

## Evidencias de Pruebas:

Iniciamos sesión con un usuario para ver que se genera el Token

The screenshot shows a Postman request to `localhost:8080/login`. The request method is `POST`, and the body contains the following JSON:

```
1 {  
2   "username": "kevin",  
3   "password": "12345"  
4 }
```

The response status is `200 OK` with a response time of `654 ms` and a size of `894 B`. The response body is:

```
1 {  
2   "message": "Hola kevin has iniciado sesion con exito",  
3   "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJrZXZpbIIsImF1dGhvcmI0aWzIjoiW3tcImF1dGhvcmI0eVwiOlwiIk9MRV9VU0VSXCJ9XSiInVzZXJuWm11jjoia2V2aw4iLCJleHaipjE3NjIx0DcxOTAsImlhC16MTc2MjE4MzU5MH0.VcoUnkgqAage3dQRllIoDLvrsdqfhfdwbPypSBkr9G1",  
4   "username": "kevin"  
5 }
```

Verificamos el Token:

The screenshot shows the `JWT Debugger` tool. The token `eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJrZXZpbIIsImF1dGhvcmI0aWzIjoiW3tcImF1dGhvcmI0eVwiOlwiIk9MRV9VU0VSXCJ9XSiInVzZXJuWm11jjoia2V2aw4iLCJleHaipjE3NjIx0DcxOTAsImlhC16MTc2MjE4MzU5MH0.VcoUnkgqAage3dQRllIoDLvrsdqfhfdwbPypSBkr9G1` is pasted into the input field. The `JSON` tab shows the decoded payload:

```
{  
  "alg": "HS256"  
}
```

The `DECODED PAYLOAD` tab shows the expanded JSON payload:

```
{  
  "sub": "kevin",  
  "authorities": "[{"authority": "ROLE_USER"}]",  
  "username": "kevin",  
  "exp": 1762187190,  
  "iat": 1762183590  
}
```

Se intenta crear una tarea con un usuario NO admin

The screenshot shows the Postman interface. At the top, it says "Home Workspaces API Network". Below that, there are two tabs: "POST localhost:8080/login" and "POST localhost:8080/api/tasks". The second tab is selected. The URL is "localhost:8080/api/tasks". On the right, there are "Save", "Share", and "Send" buttons. The "Send" button is highlighted in blue. Below the URL, the method is "POST" and the endpoint is "localhost:8080/api/tasks". There are tabs for "Params", "Authorization", "Headers (10)", "Body", "Scripts", "Tests", and "Settings". The "Body" tab is selected and has a sub-tab "raw" which is also selected. The JSON body is:

```
1 {
2   "name": "Tarea con Seguridad",
3   "description": "Otra Tarea"
4 }
```

At the bottom, the status is "403 Forbidden" with a response time of "52 ms" and a size of "300 B". There are buttons for "Raw", "Preview", "Debug with AI", and "Cancel" (which is red).

## Se crea una Tarea

This screenshot shows the Postman UI for a POST request to "/api/tasks". The "Parameters" section is empty. The "Request body" is required and set to "application/json". The JSON body is:

```
{
  "name": "Otra Tarea",
  "description": "Descripcion Tarea"
}
```

## Vemos que se creó correctamente

This screenshot shows the "Responses" section of the Postman UI. It includes a "Curl" command, a "Request URL" (http://localhost:8080/api/tasks), and a "Server response". The "Code" tab is selected, showing a status of "201 Undocumented". The "Details" tab is also visible. The "Response body" is:

```
{
  "id": 3,
  "name": "Otra Tarea",
  "description": "Descripcion Tarea"
}
```

There are "Copy" and "Download" buttons next to the response body. The "Response headers" section shows:

```
connection: keep-alive
content-type: application/json
date: Sat, 01 Nov 2025 22:31:01 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

## Listamos las Tareas

Curl

```
curl -X 'GET' \
  'http://localhost:8080/api/tasks' \
  -H "accept: */*
```

Request URL

```
http://localhost:8080/api/tasks
```

Server response

**Code** Details

200 Response body

```
[
  {
    "id": 1,
    "name": "Probar",
    "description": "Probar buen funcionamiento"
  },
  {
    "id": 2,
    "name": "Programar",
    "description": "Programar sitio Web"
  },
  {
    "id": 3,
    "name": "Otra Tarea",
    "description": "Descripción Tarea"
  }
]
```

Response headers

```
connection: keep-alive
content-type: application/json
date: Sat, 01 Nov 2025 22:31:13 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

 

Actualizamos la Tarea con el numero 3 ya que el nombre es erroneo

Curl

```
curl -X 'PUT' \
  'http://localhost:8080/api/tasks/3' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "name": "Diseñar",
    "description": "Diseñar Pagina Web"
}'
```

Request URL

```
http://localhost:8080/api/tasks/3
```

Server response

**Code** Details

201 Undocumented Response body

```
{
  "id": 3,
  "name": "Diseñar",
  "description": "Diseñar Pagina Web"
}
```

Response headers

```
connection: keep-alive
content-type: application/json
date: Sat, 01 Nov 2025 22:37:41 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

 

Volvemos a Listar y verificamos que se halla actualizado la tarea 3

Request URL

```
http://localhost:8080/api/tasks
```

Server response

**Code** Details

200 Response body

```
[
  {
    "id": 1,
    "name": "Probar",
    "description": "Probar buen funcionamiento"
  },
  {
    "id": 2,
    "name": "Programar",
    "description": "Programar sitio Web"
  },
  {
    "id": 3,
    "name": "Diseñar",
    "description": "Diseñar Pagina Web"
  }
]
```

Response headers

```
connection: keep-alive
content-type: application/json
date: Sat, 01 Nov 2025 22:38:57 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

 

Se crea un usuario con contraseña ya encriptada

POST localhost:8080/api/users

Params Authorization Headers (9) Body Scripts Tests Settings Cookies

Body raw binary GraphQL JSON Beautify

```
1 {
2   "username": "kevin",
3   "password": "12345",
4   "enable": true,
5   "admin": false
6 }
```

Body Cookies (1) Headers (11) Test Results 201 Created 773 ms 506 B

{ } JSON Preview Visualize

```
1 {
2   "id": 1,
3   "username": "kevin",
4   "password": "$2a$10$HY0faRP60XwMdjISRSK9GeF/vs1Hb2NcmFNgs2X1v.FZAJ.G84o6S",
5   "roles": [
6     {
7       "id": 2,
8       "name": "ROLE_USER"
9     }
10 ],
11 "enable": true,
12 "admin": false
13 }
```

Se valida que no existan usuarios repetidos

GET localhost:8080/api/users • POST localhost:8080/api/users • + No environment

localhost:8080/api/users/register

POST localhost:8080/api/users/register

Params Authorization Headers (9) Body Scripts Tests Settings Cookies

Body raw binary GraphQL JSON Beautify

```
1 {
2   "username": "kevin",
3   "password": "12345",
4   "enable": true,
5   "admin": false
6 }
```

Body Cookies (1) Headers (10) Test Results 400 Bad Request 242 ms 376 B

{ } JSON Preview Debug with AI

```
1 {
2   "username": "El campo username Ya existe en la base de datos"
3 }
```

Se valida que se logue con un usuario existente

POST localhost:8080/login • +

HTTP localhost:8080/login No environment

POST localhost:8080/login Send

Params Authorization Headers (9) **Body** Scripts Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "username": "kevin",  
3   "password": "12345"  
4 }
```

Body Cookies (1) Headers (12) Test Results 200 OK 613 ms 894 B

{ } JSON Preview Visualize

```
1 {  
2   "message": "Hola kevin has iniciado seccion con exito",  
3   "token": "eyJhbGciOiJIUzI1NiJ9.  
4 iLCJleHAiOjE3NjIxMzA2NDEsImhdCi6MTc2MjEyNzA0MX0.XL197oZtvNtJIDJLVDrQelyn0tWVH4LaUbdZQwNi6dw",  
5   "username": "kevin"  
6 }
```