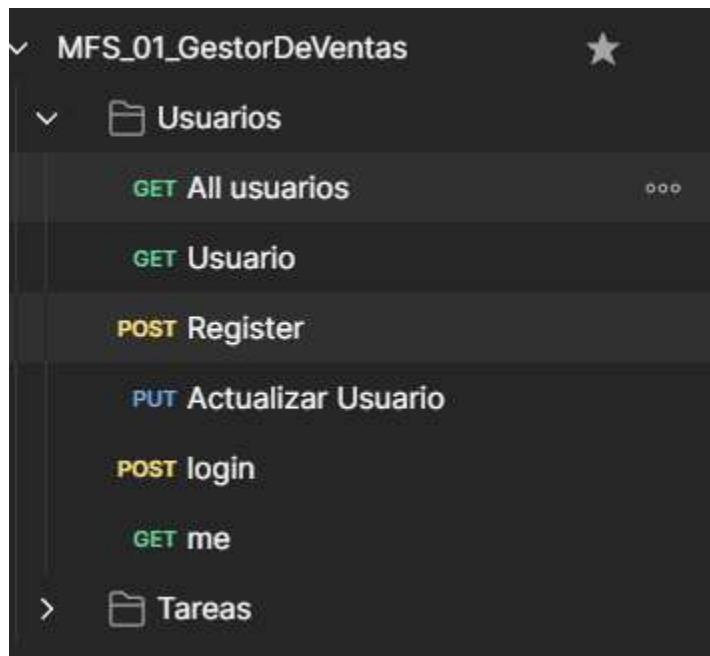


Repositorio	https://github.com/joseluisdaza/MFS_01_ProyectoFinal Contiene Backend y front end bajo diferentes carpetas
Backend Despliegue	
Frontend Despliegue	

Ejemplos de postman

Usuarios



Registrar un nuevo usuario

The screenshot shows a REST client interface with a POST request to `http://localhost:3000/api/auth/register`. The request body is a JSON object with the following fields:

```
1 {  
2   "Nombre": "prueba",  
3   "Correo": "b@test.com",  
4   "Password": "abcd"  
5 }
```

The client is configured with the following options:

- Method: POST
- URL: `http://localhost:3000/api/auth/register`
- Body Type: raw (selected)
- Content Type: JSON (selected)

The response body is also shown in JSON format:

```
1 {  
2   "name": "prueba",  
3   "email": "b@test.com"  
4 }
```

Obtener todos los usuarios

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

- URL:** `http://localhost:3000/api/auth`
- Method:** `GET`
- Headers:** 6 hidden
- Body:** JSON
- Test Results:** 8

The JSON response is displayed in a code editor with line numbers 1 through 12:

```
1  [  
2    {  
3      "id": 4,  
4      "Nombre": "prueba",  
5      "Correo": "a@test.com"  
6    },  
7    {  
8      "id": 6,  
9      "Nombre": "prueba",  
10     "Correo": "b@test.com"  
11   }  
12 ]
```

Obtener usuario por ID

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

- URL:** `http://localhost:3000/api/auth/4`
- Method:** GET
- Authorization:** Bearer Token
- Response Body (JSON):**

```
1 {
2   "id": 4,
3   "Nombre": "prueba",
4   "Correo": "a@test.com"
5 }
```

Login

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

- URL:** `http://localhost:3000/api/auth/login`
- Method:** `POST`
- Body Type:** `JSON`
- Request Body:**

```
{  "Correo": "a@test.com",  "Password": "abcd"}
```
- Response Status:** `200 OK` (59 ms)
- Response Body (JSON):**

```
{  "message": "Login exitoso",  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6NCwiYWQiOiJ0eXNzQyNjQ1MTc3LCJleHAiOiJlE3NDI2NDg3Nzd9.5R_rsI8dNgyha_ud2p-2N4n-c290hy_Q1KEN8Ou10dw"}
```

Me Con token valido

The screenshot shows a Postman interface for a GET request to `http://localhost:3000/api/auth/me`. The **Authorization** tab is selected, showing a **Bearer Token** type. A warning message states: "Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables." The token field contains a long alphanumeric string. The **Body** tab is also visible, showing a JSON response with the following structure:

```
1 {
2   "id": 4,
3   "Nombre": "Actualizado01",
4   "Correo": "a@test.com"
5 }
```

The status bar at the bottom indicates a **200 OK** response with a 50 ms duration and 322 B of data.

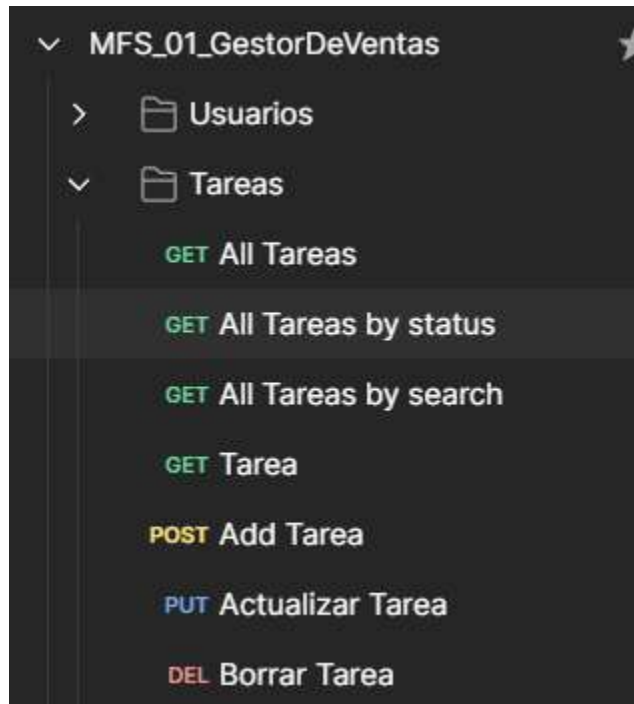
Me con token invalido

The screenshot shows a Postman interface for a GET request to `http://localhost:3000/api/auth/me`. The **Authorization** tab is selected, showing a **Bearer Token** type. A warning message states: "Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables." The token field contains a long alphanumeric string. The **Body** tab is also visible, showing a JSON response with the following structure:

```
1 {
2   "message": "Autenticación fallida."
3 }
```

The status bar at the bottom indicates a **401 Unauthorized** response with a 6 ms duration and 314 B of data.

TAREAS



GET ALL

MFS_01_GestorDeVentas / Tareas / All Tareas

GET ▼ http://localhost:3000/api/tasks

Params **Authorization** Headers (7) Body Scripts Settings

Auth Type

Bearer Token ▼

Heads up! These parameters hold sensitive data. To keep this data secure while working in a co recommend using variables. Learn more about [variables](#).

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...

Body Cookies Headers (8) Test Results ↻ **200 OK** - 53 ms - 55

{} JSON ▼ ▶ Preview 🔗 Visualize ▼

```
1  [
2    {
3      "id": 5,
4      "Titulo": "titulo modificado",
5      "FechaLimite": "2025-04-01T00:00:00.000Z",
6      "Descripcion": "descripcion modificada",
7      "Estado": "pendiente",
8      "UsuarioId": 4
9    },
10   {
11     "id": 14,
12     "Titulo": "Jose Luis DAza",
13     "FechaLimite": "2025-03-22T11:40:43.876Z",
14     "Descripcion": "Pao",
15     "Estado": "en progreso",
16     "UsuarioId": 4
17   }
18 ]
```


GET filter by status

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

- URL:** `http://localhost:3000/api/tasks?status=pendiente`
- Method:** GET
- Authorization:** Bearer Token. The token value is `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...`
- Response:** 200 OK, 53 ms, 421 B. The response body is a JSON object:

```
{  "id": 5,  "titulo": "titulo modificado",  "fechaLimite": "2025-04-01T00:00:00.000Z",  "descripcion": "Descripcion modificada",  "estado": "pendiente",  "usuarioId": 4}
```

Get TASK BY SEARCH

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

- URL:** `http://localhost:3000/api/tasks?search=modif`
- Method:** GET
- Authorization:** Bearer Token. The token value is `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...`
- Response:** 200 OK, 180 ms, 421 B. The response body is a JSON object:

```
{  "id": 5,  "titulo": "titulo modificado",  "fechaLimite": "2025-04-01T00:00:00.000Z",  "descripcion": "Descripcion modificada",  "estado": "pendiente",  "usuarioId": 4}
```

GET TASK BY ID

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

- URL Bar:** Method **GET** and URL **http://localhost:3000/api/tasks/5**.
- Tabs:** Params, **Authorization** (active), Headers (7), Body, Scripts, Settings.
- Authorization Section:**
 - Auth Type:** Bearer Token (selected from a dropdown).
 - Token:** A text input field.
 - Info:** "Heads up! These parameters recommend using variables".
 - Description:** "The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization."
- Body Section:**
 - Body:** (active tab)
 - Format:** JSON (selected from a dropdown).
 - Actions:** Preview, Visualize.
 - Response:**

```
1  {
2    "title": "titulo modificado",
3    "description": "Descripcion modifcada",
4    "status": "pendiente",
5    "dueDate": "2025-04-01T00:00:00.000Z"
6  }
```

GET TASK BY ID QUE NO PERTENECE AL USUARIO

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

- URL Bar:** Shows the method **GET** and the URL `http://localhost:3000/api/tasks/3`.
- Authorization Tab:** Selected, showing **Auth Type** as **Bearer Token**. A note states: "The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization." A **Token** input field is present.
- Body Tab:** Selected, showing the response body in **JSON** format:

```
{
  "message": "Tarea no encontrada."
}
```

Add Tarea

HTTP MFS_01_GestorDeVentas / Tareas / Add Tarea

POST ▼ http://localhost:3000/api/tasks

Params Authorization ● Headers (9) **Body** ● Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼

```
1 {
2   "title": "Ejemplo de titulo xyz",
3   "description": "Esta es la Descripcion del articulo",
4   "status": "pendiente",
5   "dueDate": "2025-03-20"
6 }
```

Body Cookies Headers (8) Test Results ↺

{ } JSON ▼ ▶ Preview 🔗 Visualize ▼

```
1 {
2   "message": "Tarea creada exitosamente.",
3   "task": {
4     "id": 16,
5     "title": "Ejemplo de titulo xyz",
6     "description": "Esta es la Descripcion del articulo",
7     "status": "pendiente",
8     "dueDate": "2025-03-20T00:00:00.000Z",
9     "userId": 4
10   }
11 }
```

Actualizar Tarea

The screenshot shows a REST client interface with a PUT request to `http://localhost:3000/api/tasks/5`. The 'Body' tab is selected, and the request body is a JSON object. The response body is also shown in the 'Body' tab, indicating a successful update.

Request:

```
PUT http://localhost:3000/api/tasks/5
```

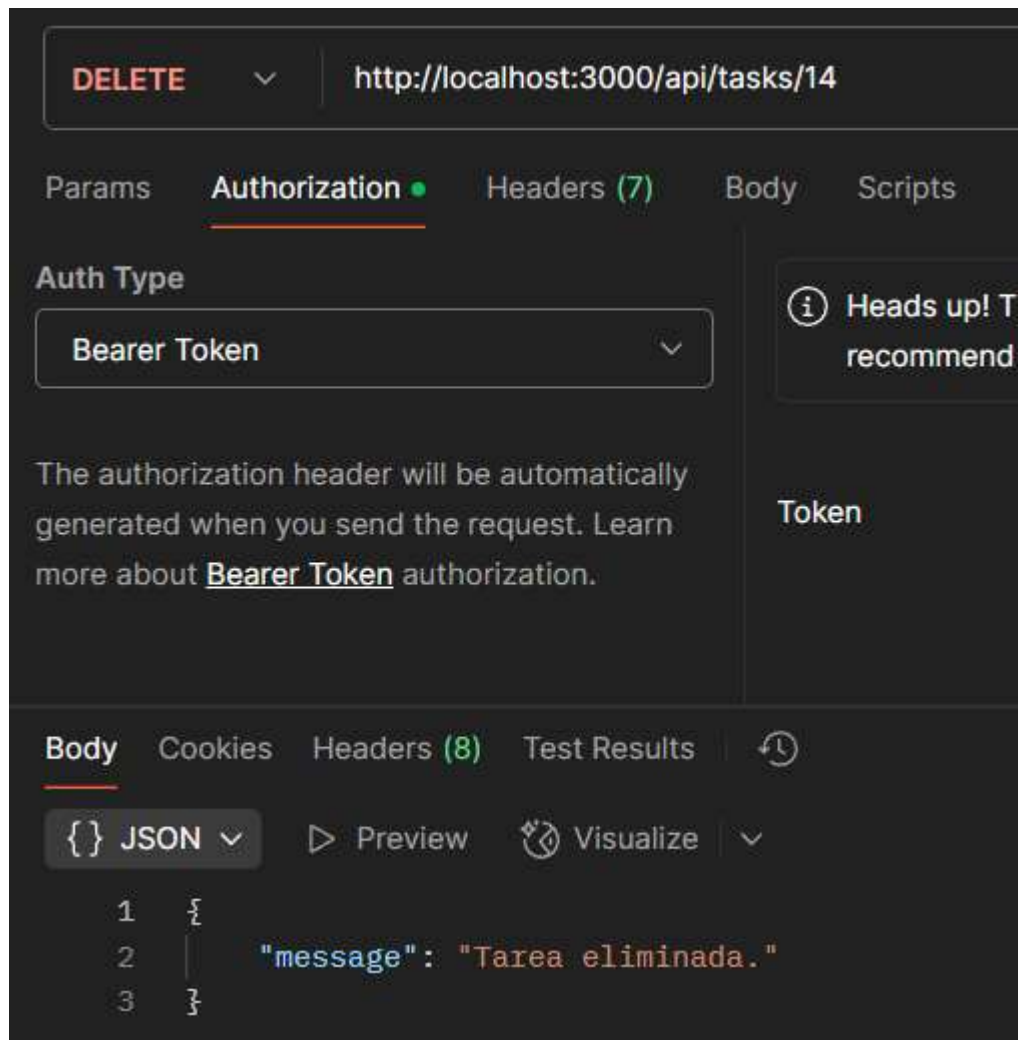
Body (Request):

```
{
  "title": "titulo modificado",
  "description": "Descripcion modifcida",
  "status": "pendiente",
  "dueDate": "2025-04-01"
}
```

Body (Response):

```
{
  "message": "Tarea actualizada exitosamente.",
  "task": {
    "id": 5,
    "title": "titulo modificado",
    "description": "Descripcion modifcida",
    "status": "pendiente",
    "dueDate": "2025-04-01T00:00:00.000Z",
    "userId": 4
  }
}
```

Borrar tarea



DELETE ⌵ `http://localhost:3000/api/tasks/14`

Params **Authorization** ● Headers (7) Body Scripts

Auth Type

Bearer Token ⌵

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

ⓘ Heads up! T recommend

Token

Body Cookies Headers (8) Test Results 🔄

{ } JSON ⌵ ▶ Preview 🔍 Visualize ⌵

```
1 {
2   "message": "Tarea eliminada."
3 }
```