Este documento brinda información detallada sobre la implementación de tokens JWT en un servidor Express y el desarrollo de un módulo de usuarios para expandir una API existente.

JWT en EXPRESS

Luis Pascal Barros Suarez

Introducción:

Implementación de un sistema de autenticación con nombre de usuario y contraseña para restringir el acceso a ciertas funciones de la API.

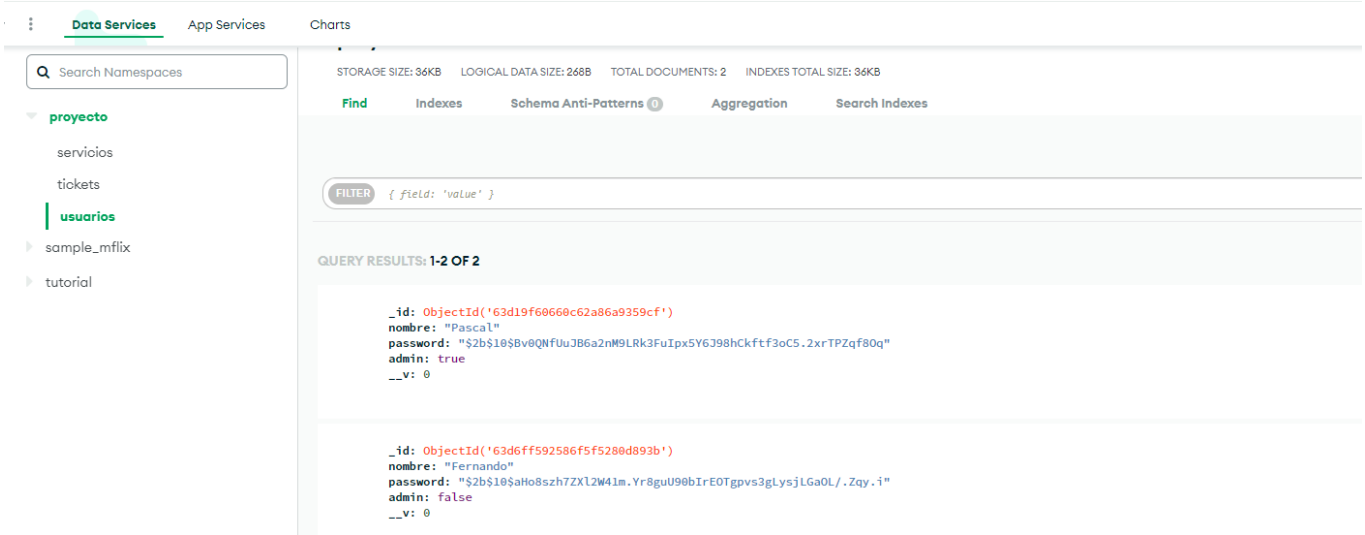
Existen dos rutas protegidas: una requiere iniciar sesión y la otra, además de iniciar sesión, requiere tener permisos de administrador.

Datos de usuario:

- **nombre de usuario**
- **contraseña**
- **Administrador(Bolean)**

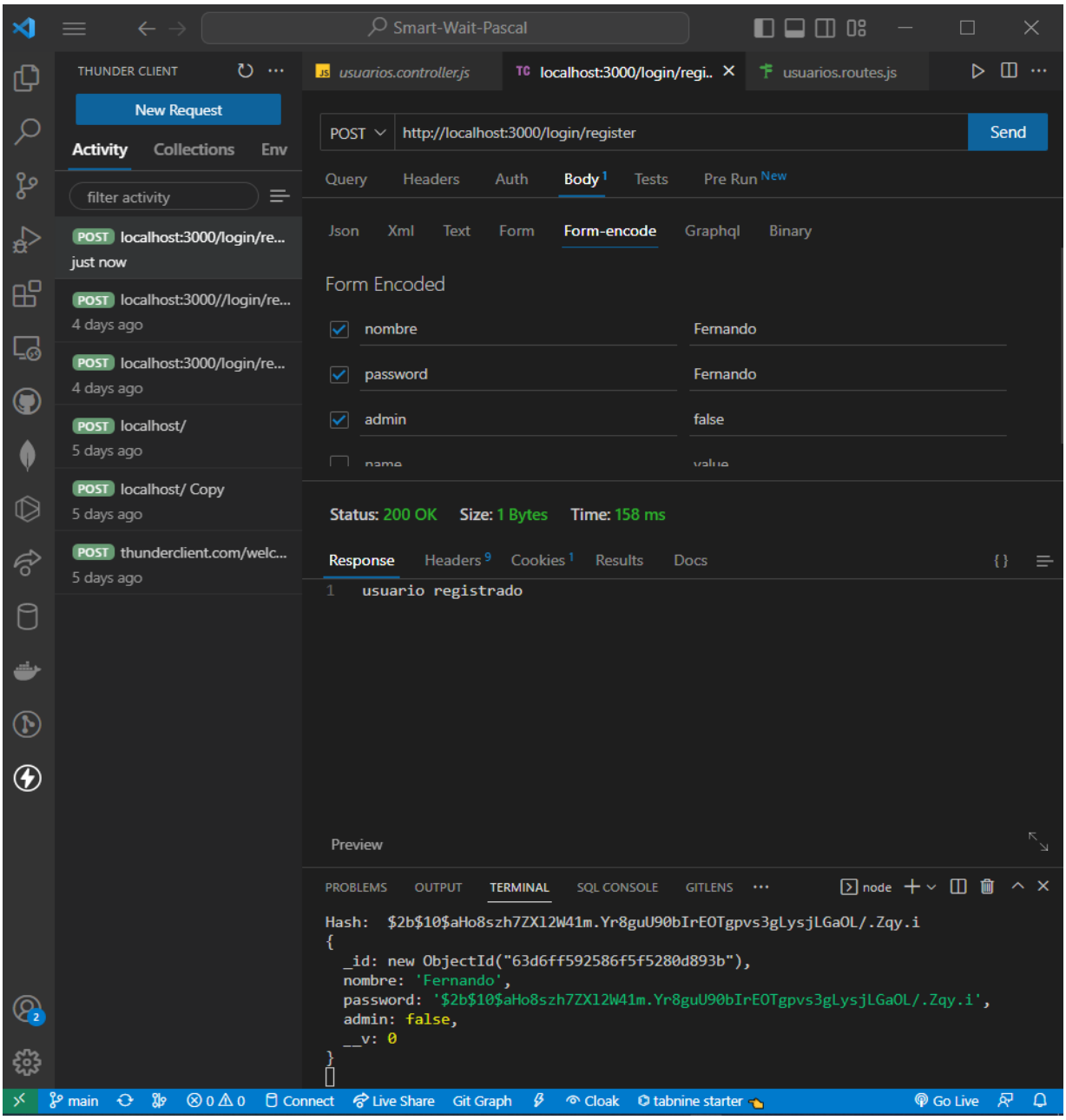
Almacenamiento de los usuarios:

- Utilizamos MongoDB para almacenar información de los usuarios, incluyendo contraseñas cifradas con la librería Bcrypt.



Operaciones

- **Se registran el la aplicación:**



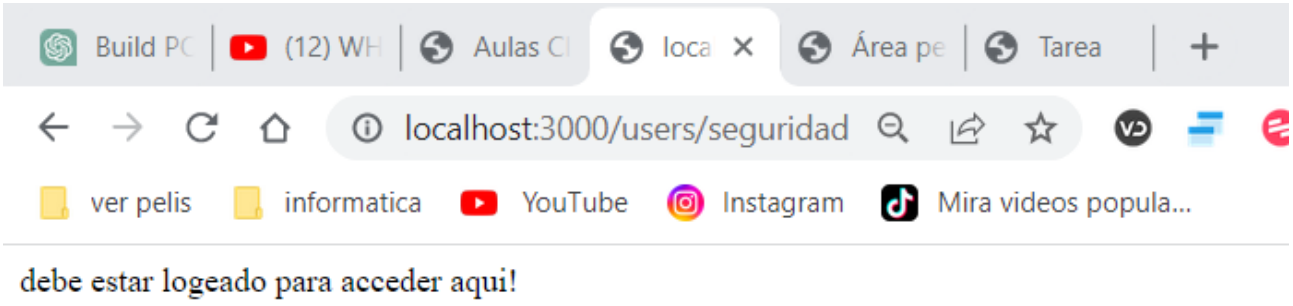
- **Login** (inicia una sesión si los datos de acceso son correctos)

```
Server is running on port 3000.
Server is connected to the database!
Ha entrado un cliente
[
  {
    _id: new ObjectId("63d19f60660c62a86a9359cf"),
    nombre: 'Pascal',
    password: '$2b$10$Bv0QNfUuJB6a2nM9LRk3FuIpx5Y6J98hCkftf3oC5.2xrTPZqf8Oq',
    admin: true,
    __v: 0
  }
]
Passwords match
```

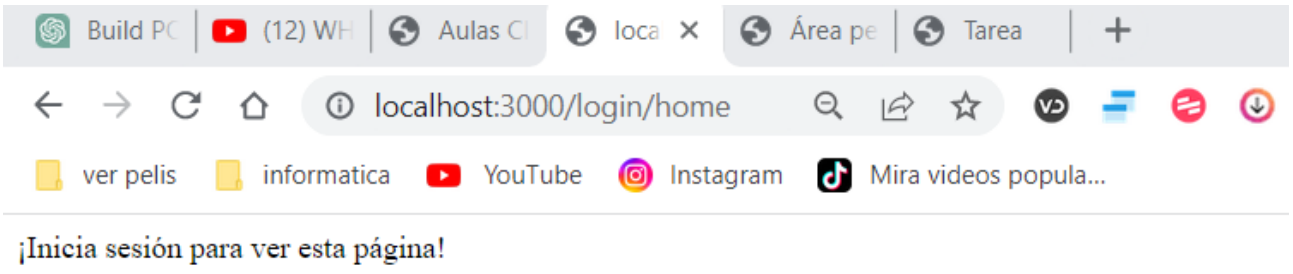
- **Logout** (elimina la información de sesión)

1. **Acceso denegado** a una operación ‘privada’ que necesite autenticación.

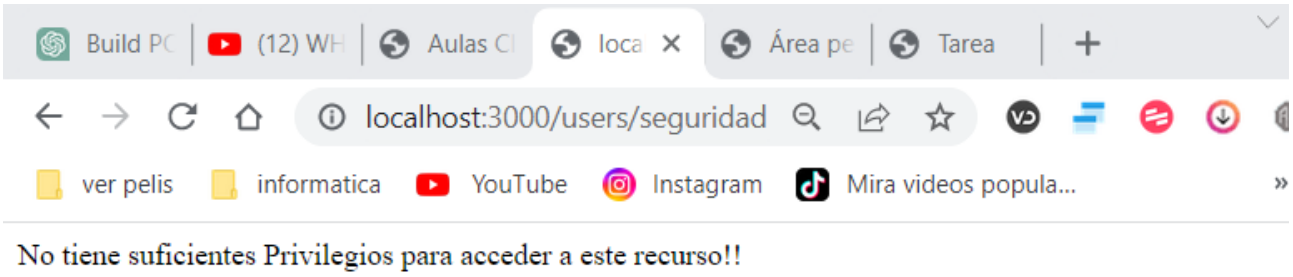
   Para acceder a esta operación "privada" que seria users/seguridad que devuelve todos los usuarios administradores, es necesario estar autenticado y tener permisos de administrador:



   Para acceder a esta operación "privada" que seria login/home es necesario estar autenticado:

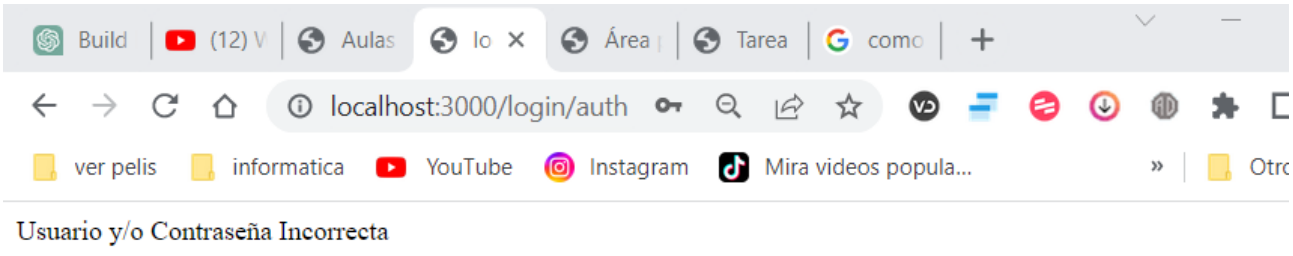


   Aquí se muestra que un usuario que no es administrador no puede acceder al recurso users/seguridad:



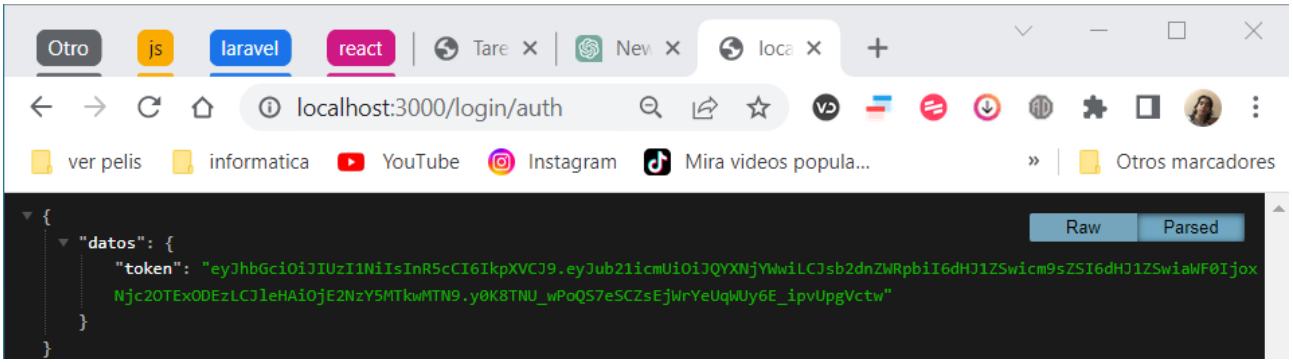
2. Intento de **login fallido**:

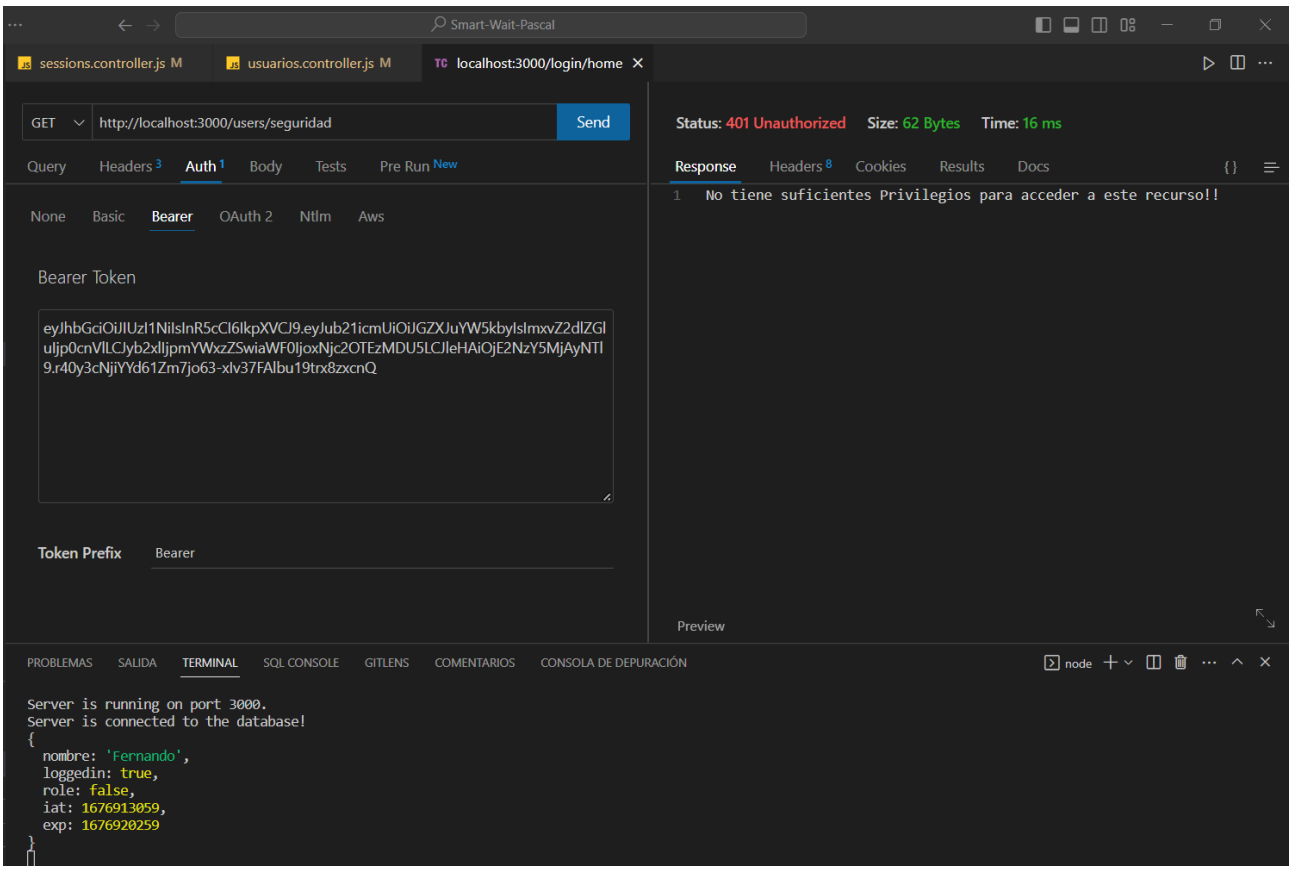
   Aquí intentamos conectarnos con un usuario o contraseña que no es correcta:



3. **Login correcto:**

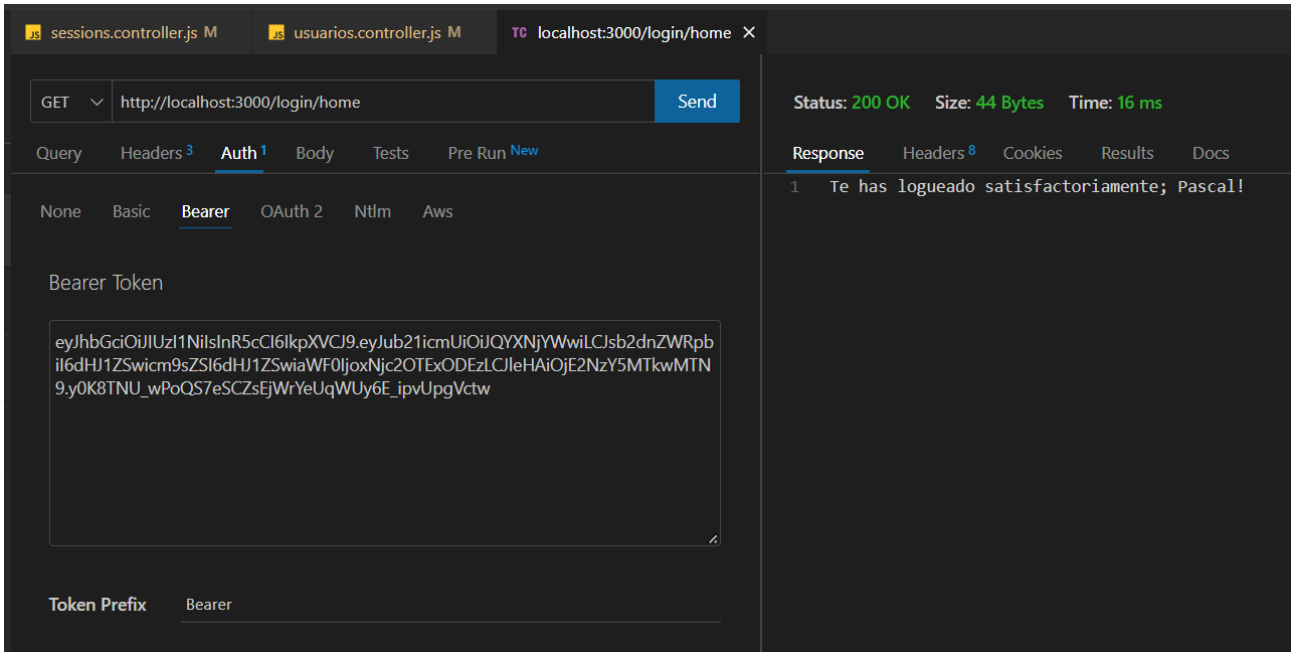
   Aquí nos hemos conectado con el usuario Fernando y ya tenemos acceso al recurso login/home pero no a users/seguridad ya que no es admin:

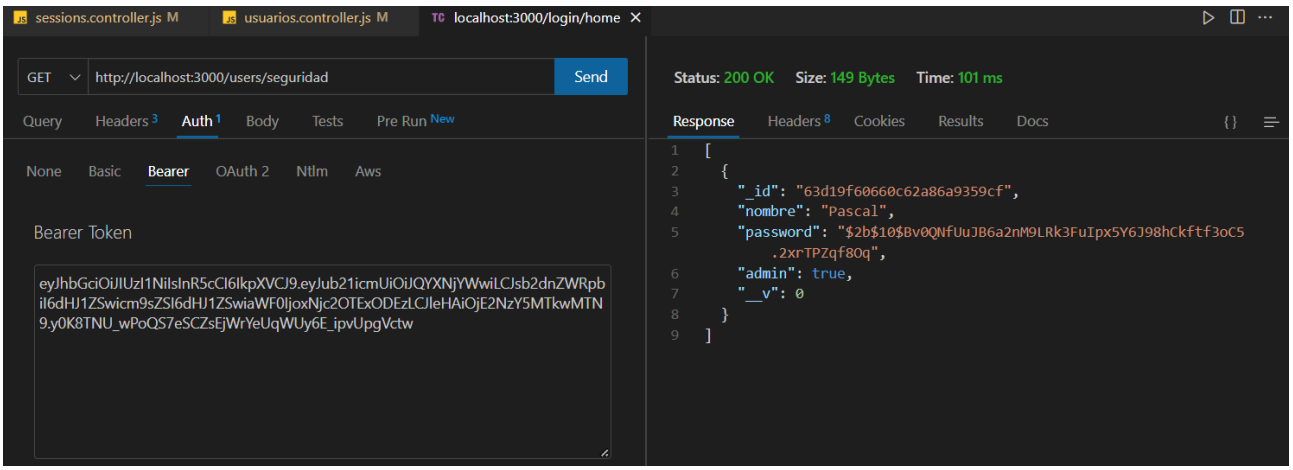




4. **Acceso permitido a la misma operación** una vez autentificado con éxito (siendo administrador).

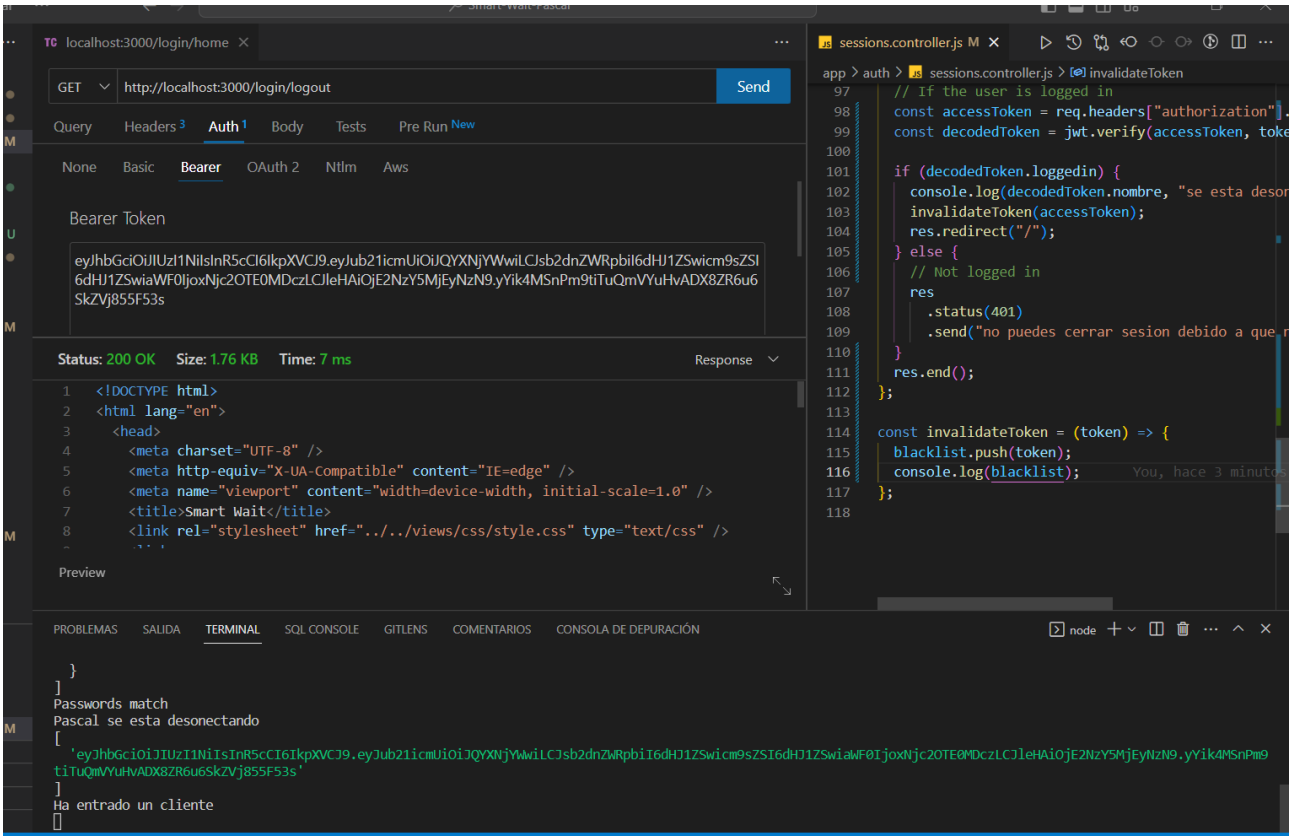
   Aquí con el usuario Pascal que es administrador podremos acceder al recurso users/seguridad:



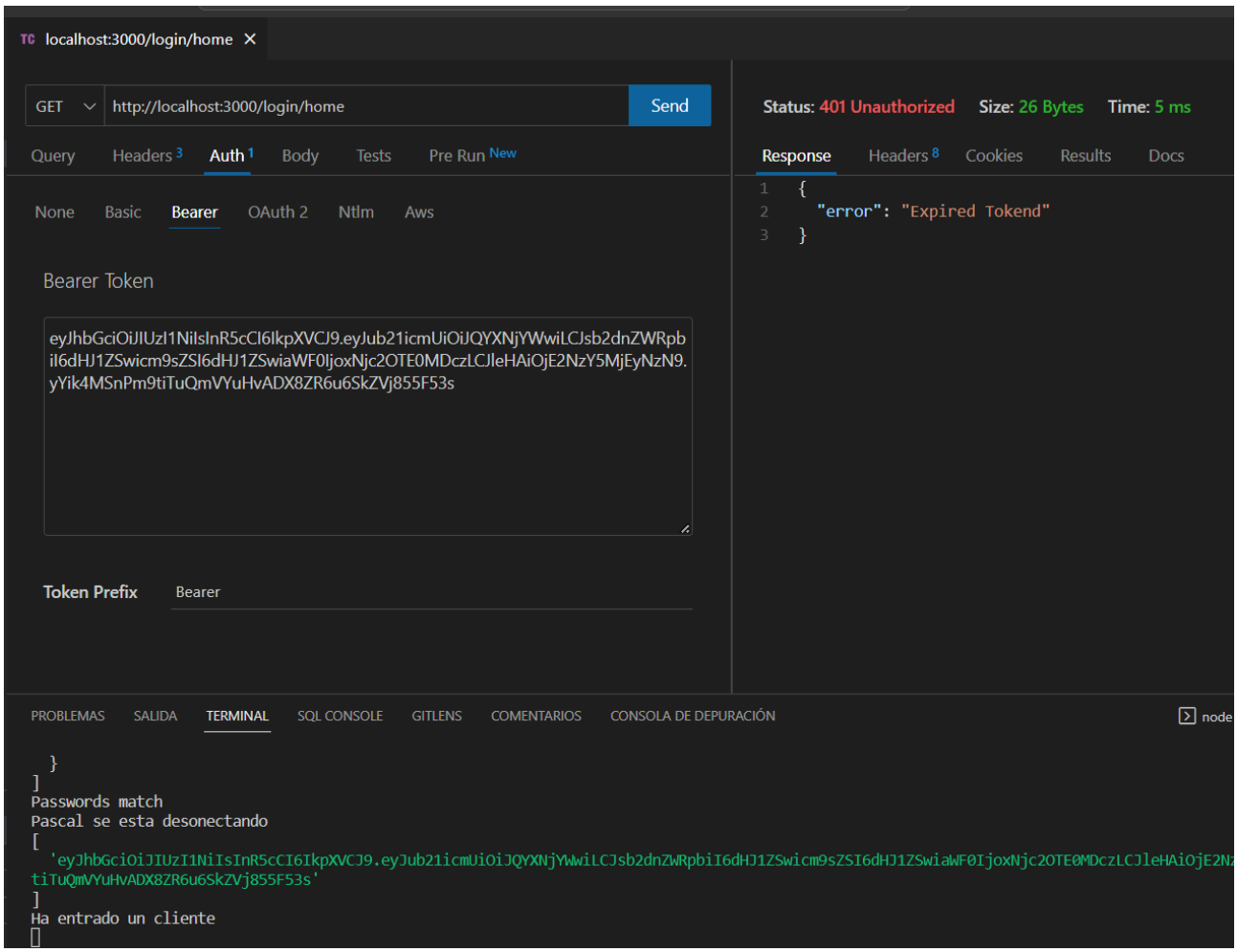

sessions.controller.js M
usuarios.controller.js M
localhost:3000/login/home
GET
http://localhost:3000/users/seguridad
Send
Query
Headers 3
Auth 1
Body
Tests
Pre Run New
None
Basic
Bearer
OAuth 2
Ntlm
Aws
Bearer Token
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJub21icmUiOiJQYXNjYWwiLCJsb2dnZWRpbiI6dHJ1ZSwicm9sZSI6dHJ1ZSwiaWF0IjoxNjc2OTExODEzLCJleHAiOjE2NzY5MTkwMTN9.y0K8TNU_wPoQS7eSCZsEjWrYeUqWUy6E_ipvUpgVctw
Status: 200 OK
Size: 149 Bytes
Time: 101 ms
Response
Headers 8
Cookies
Results
Docs
{ }
[
{
"_id": "63d19f60660c62a86a9359cf",
"nombre": "Pascal",
"password": "$2b$10$Bv0QNfUuJB6a2nM9LRk3FuIpx5Y6J98hCkftf3oC5.2xrTPZqf8Oq",
"admin": true,
"__v": 0
}
]

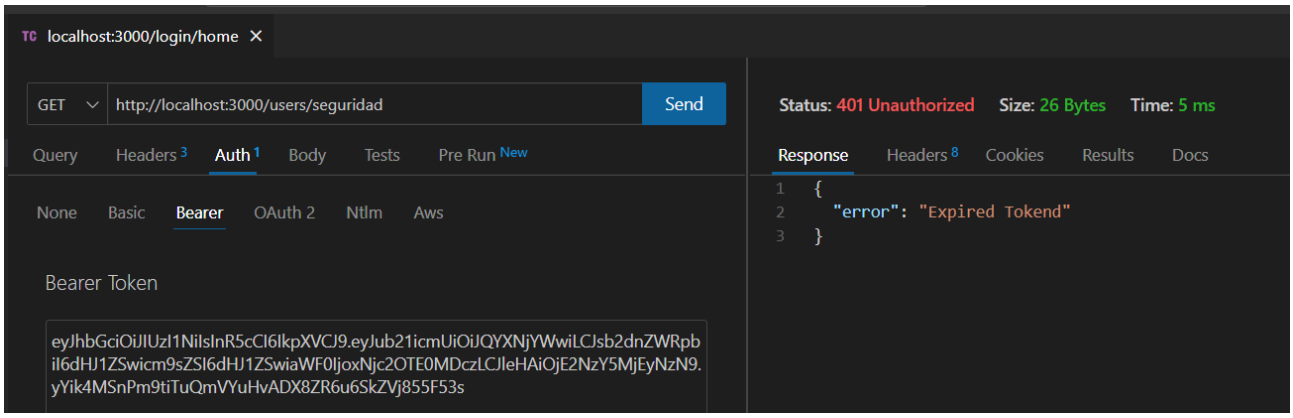
5. **Logout**.

   aqui nos desconectamos del sevidor e intentamos accer la misma operación de antes acceder al recurso login/home



6. **Prueba de acceso a la misma operación ‘privada’** después de hacer logout. Debería dar un mensaje de acceso denegado o algo así.


localhost:3000/login/home
GET
http://localhost:3000/login/home
Send
Status: 401 Unauthorized
Size: 26 Bytes
Time: 5 ms
Query
Headers 3
Auth 1
Body
Tests
Pre Run New
Response
Headers 8
Cookies
Results
Docs
None
Basic
Bearer
OAuth 2
Ntlm
Aws
{
"error": "Expired Tokend"
}
Bearer Token
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJub21icmUiOiJQYXNjYWwiLCJsb2dnZWRpbiI6dHJ1ZSwicm9sZSI6dHJ1ZSwiaWF0IjoxNjc2OTE0MDczLCJleHAiOjE2NzY5MjEyNzN9.yYik4MSnPm9tiTuQmVYuHvADX8ZR6u6SkZVj855F53s
Token Prefix
Bearer
PROBLEMAS
SALIDA
TERMINAL
SQL CONSOLE
GITLENS
COMENTARIOS
CONSOLA DE DEPURACIÓN
node
}
]
Passwords match
Pascal se esta desonectando
[
'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJub21icmUiOiJQYXNjYWwiLCJsb2dnZWRpbiI6dHJ1ZSwicm9sZSI6dHJ1ZSwiaWF0IjoxNjc2OTE0MDczLCJleHAiOjE2NzY5MjEyNzN9.yYik4MSnPm9tiTuQmVYuHvADX8ZR6u6SkZVj855F53s'
]
Ha entrado un cliente


localhost:3000/login/home
GET
http://localhost:3000/users/seguridad
Send
Status: 401 Unauthorized
Size: 26 Bytes
Time: 5 ms
Query
Headers 3
Auth 1
Body
Tests
Pre Run New
Response
Headers 8
Cookies
Results
Docs
None
Basic
Bearer
OAuth 2
Ntlm
Aws
{
"error": "Expired Tokend"
}
Bearer Token
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJub21icmUiOiJQYXNjYWwiLCJsb2dnZWRpbiI6dHJ1ZSwicm9sZSI6dHJ1ZSwiaWF0IjoxNjc2OTE0MDczLCJleHAiOjE2NzY5MjEyNzN9.yYik4MSnPm9tiTuQmVYuHvADX8ZR6u6SkZVj855F53s