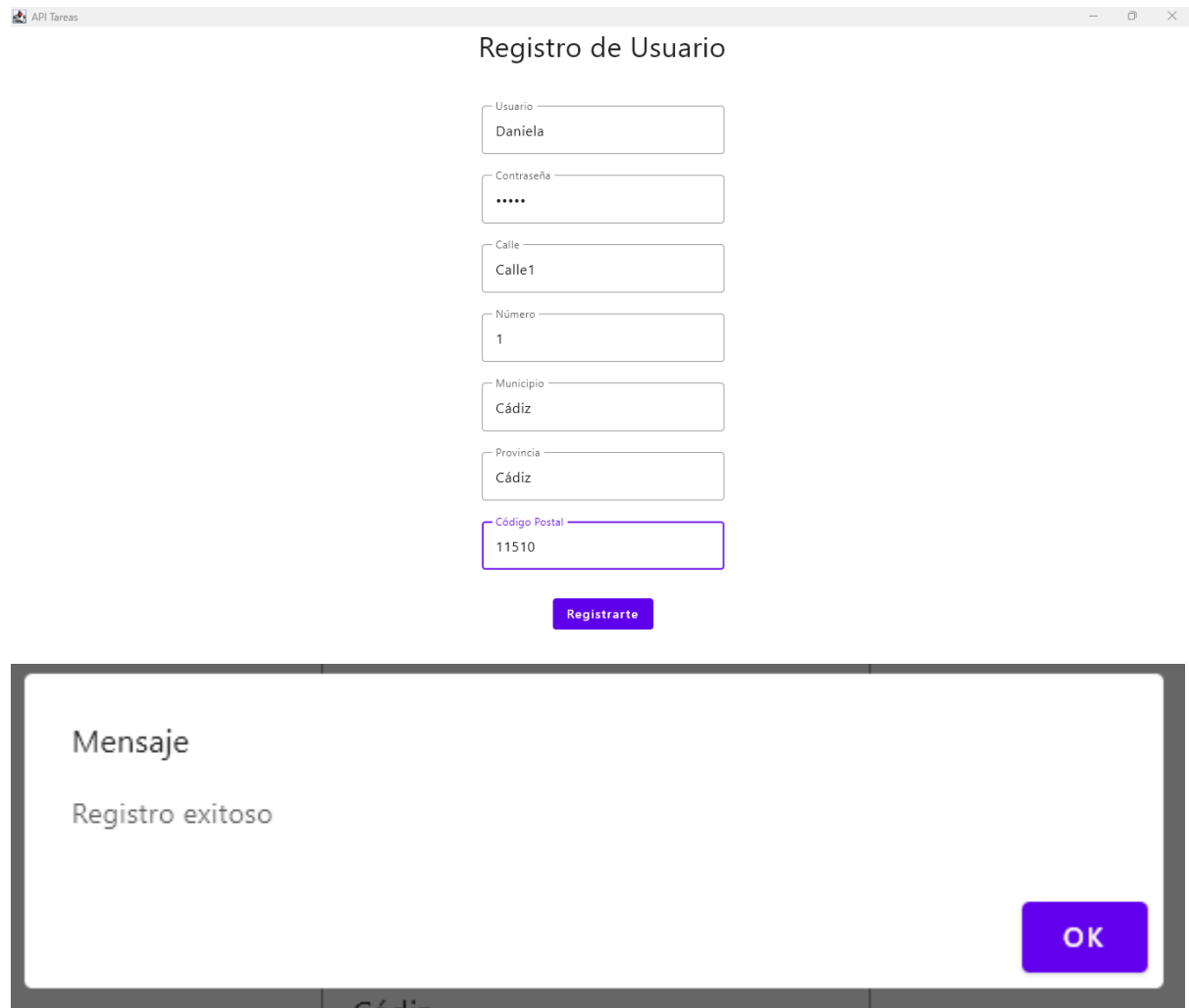


PRUEBAS GESTIÓN USUARIOS

Registro

Un ejemplo de que el registro funciona:



The screenshot shows a web browser window titled "API Tareas". The main content is a registration form titled "Registro de Usuario". The form contains several input fields: "Usuario" (filled with "Daniela"), "Contraseña" (filled with "*****"), "Calle" (filled with "Calle1"), "Número" (filled with "1"), "Municipio" (filled with "Cádiz"), "Provincia" (filled with "Cádiz"), and "Código Postal" (filled with "11510"). Below the form is a blue button labeled "Registrarte". Below the form, there is a modal dialog box with the title "Mensaje" and the text "Registro exitoso". In the bottom right corner of the modal is a blue button labeled "OK".

Registro de Usuario

Usuario
Daniela

Contraseña

Calle
Calle1

Número
1

Municipio
Cádiz

Provincia
Cádiz

Código Postal
11510

Registrarte

Mensaje

Registro exitoso

OK

```
_id: ObjectId('67b86d8f4de1851ddb1286db')
username : "Daniela"
password : "$2a$10$VwDKRJwRnLPV0UtMmxVvA.U49BP5mrWpSuBp6xNmabbJzSRPRJwnZa"
roles : "USER"
▸ direccion : Object
  _class : "com.es.aplicacion.model.Usuario"
```

Ejemplo de que comunica cuando no funciona y te avisa por qué:

API Tareas

— □ ×

Registro de Usuario

Usuario

Contraseña

Calle

Número

Municipio

Provincia

Código Postal

Registrarte

Mensaje

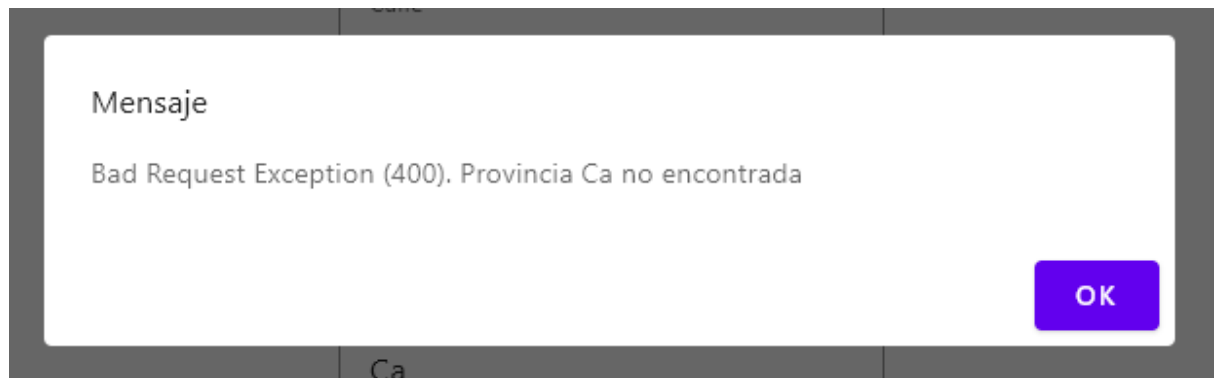
Bad Request Exception (400). Uno o más campos vacíos

OK

Mensaje

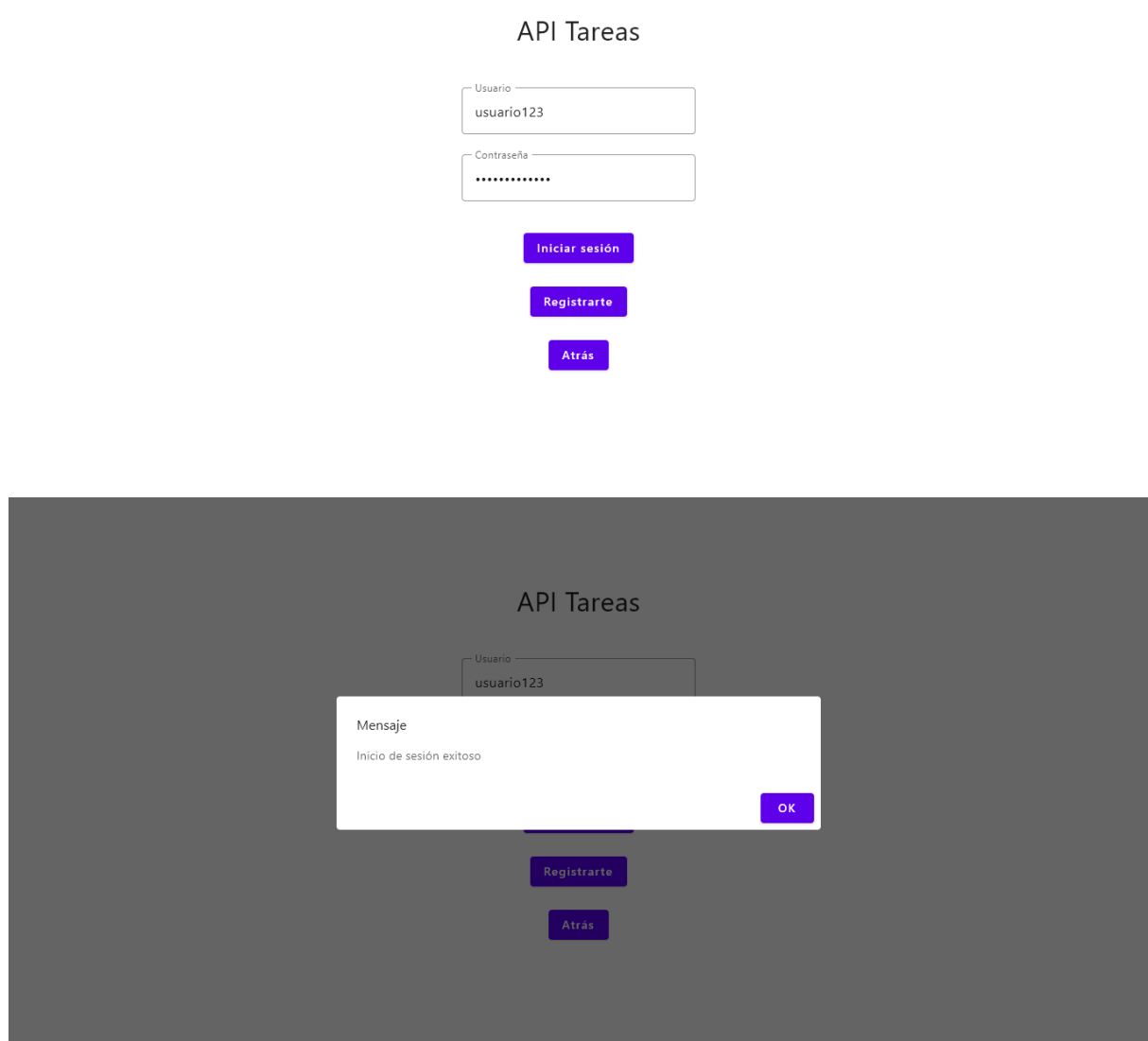
Usuario Daniela ya está registrado

OK

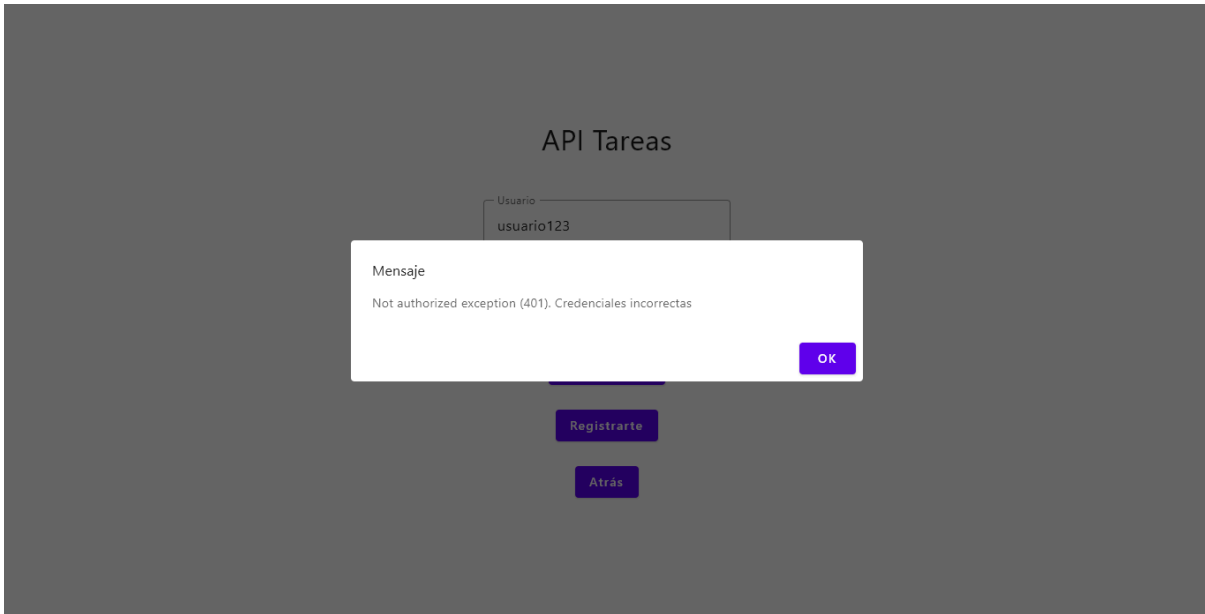


Login

Un ejemplo de que el login funciona:



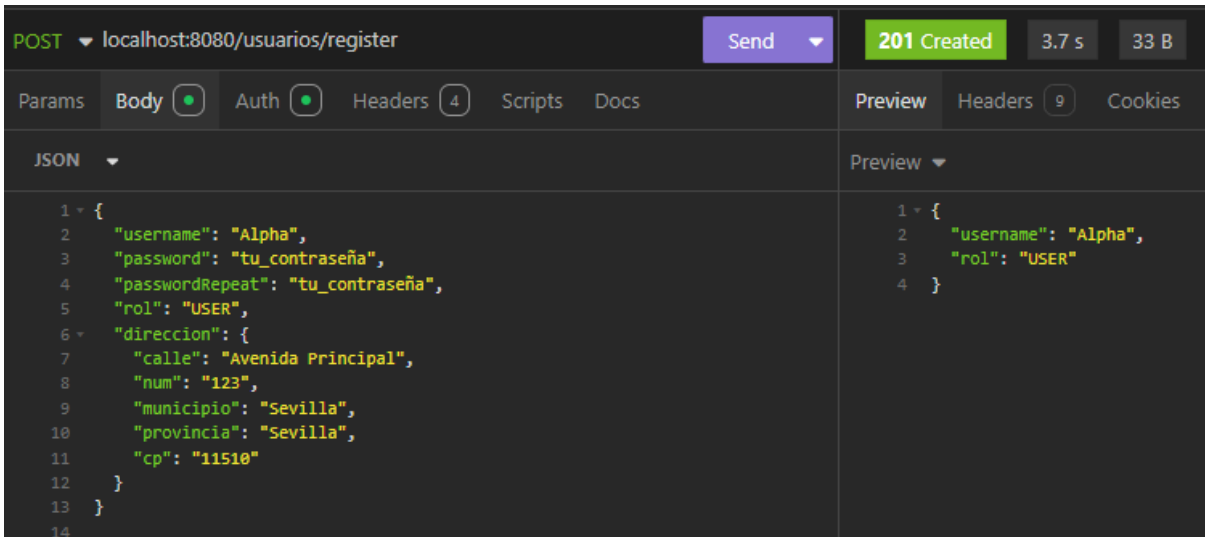
Ejemplo de que comunica cuando no funciona y te avisa por qué:



Insomniac

Capturas del funcionamiento

Register



Si ya existe ese usuario

POST localhost:8080/usuarios/register

Send

500 Internal Server Error

271 s

75 B

Params Body Auth Headers 4 Scripts Docs

Preview Headers 10 Cookies Tests 0/0 → Mock

JSON

```
1 {
2   "username": "Daniela",
3   "password": "tu_contraseña",
4   "passwordRepeat": "tu_contraseña",
5   "rol": "USER",
6   "direccion": {
7     "calle": "Avenida Principal",
8     "num": "123",
9     "municipio": "Cádiz",
10    "provincia": "Cádiz",
11    "cp": "11510"
12  }
13 }
14
```

Preview

```
1 {
2   "message": "Usuario Daniel ya está registrado",
3   "uri": "/usuarios/register"
4 }
```

Si hay algún campo vacío

The screenshot shows the Swagger UI interface. At the top, the method is **POST** and the URL is `localhost:8080/usuarios/register`. The status bar indicates a **400 Bad Request** with a response time of **29 ms** and a body size of **95 B**. The **Body** tab is selected, showing the request JSON. The **Preview** tab is also visible, showing the response JSON.

Request Body (JSON)	Response Body (JSON)
<pre> 1 { 2 "username": "", 3 "password": "tu_contraseña", 4 "passwordRepeat": "tu_contraseña", 5 "rol": "USER", 6 "direccion": { 7 "calle": "Avenida Principal", 8 "num": "123", 9 "municipio": "Sevilla", 10 "provincia": "Sevilla", 11 "cp": "11510" 12 } 13 }</pre>	<pre> 1 { 2 "message": "Bad Request Exception (400). Uno o más campos vacíos", 3 "uri": "/usuarios/register" 4 }</pre>

Login

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

- Request:** POST to `localhost:8080/usuarios/login`. The body is a JSON object: `{ "username": "usuario123", "password": "tu_contraseña" }`.
- Response:** 201 Created, 2.95 s, 492 B. The response body is a JSON object: `{ "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpzZW50L3V1ZXQzIiwiaWF0IjE1IjE3PTVfVWVudF0iOiJ1c2wzS1M5Ni5Ga0Q7Yal1MUMoFMe7lRVqc8Zmog_Z-mv0T8vdsB7UzFr2TkSeIHu3rbou-060JlFrG1aMu23Gvj1RtnCR-gFLV4vTH23k8EUA6AY7fC-qQc52vcu8exHt6LPCFwPp9noyG-Qv--IhrdFPrFPFPlgde2R2-ZC_3nQ3G5Fw8ZyM7F90Itald62Vnxq9LKYHMR-B3Y0duaIlgt1Sbbvj1CHIEPoYt_M07Igv1Tv280z39HS-Wau_975q" }`.

si hay datos incorrectos

The screenshot shows a REST client interface. The top bar indicates a POST request to `localhost:8080/usuarios/login` with a status of **401 Unauthorized**, a response time of **979 ms**, and a body size of **94 B**. The request body is a JSON object:

```
{  "username": "usuario123",  "password": "tu_contrasea"}
```

. The response body is a JSON object:

```
{  "message": "Not authorized exception (401). Credenciales incorrectas",  "uri": "/usuarios/login"}
```

Crear Tarea

```
POST localhost:8080/tareas
```

Send 201 Created 1.33 s 153 B

Params Body Auth Headers 4 Scripts Docs

JSON

```
1 {
2   "objetivo": "Preparar presentación de proyecto",
3   "fecha": "2025-03-05T09:00:00.000Z"
4 }
```

Preview

```
1 {
2   "id": "67c1b09d5752f14749e82a5d",
3   "autor": "ab",
4   "objetivo": "Preparar presentación de proyecto",
5   "fecha": "2025-03-05T09:00:00.000+00:00",
6   "completada": false
7 }
```

Si un usuario que no es admin intenta crear la tarea a otra persona:

```
POST localhost:8080/tareas
```

Send 401 Unauthorized 100 ms 118 B Just Now

Params Body Auth Headers 4 Scripts Docs

JSON

```
1 {
2   "autor": "usuario2",
3   "objetivo": "Preparar presentación de proyecto",
4   "fecha": "2025-03-05T09:00:00.000Z"
5 }
```

Preview

```
1 {
2   "message": "Not authorized exception (401). No tienes permisos para
3   crear tareas para otros usuarios",
4   "uri": "/tareas"
5 }
```

Si hay campos vacios

```
POST localhost:8080/tareas
```

Send 400 Bad Request 80 ms 104 B Just Now

Params Body Auth Headers 4 Scripts Docs

JSON

```
1 {
2   "objetivo": "",
3   "fecha": "2025-03-05T09:00:00.000Z"
4 }
```

Preview

```
1 {
2   "message": "Bad Request Exception (400). El objetivo de la tarea no
3   puede estar vacío",
4   "uri": "/tareas"
5 }
```

Obtener tareas

Si eres usuario solo puedes ver las tuyas si eres admin puedes ver todas

```
GET localhost:8080/tareas
```

Send 200 OK 190 ms 155 B

Params Body Auth Headers 3 Scripts Docs

No Body

Preview

```
1 [
2   {
3     "id": "67c1b09d5752f14749e82a5d",
4     "autor": "ab",
5     "objetivo": "Preparar presentación de proyecto",
6     "fecha": "2025-03-05T09:00:00.000+00:00",
7     "completada": false
8   }
9 ]
```

```
val tareas = if (roles.contains("ROLE_ADMIN")) {
    // Administrador puede ver todas las tareas
    tareaRepository.findAll()
} else {
    // Usuario normal solo puede ver sus tareas
    tareaRepository.findByAutor(username)
}
```

Marcar Tarea como Completada

A screenshot of a REST client interface. The top bar shows a PUT request to `localhost:8080/tareas/67c1b09d5752f14749e82a5d/completar` with a status of **200 OK**, a response time of **2.11 s**, and a size of **152 B**. The interface has tabs for Params, Body, Auth, Headers, Scripts, and Docs. The Body tab is selected, showing "No Body". The Preview tab is also selected, displaying a JSON response:

```
1 {
2   "id": "67c1b09d5752f14749e82a5d",
3   "autor": "ab",
4   "objetivo": "Preparar presentación de proyecto",
5   "fecha": "2025-03-05T09:00:00.000+00:00",
6   "completada": true
7 }
```

Si no eres Admin o el Usuario de la tarea

A screenshot of a REST client interface showing a 401 Unauthorized response. The top bar shows a PUT request to `localhost:8080/tareas/67c1b09d5752f14749e82a5d/completar` with a status of **401 Unauthorized**, a response time of **212 ms**, and a size of **141 B**. The interface has tabs for Params, Body, Auth, Headers, Scripts, and Docs. The Body tab is selected, showing "No Body". The Preview tab is also selected, displaying a JSON response:

```
1 {
2   "message": "Not authorized exception (401). No tienes permisos para
3   modificar esta tarea",
4   "uri": "/tareas/67c1b09d5752f14749e82a5d/completar"
5 }
```

Si la tarea no existe

A screenshot of a REST client interface showing a 400 Bad Request response. The top bar shows a PUT request to `localhost:8080/tareas/64a3e4f12b7c1e8a9d3f5b2a/completar` with a status of **400 Bad Request**, a response time of **1.59 s**, and a size of **113 B**. The interface has tabs for Params, Body, Auth, Headers, Scripts, and Docs. The Body tab is selected, showing "No Body". The Preview tab is also selected, displaying a JSON response:

```
1 {
2   "message": "Bad Request Exception (400). Tarea no encontrada",
3   "uri": "/tareas/64a3e4f12b7c1e8a9d3f5b2a/completar"
4 }
```

Desmarcar tarea como completada

Sigue la misma lógica que marcar como completada

A screenshot of a REST client interface. The top bar shows a PUT request to `localhost:8080/tareas/67c1b09d5752f14749e82a5d/descompletar` with a status of **200 OK**, a response time of **998 ms**, and a size of **153 B**. The interface has tabs for Params, Body, Auth, Headers, Scripts, and Docs. The Body tab is selected, showing "No Body". The Preview tab is also selected, displaying a JSON response:

```
1 {
2   "id": "67c1b09d5752f14749e82a5d",
3   "autor": "ab",
4   "objetivo": "Preparar presentación de proyecto",
5   "fecha": "2025-03-05T09:00:00.000+00:00",
6   "completada": false
7 }
```

Eliminar Tarea

DELETE

localhost:8080/tareas/67c1b09d5752f14749e82a5d

Send

204 No Content

505 ms

0 B

Params

Body

Auth

Headers

Scripts

Docs

Preview

Headers

Cookies

Tests

No Body

Preview

No body returned for response

Si no existe la tarea

DELETE

localhost:8080/tareas/67c1b09d5752f14749e82

Send

400 Bad Request

1.88 s

100 B

Params

Body

Auth

Headers

Scripts

Docs

Preview

Headers

Cookies

Tests

Mock

Console

No Body

Preview

```
1 - {
2   "message": "Bad Request Exception (400). Tarea no encontrada",
3   "uri": "/tareas/67c1b09d5752f14749e82"
4 }
```

Si no eres el autor de la tarea o un admin

DELETE

localhost:8080/tareas/67c1b09d5752f14749e82a5d

Send

401 Unauthorized

172 ms

130 B

Params

Body

Auth

Headers

Scripts

Docs

Preview

Headers

Cookies

Tests

Mock

Console

No Body

Preview

```
1 - {
2   "message": "Not authorized exception (401). No tienes permisos para
eliminar esta tarea",
3   "uri": "/tareas/67c1b09d5752f14749e82a5d"
4 }
```