

Evidencias de Pruebas:

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

The screenshot shows a POST request to `localhost:8080/login`. The request 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.eJzdwIiOjIjZXZpbIIsImF1dGhvcml0aWzIjoiW3tcImF1dGh
4   cml0eVwiOlwiUk9MRV9VU0VSXCJ9XSiInVzZXJuYW1lIjoia2V2aW4iLCleHaiOjE3NjIx0
DcxOTAsImIhdCI6MTc2MjE4MzU5MH0.VcoUnkgqAage3dQrlIioDLvrsdqfhfdwbPypSBkr9G
5   "username": "kevin"
```

Verificamos el Token:

The JWT Debugger tool shows a valid JWT token. The header (JSON) is:

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

The payload (JSON) is:

```
{
  "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. In the top navigation bar, 'API Network' is selected. Below it, a search bar and several icons are visible. The main workspace shows a 'POST localhost:8080/api/tasks' request. The 'Body' tab is selected, showing raw JSON input:

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

Below the body, the response status is '403 Forbidden'. The response details show a 52 ms duration and 300 B size. The response body is empty.

Verificamos la autenticidad con Swagger y vemos que se genere el Token

The screenshot shows the Swagger UI. A curl command is displayed at the top:

```
curl -X 'POST' \
  'http://localhost:8080/login' \
  -H 'Accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "username": "juanco",
    "password": "12345"
}'
```

Below it, the 'Request URL' is set to 'http://localhost:8080/login'. The 'Server response' section shows a 200 status with a JSON response body containing a message and a token:

```
{
  "message": "Hola Juanco has iniciado sesion con exito",
  "token": "eyJhbGciOiJIUzI1NiJ9.eJzdkIi0iJKdWFuY28iLCJhdXRob3JpdGlycyI6Ilt7KCJhdXRob3JpdHlcIjpcI1JPTEVfQURNSU5cIn0se1wiYXV0aG9yaXR5XC16XC3ST0xFX1VTRVJcIn1dIividXN1cm5hbWUiOjIkdFUyZ8ilC1eHAiOjE3NTIxOTUzMzs1MzIhdC16MTc2MjE5MTczOX0.0CdG19puKZfwuIvIM4ANzOR2DwCT-hz3qsyIPsQR0es",
  "username": "Juanco"
}
```

The 'Response headers' section shows a detailed list of HTTP headers, including 'authorization: Bearer' with the generated token.

Y nos autorizamos

The screenshot shows the Swagger UI with the 'Servers' dropdown set to 'http://localhost:8080 - Generated server url'. A green 'Authorize' button with a lock icon is visible.

Después debe salir así

The screenshot shows the Swagger UI with the 'Servers' dropdown set to 'http://localhost:8080 - Generated server url'. The 'Authorize' button is now greyed out and inactive.

Se crea una Tarea

A screenshot of a POST request to the endpoint `/api/tasks`. The request body contains the following JSON:

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

Vemos que se creó correctamente

A screenshot of a successful POST response. The curl command used was:

```
curl -X 'POST' \
  'http://localhost:8080/api/tasks' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "name": "Otra Tarea",
    "description": "Descripcion Tarea"
}'
```

The request URL is `http://localhost:8080/api/tasks`. The server response code is 201 (Undocumented) with the following details:

Response body:

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

Response headers:

```
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

A screenshot of a GET request to the endpoint `/api/tasks`. The response code is 200 (OK) and the response body is:

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

The response headers are:

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

Buscamos Tarea Por ID

Name Description

id * required integer(\$int64) 1 (path)

Execute Clear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:8080/api/tasks/1' \
-H 'accept: */*'
```

Request URL

```
http://localhost:8080/api/tasks/1
```

Server response

Code Details

200 Response body

```
{
  "id": 1,
  "name": "Probar",
  "description": "Probar buen funcionamiento"
}
```

Download

Verificamos que no se creen Tareas con nombre vacío

Responses

Curl

```
curl -X 'POST' \
'http://localhost:8080/api/tasks' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
  "name": "",
  "description": "Descripcion"
}'
```

Request URL

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

Server response

Code Details

400 Error: response status is 400 Undocumented

Response body

```
{
  "name": "El campo name el tamaño debe estar entre 3 y 25"
}
```

Download

Response headers

```
connection: close
content-type: application/json
date: Mon, 03 Nov 2025 17:36:47 GMT
transfer-encoding: chunked
```

Vemos que se eliminan por ID las tareas

Name Description

id * required
integer(\$int64)
(path)

4

Execute Clear

Responses

Curl

```
curl -X 'DELETE' \
'http://localhost:8080/api/tasks/4' \
-H 'accept: */*'
```

Request URL

http://localhost:8080/api/tasks/4

Server response

Code Details

200 Response body

```
{
  "id": 4,
  "name": "Tarea 4",
  "description": "Descripcion Tarea 4"
}
```

Download

Actualizamos la Tarea con el número 3 ya que el nombre es erróneo

Curl

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

Download

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 haya 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"
  }
]
```

Download

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 desde Postman

POST localhost:8080/api/users

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

Body

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

201 Created 773 ms 506 B

Body Cookies (1) Headers (11) Test Results

{ } 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 Postman

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 Beautify

Body

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

400 Bad Request 242 ms 376 B

Body Cookies (1) Headers (10) Test Results

{ } 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 Postman

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 }
```