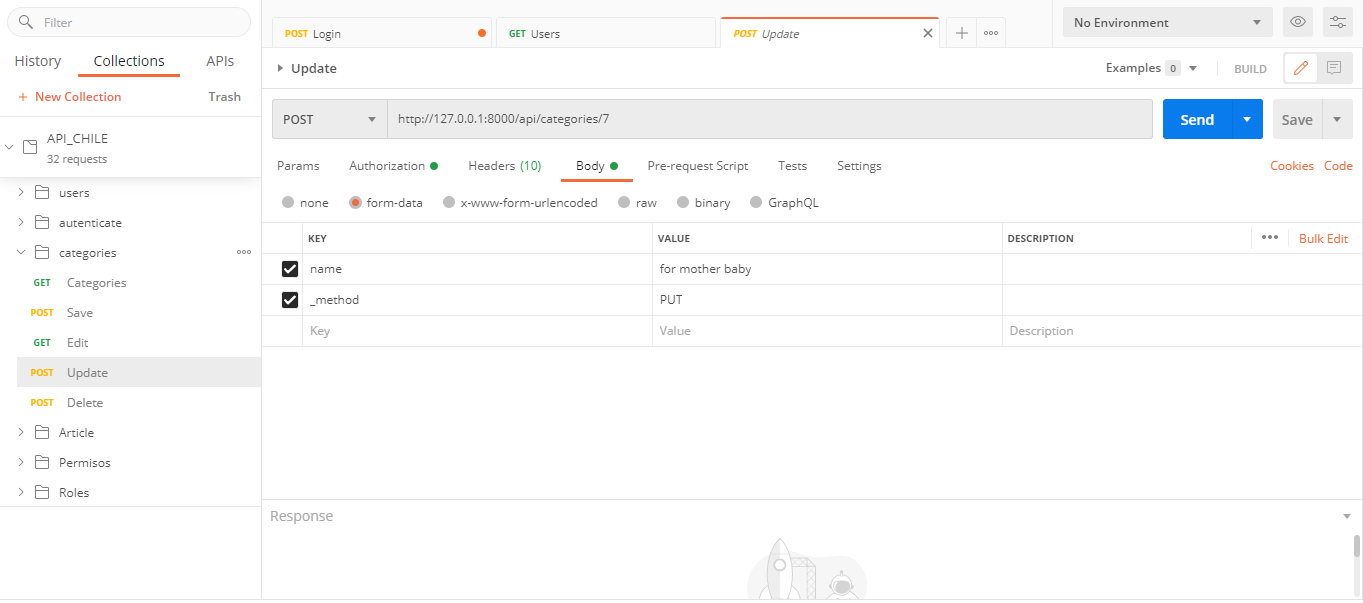


Editar Categoría (GET) <http://127.0.0.1:8000/api/categories/2/edit>

Actualizar Categoría (POST a PUT):



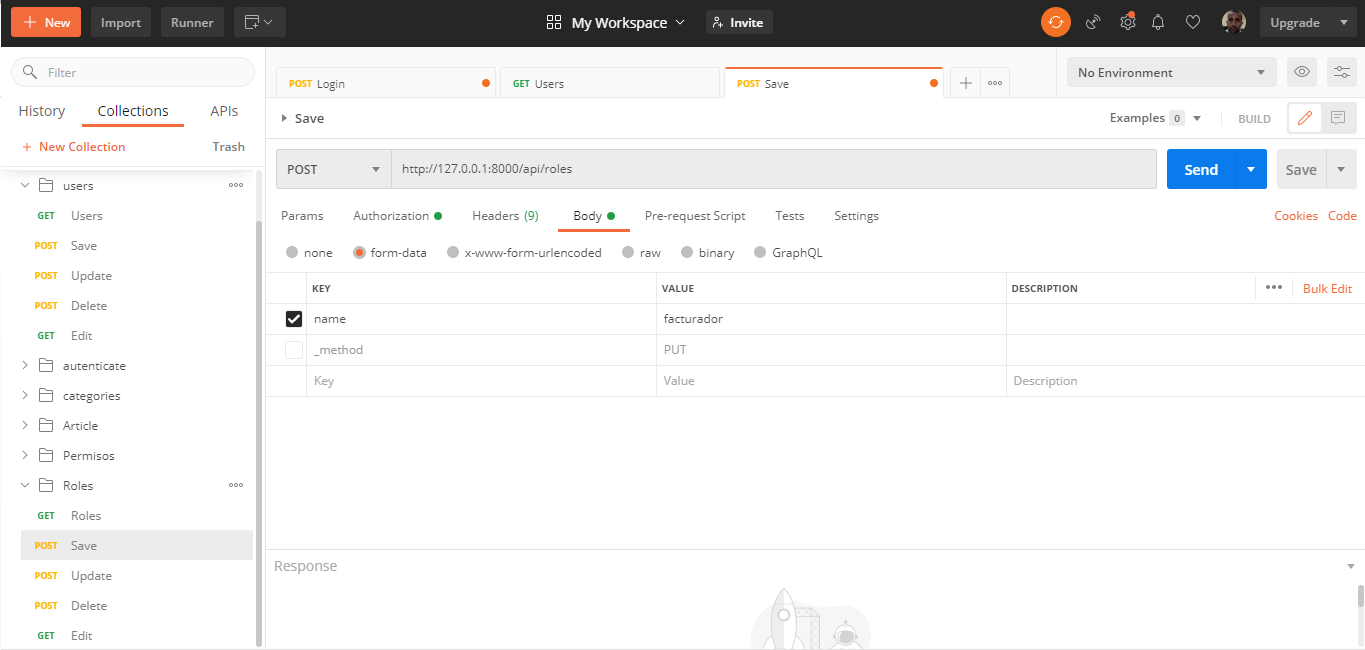
Eliminar Categorías(POST a DELETE) <http://127.0.0.1:8000/api/categories/7>



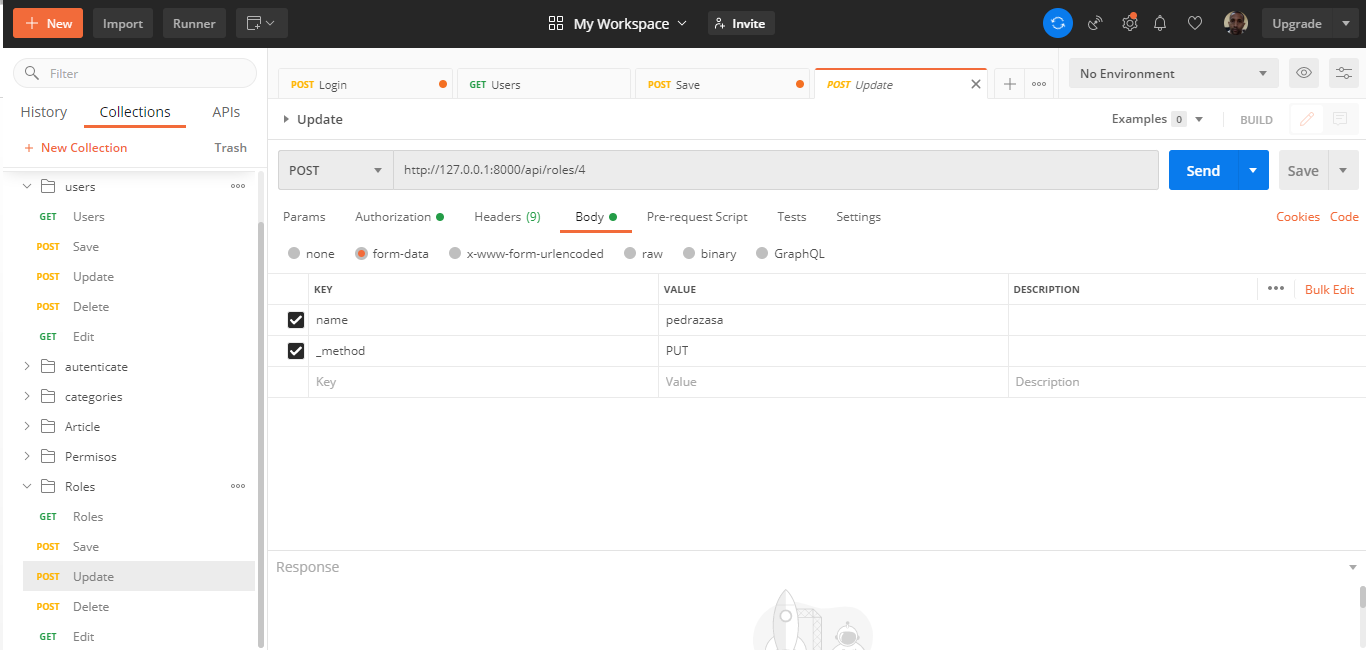
**Caso de uso Roles:**

Mostrar Roles (GET): <http://127.0.0.1:8000/api/roles>

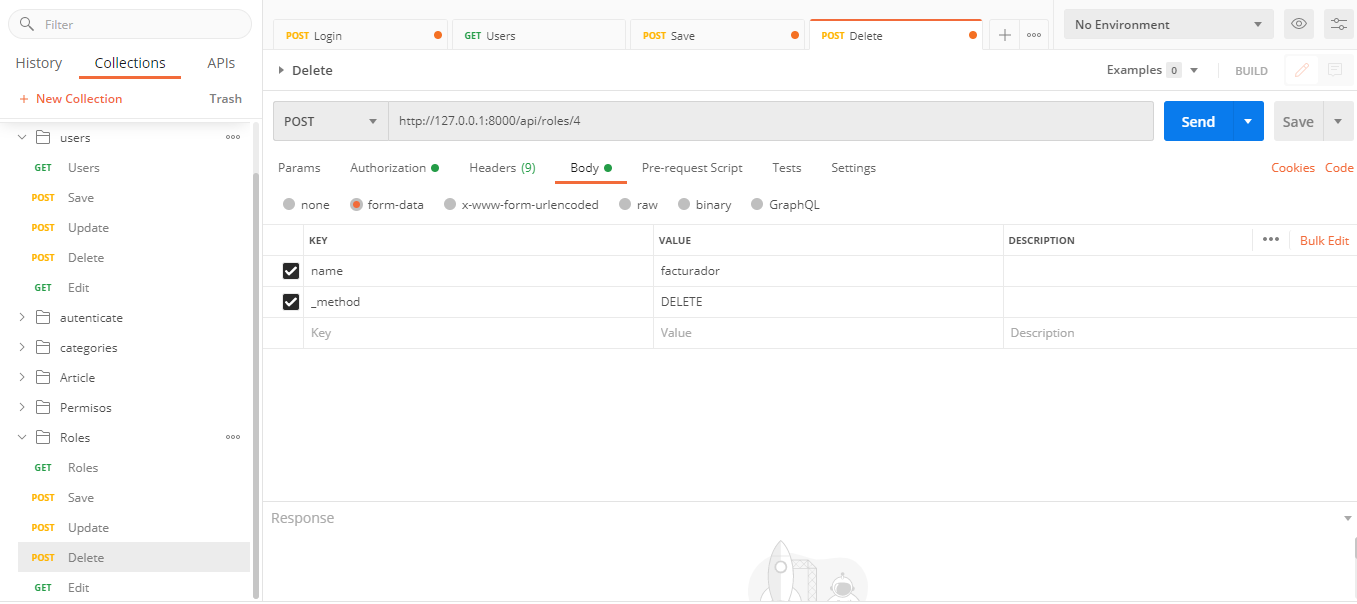
Salvar Roles (POST): <http://127.0.0.1:8000/api/roles>



Actualizar Roles (POST a PUT): <http://127.0.0.1:8000/api/roles/4>



Eliminar Roles (POST a DELETE): <http://127.0.0.1:8000/api/roles/4>

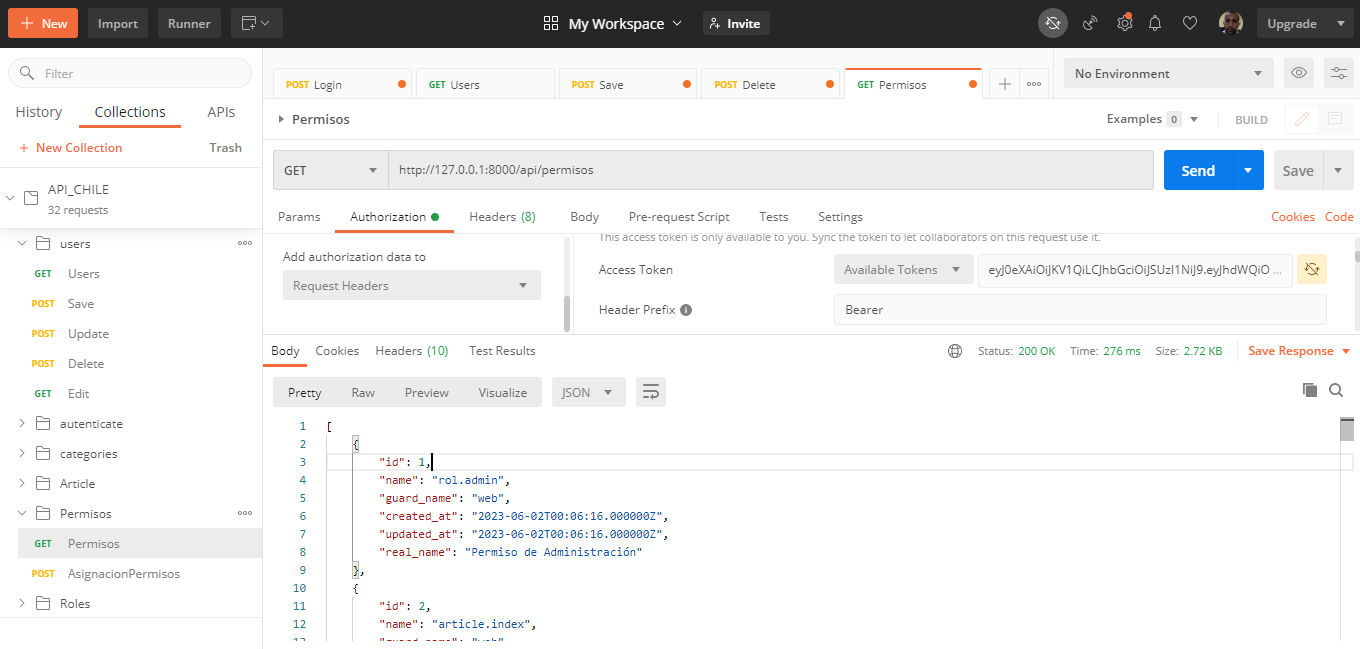


Editar Roles (GET): <http://127.0.0.1:8000/api/roles/3/edit>

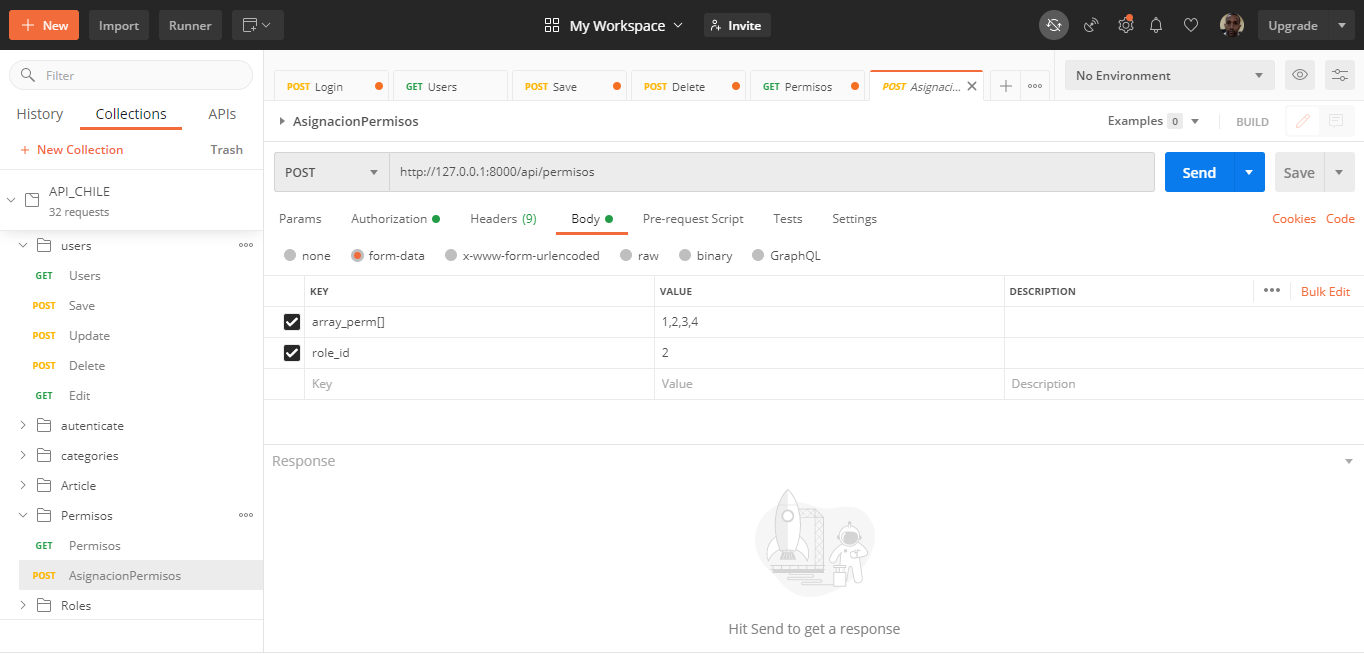
**Caso de Uso Permisos:**

Este caso de uso solo muestra los permisos y los asigna.

Mostrar Permisos (GET): <http://127.0.0.1:8000/api/permisos>



Asignar Permisos (POST):

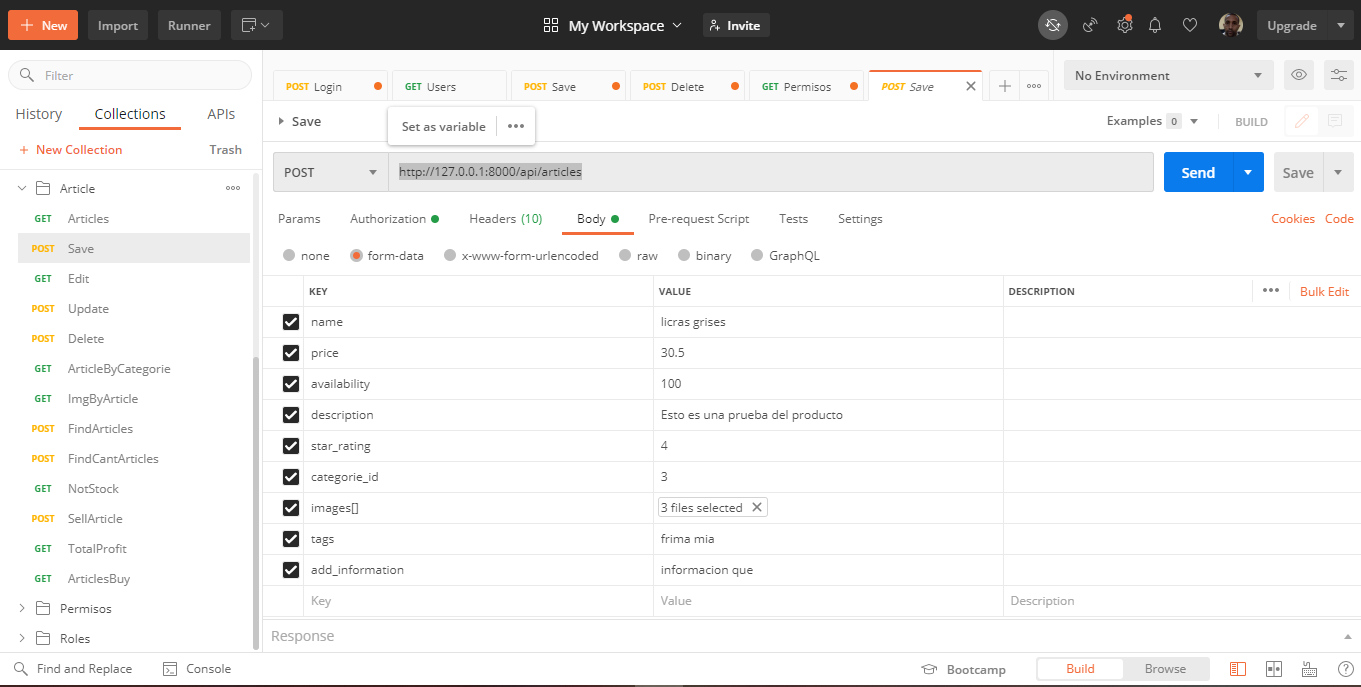


**Caso de Uso Artículos:**

Aquí están todas las funcionalidades pedidas de la prueba.

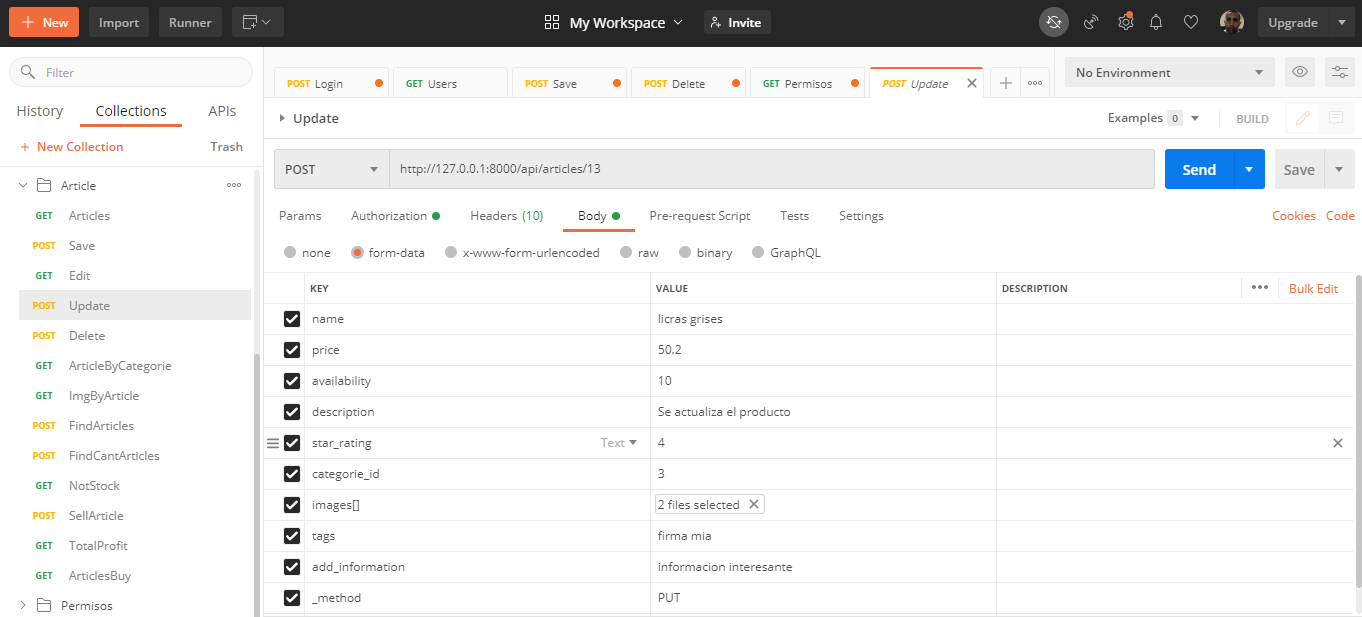
Mostrar Artículos (GET): <http://127.0.0.1:8000/api/articles>

Salvar Artículo (POST): <http://127.0.0.1:8000/api/articles>

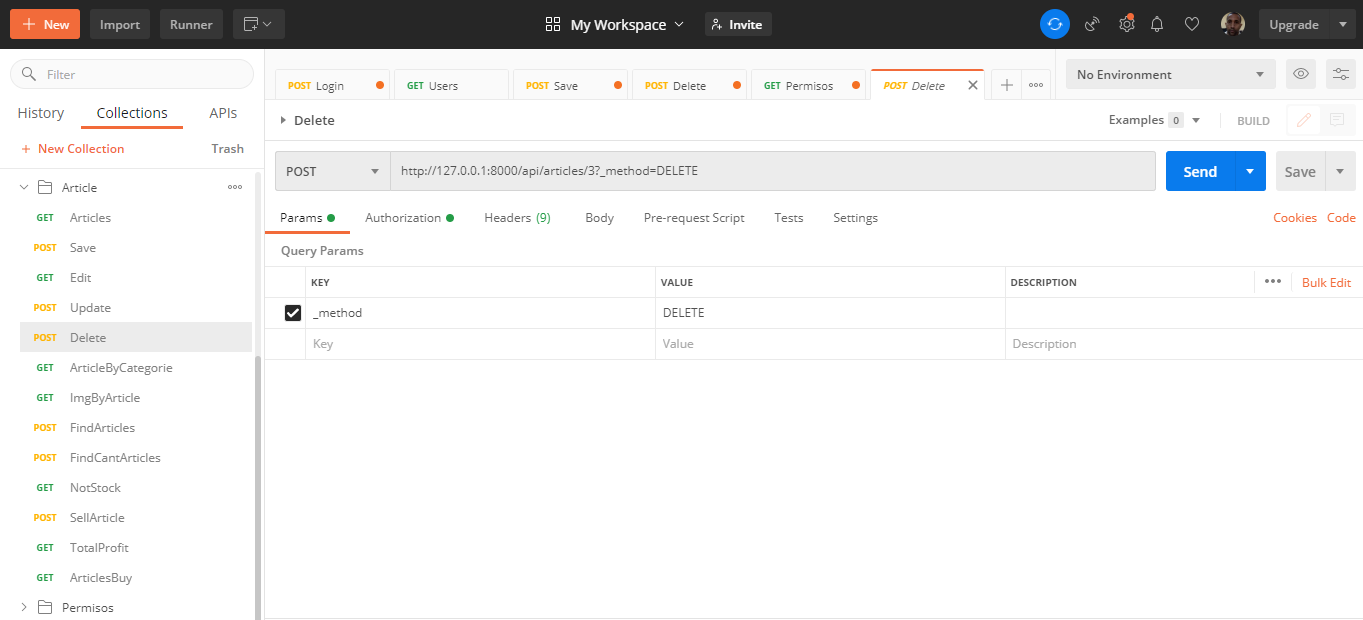


Editar artículo (GET): <http://127.0.0.1:8000/api/articles/4/edit>

Actualizar artículo (POST a PUT): <http://127.0.0.1:8000/api/articles/4/edit>



Eliminar Articulo (POST a PUT)



Articulo por categoría (GET): <http://127.0.0.1:8000/api/articles/bycategorie/1>

Imagen por Artículo (GET): <http://127.0.0.1:8000/api/articles/byarticle/13>

Buscar Artículos (POST): <http://127.0.0.1:8000/api/articles/findart>

Buscar Cantidad de artículos (POST): <http://127.0.0.1:8000/api/articles/findcantart>

Artículos que se quedaron sin disponibilidad (GET): <http://127.0.0.1:8000/api/articles/nostock>

Comprar artículo (POST): <http://127.0.0.1:8000/api/articles/sellart/4>

Ganancia Total (GET): <http://127.0.0.1:8000/api/articles/totalprofit>

Artículos Vendidos (GET): <http://127.0.0.1:8000/api/articles/articlesbuy>