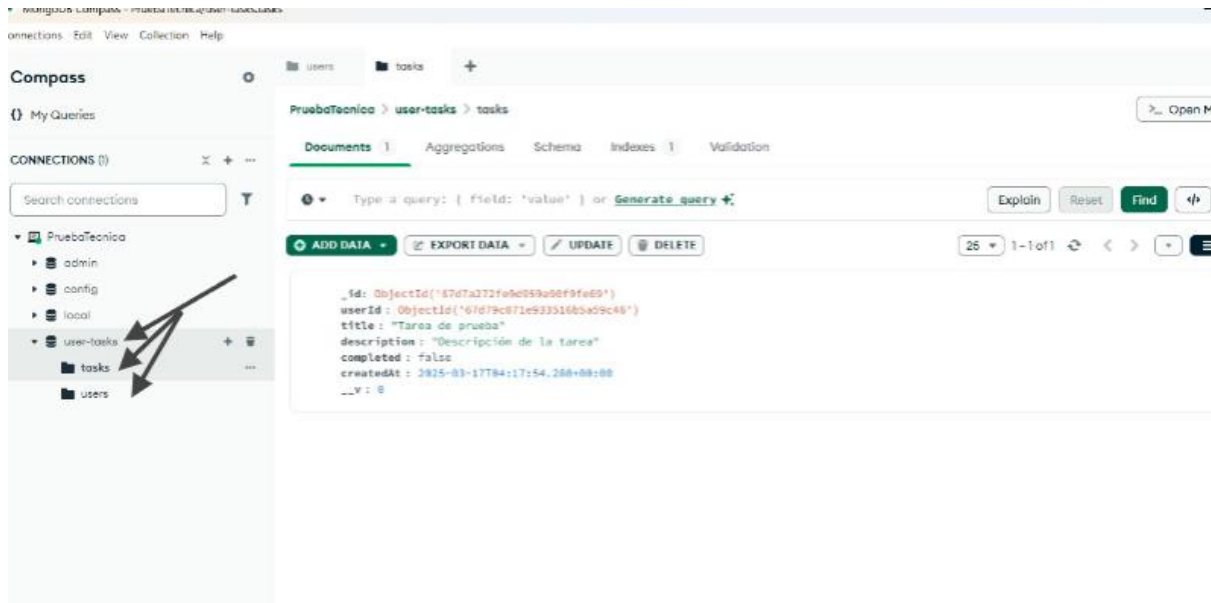


Brayan David Pulido Romero

Documentacion

Base de datos :

Para la base de datos debemos de usar mongo db, tener una base de datos creada con estas dos colecciones dentro :



Microservicio : microservicio-registro

Para iniciar el proyecto abrimos una consola dentro de nuestro proyecto y escribimos

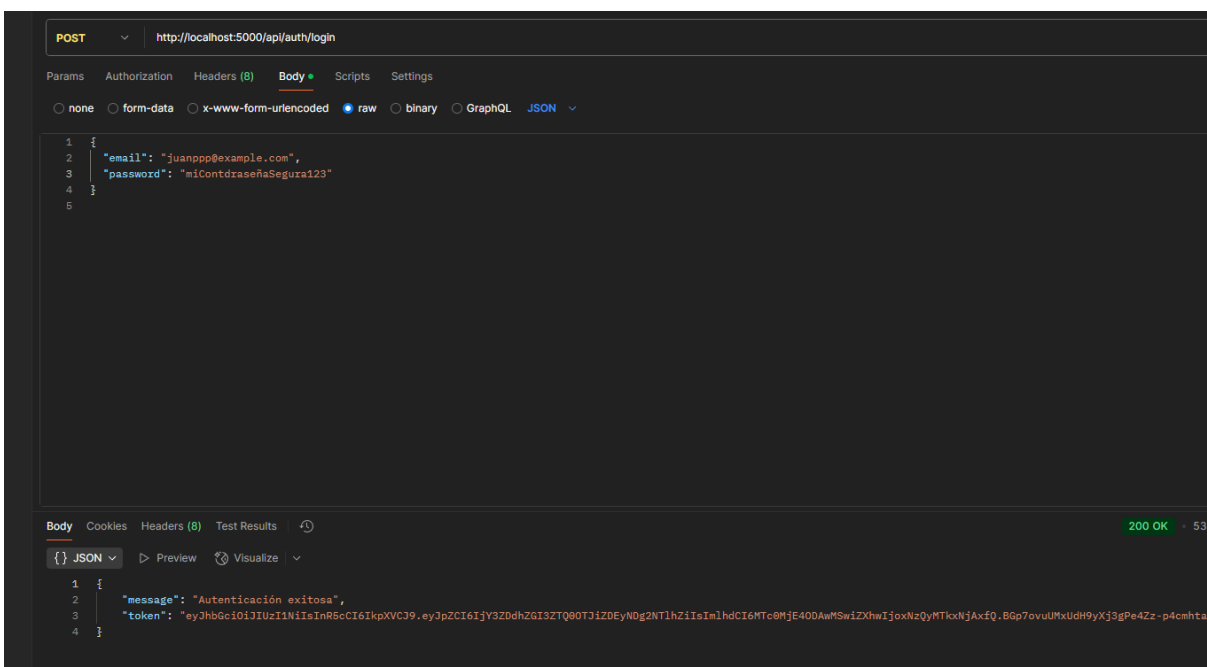
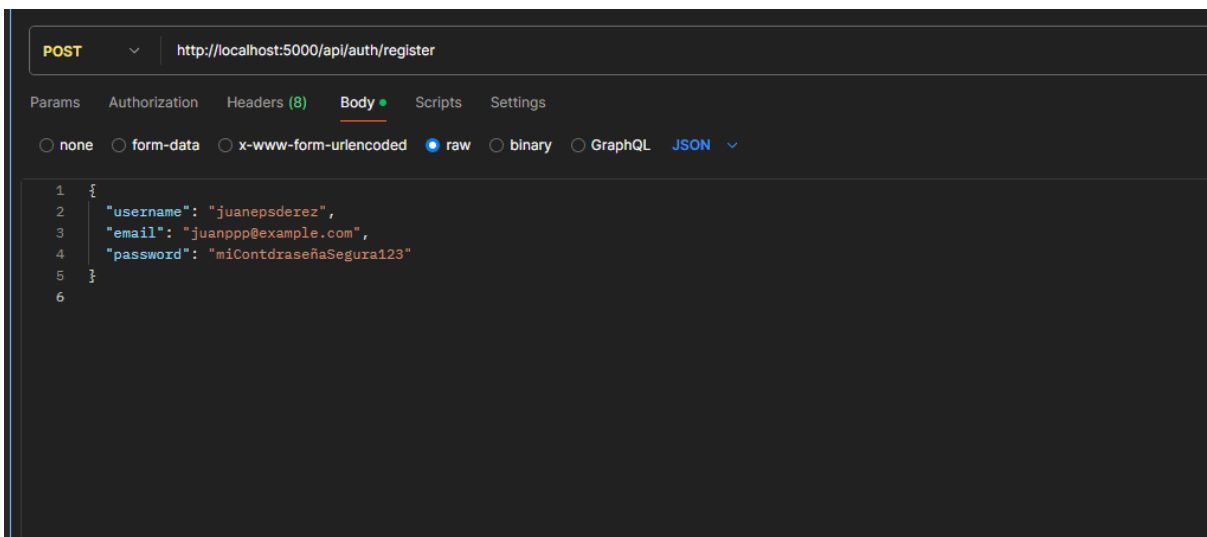
npm i



despues de ejecutar npm i para instalar las dependencias ejecutamos el comando en la consola node server.js para correr la aplicacion

```
braya@Brayan MINGW64 ~/Downloads/TC/microservicio-registro
$ node server.js
Servidor corriendo en el puerto 5000
Conectado a MongoDB
```

Servicios :



Microservicio : microservicio-tareas

Para iniciar el proyecto abrimos una consola dentro de nuestro proyecto y escribimos

npm i

```
braya@Brayan MINGW64 ~/Downloads/TC/microservicio-tareas
$ npm i
```

despues de ejecutar npm i para instalar las dependencias ejecutamos el comando en la consola node server.js para correr la aplicacion

```
braya@Brayan MINGW64 ~/Downloads/TC/microservicio-tareas
$ node server.js
Servidor corriendo en el puerto 5001
Conectado a MongoDB
```

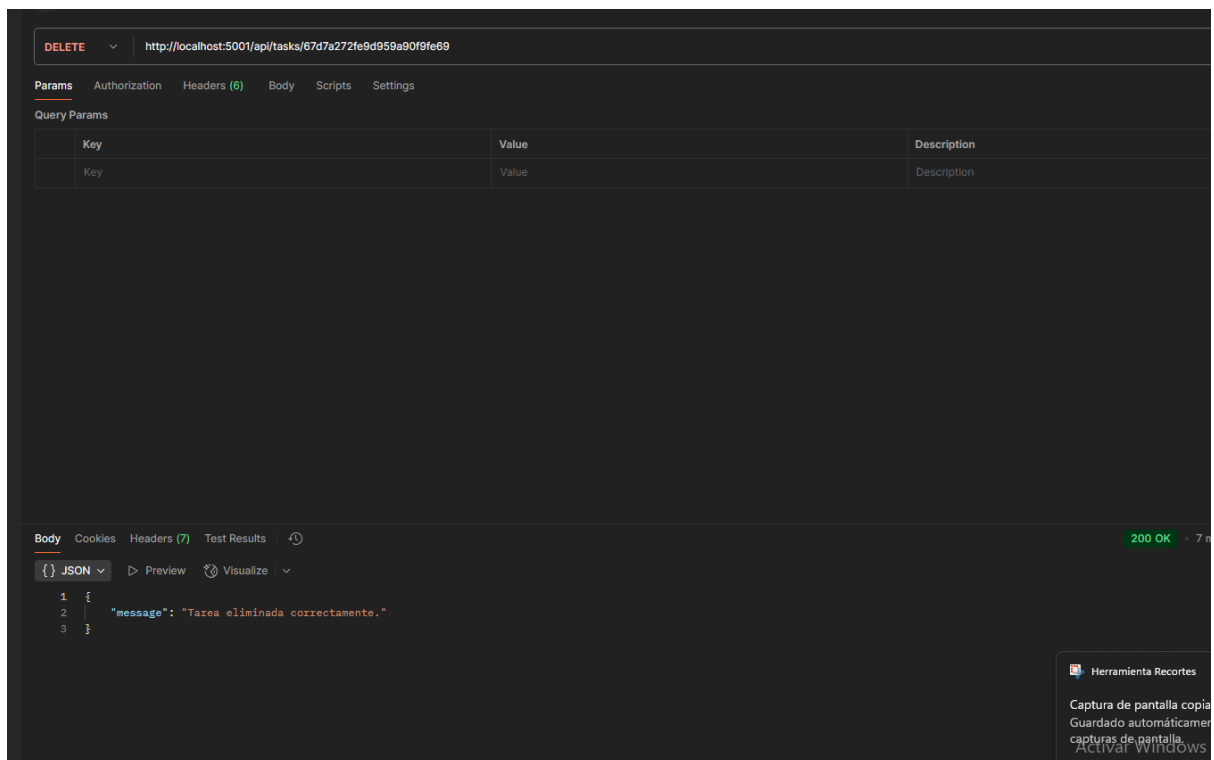
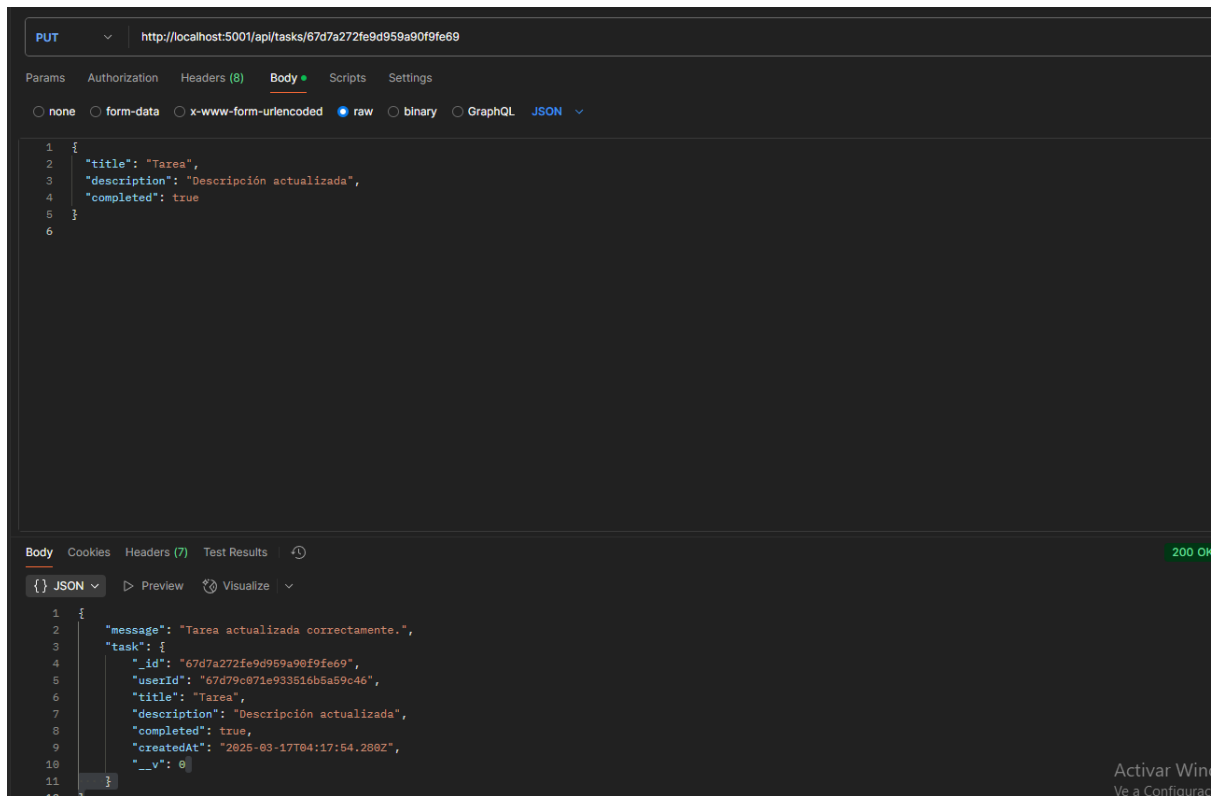
Servicios:

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://localhost:5001/apl/tasks/
- Body Type:** raw (selected)
- Request Body (JSON):**

```
{  "userId": "67d79c071e933516b5a59c46",  "title": "Tarea de prueba",  "description": "Descripción de la tarea"}
```
- Response Status:** 201 Created (21 ms)
- Response Body (JSON):**

```
{  "message": "Tarea creada correctamente.",  "task": {    "userId": "67d79c071e933516b5a59c46",    "title": "Tarea de prueba",    "description": "Descripción de la tarea",    "completed": false,    "_id": "67d7a272fe9d959a98f9fe69",    "createdAt": "2025-03-17T04:17:54.268Z",    "__v": 0  }  }}
```



Frontend : app-frontend

Para iniciar el proyecto abrimos una consola dentro de nuestro proyecto y escribimos

npm i

```
braya@Brayan MINGW64 ~/Downloads/TC/app-frontend (master)
○ $ npm i
```

Posterior a ello iniciamos el proyecto con el comando

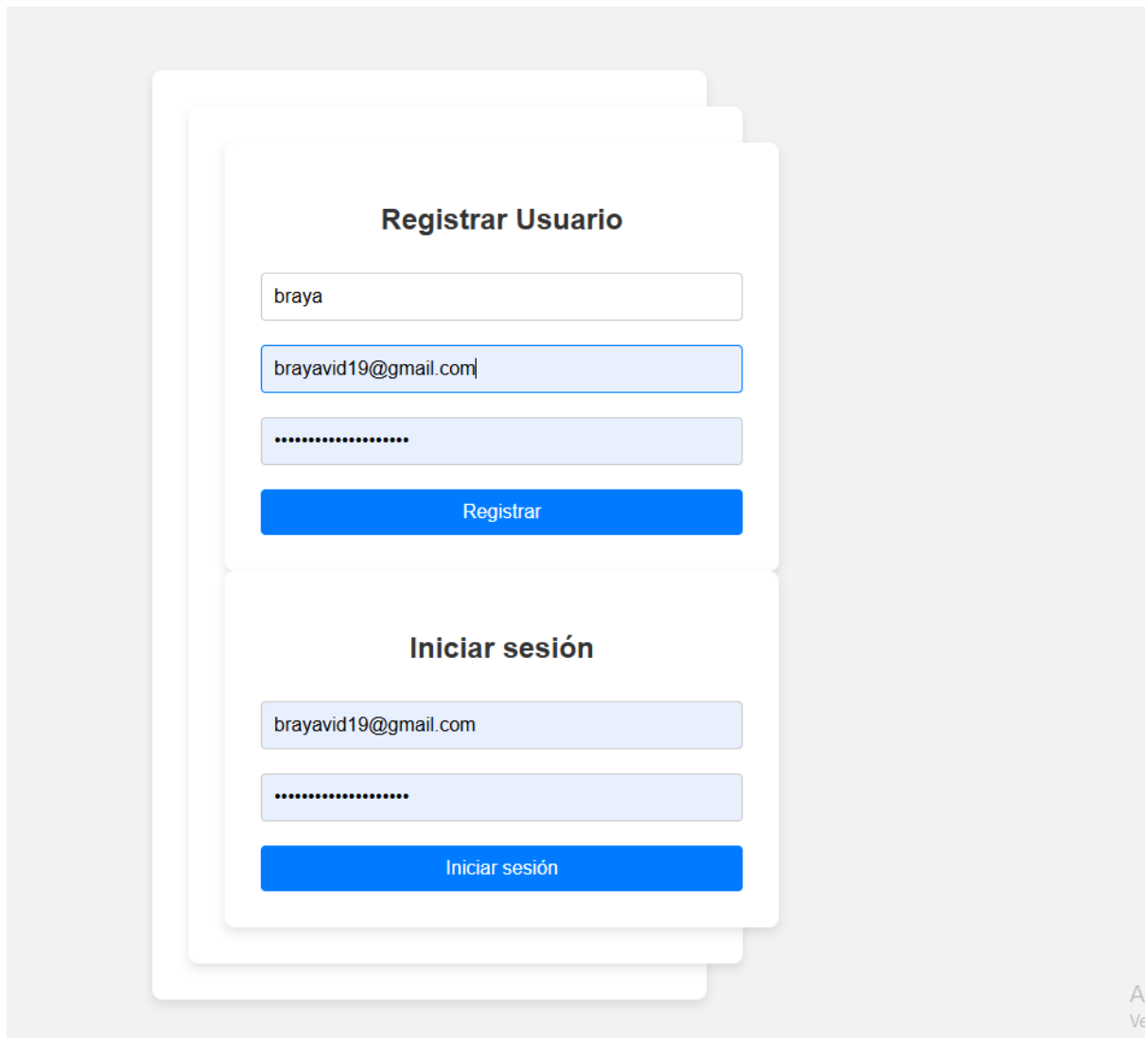
npm start

```
webpack compiled successfully
braya@Brayan MINGW64 ~/Downloads/TC/app-frontend (master)
○ $ npm start
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Compiled successfully!
You can now view app-frontend in the browser.
Local: http://localhost:3000
Compiled successfully!
You can now view app-frontend in the browser.
Local: http://localhost:3000
On Your Network: http://192.168.0.13:3000
Note that the development build is not optimized.
To create a production build, use npm run build.
webpack compiled successfully
```

para que el proyecto funcione correctamente tenemos que asegurarnos de tener los dos microservicios backend corriendo localmente

Pantalla de registrar usuario y login :



The image displays two overlapping white rectangular forms on a light gray background. The top form is titled "Registrar Usuario" and contains three input fields: a text field with "braya", an email field with "brayavid19@gmail.com", and a password field with masked characters. Below these is a blue button labeled "Registrar". The bottom form is titled "Iniciar sesión" and contains two input fields: an email field with "brayavid19@gmail.com" and a password field with masked characters. Below these is a blue button labeled "Iniciar sesión".

Registrar Usuario

braya

brayavid19@gmail.com

.....

Registrar

Iniciar sesión

brayavid19@gmail.com

.....

Iniciar sesión

pantalla para ver las tareas creadas y crear nuevas tareas

Tareas

- prueba

Eliminar

- PRUEBA 2

Eliminar

- TAREA DE LIBROS

Eliminar

Crear Tarea

Título

Descripción

Crear

Tareas

- prueba

Eliminar

- Tarea de prueba

Eliminar

Crear Tarea

Título

Descripción

Crear