

PRUEBAS GESTIÓN USUARIOS

Insomniac

Capturas del funcionamiento

Register

POST localhost:8080/usuarios/register Send 201 Created 3.7 s 33 B

Params Body Auth Headers 4 Scripts Docs Preview Headers 9 Cookies

JSON

```
1 {
2   "username": "Alpha",
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 }
14
```

Preview

```
1 {
2   "username": "Alpha",
3   "rol": "USER"
4 }
```

Si ya existe ese usuario

POST localhost:8080/usuarios/register Send 500 Internal Server Error 2.71 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

POSTlocalhost:8080/usuarios/register

Send

400 Bad Request29 ms95 BJust No

ParamsBodyAuthHeaders4ScriptsDocs

PreviewHeaders10CookiesTests0/0MockConsole

JSON

Preview

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

1- {
2 "message": "Bad Request Exception (400). Uno o más campos vacíos",
3 "uri": "/usuarios/register"
4 }

Login

POSTlocalhost:8080/usuarios/login

Send

201 Created2.95 s492 BJust Now

ParamsBodyAuthHeaders4ScriptsDocs

PreviewHeaders9CookiesTests0/0MockConsole

JSON

Preview

1- {
2 "username": "usuario123",
3 "password": "tu_contraseña"
4 }
5 }

1- {
2 "token":
 "eyJhbGciOiJIUzI1NiJ9.eyJ3ZmludCI6Zm9udC3ViIjoidXN1YXpjb2E5MyIsImV4cCI6MTk0MDE1MzY1NiwiaWF0IjoxNzQ5MTQ5NzU1LCJyb2xlcyciOiI1PTEVFNWVU1Ij9uLmV4cCQ5OeMR3cddF1xRbvnayXcu8j2NYNvxcC0gz2IA72Q007SKpe9N9XY5N004ni2cw3tSM56dw0_Ni56V5ae0Q7YalIMUoFMe7LRvqc8Zmog_Z-mvot8vds587UzpFr2TkSeIHu3rbou-060jTfLrG1Am2u3GeIjg1RtnCR-gFLV4vTH23k8eFUAv6AY7fC_qQc52vcu8exNIGbLPCFWvp9noyG-Qv--IhrdFPrPFRPLgde2rZ-ZC_3nQ3GsFw8ZYW7F90Italdy6ZVbxq9LKYHMR-B3Y0duaIlgTtsbbvj1CHmEPoYt_M07Igv1TvzBoz39H5-WaU_975Q"
3 }

si hay datos incorrectos

POSTlocalhost:8080/usuarios/login

Send

401 Unauthorized979 ms94 BJust Now

ParamsBodyAuthHeaders4ScriptsDocs

PreviewHeaders9CookiesTests0/0MockConsole

JSON

Preview

1- {
2 "username": "usuario123",
3 "password": "tu_contrasea"
4 }
5 }

1- {
2 "message": "Not authorized exception (401). Credenciales incorrectas",
3 "uri": "/usuarios/login"
4 }

Crear Tarea

POSTlocalhost:8080/tareas

Send

201 Created1.33 s153 B

ParamsBodyAuthHeaders4ScriptsDocs

PreviewHeaders9CookiesTests0/0MockConsole

JSON

Preview

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

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:

The screenshot shows a REST client interface with a POST request to `localhost:8080/tareas`. The request body is a JSON object: `{ "autor": "usuario2", "objetivo": "Preparar presentación de proyecto", "fecha": "2025-03-05T09:00:00.000Z" }`. The response is a 401 Unauthorized status with a message: `"message": "Not authorized exception (401). No tienes permisos para crear tareas para otros usuarios", "uri": "/tareas"`.

Si hay campos vacios

The screenshot shows a REST client interface with a POST request to `localhost:8080/tareas`. The request body is a JSON object: `{ "objetivo": "", "fecha": "2025-03-05T09:00:00.000Z" }`. The response is a 400 Bad Request status with a message: `"message": "Bad Request Exception (400). El objetivo de la tarea no puede estar vacío", "uri": "/tareas"`.

Obtener tareas

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

The screenshot shows a REST client interface with a GET request to `localhost:8080/tareas`. The response is a 200 OK status with a JSON array containing one task object: `[{ "id": "67c1b09d5752f14749e82a5d", "autor": "ab", "objetivo": "Preparar presentación de proyecto", "fecha": "2025-03-05T09:00:00.000+00:00", "completada": false }]`.

```
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

The screenshot shows a REST client interface with a PUT request to `localhost:8080/tareas/67c1b09d5752f14749e82a5d/completar`. The response is a 200 OK status with a 2.11 s response time and 152 B of data. The response body is a JSON object:

```
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

The screenshot shows a REST client interface with a PUT request to `localhost:8080/tareas/67c1b09d5752f14749e82a5d/completar`. The response is a 401 Unauthorized status with a 212 ms response time and 141 B of data. The response body is a JSON object:

```
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

The screenshot shows a REST client interface with a PUT request to `localhost:8080/tareas/64a3e4f12b7c1e8a9d3f5b2a/completar`. The response is a 400 Bad Request status with a 1.59 s response time and 113 B of data. The response body is a JSON object:

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

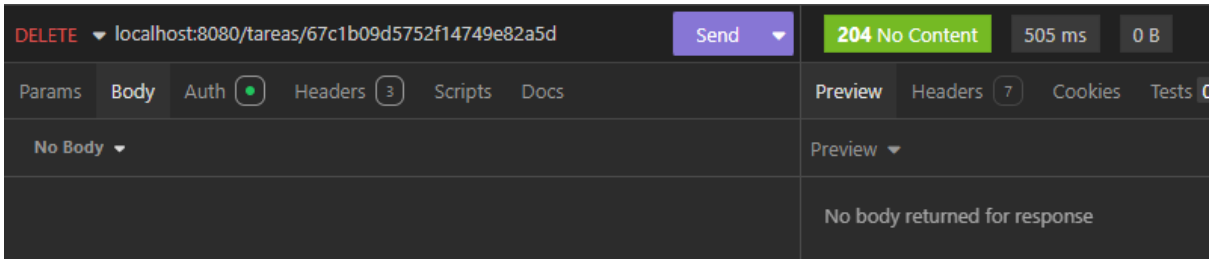
Desmarcar tarea como completada

Sigue la misma lógica que marcar como completada

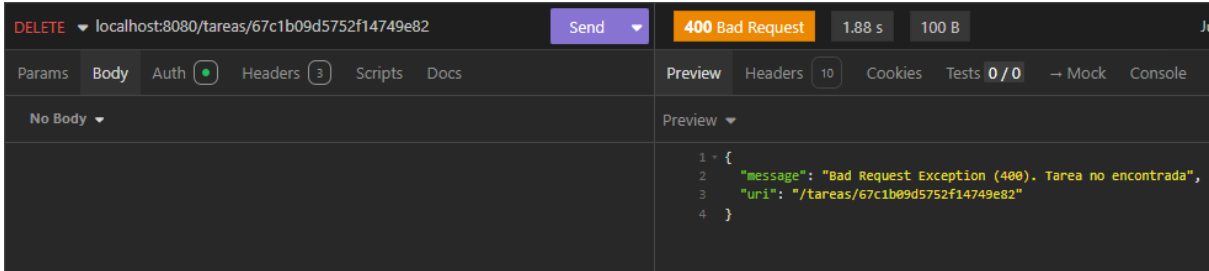
The screenshot shows a REST client interface with a PUT request to `localhost:8080/tareas/67c1b09d5752f14749e82a5d/descompletar`. The response is a 200 OK status with a 998 ms response time and 153 B of data. The response body is a JSON object:

```
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 }
```

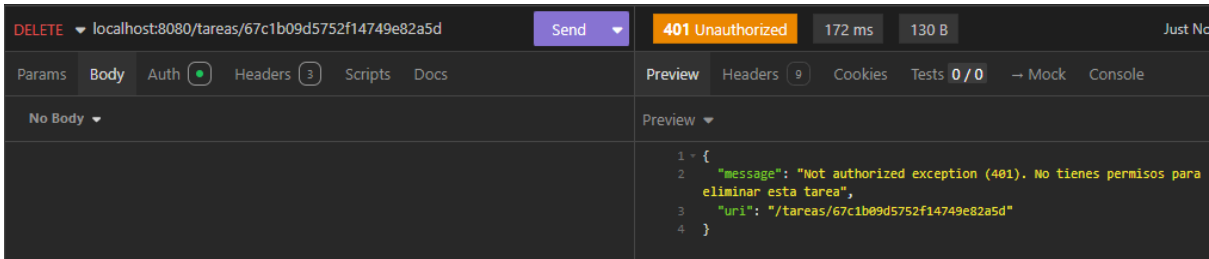
Eliminar Tarea



Si no existe la tarea



Si no eres el autor de la tarea o un admin



Interfaz Gráfica

A continuación, se muestra el funcionamiento de la interfaz gráfica de la aplicación. Se trata de un software de escritorio desarrollado con Jetpack Compose, diseñado para ofrecer una experiencia fluida e intuitiva. La aplicación se comunica con una API para obtener y gestionar datos en tiempo real.

Página de Inicio



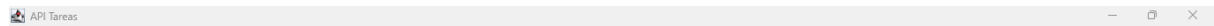
Inicia Sesión para entrar a las tareas

Iniciar sesión

Registrarte

Registro

Un ejemplo de que el registro funciona:



Registro de Usuario

Usuario

Daniela

Contraseña

Calle

Calle1

Número

1

Municipio

Cádiz

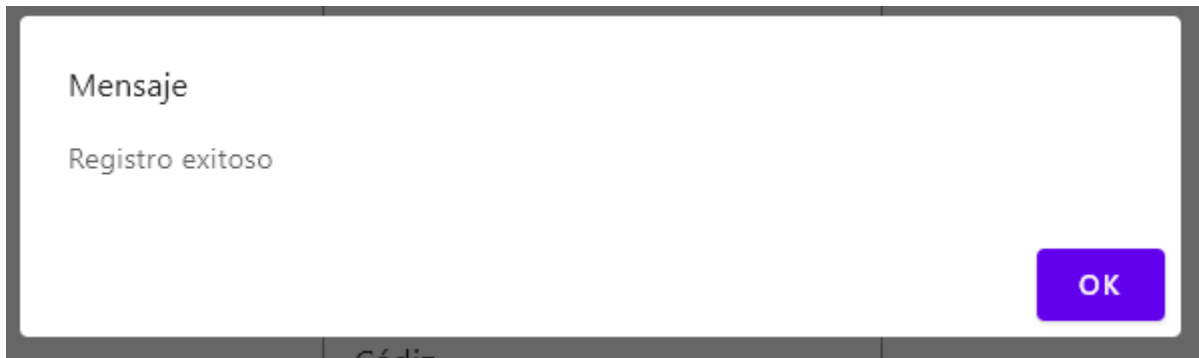
Provincia

Cádiz

Código Postal

11510

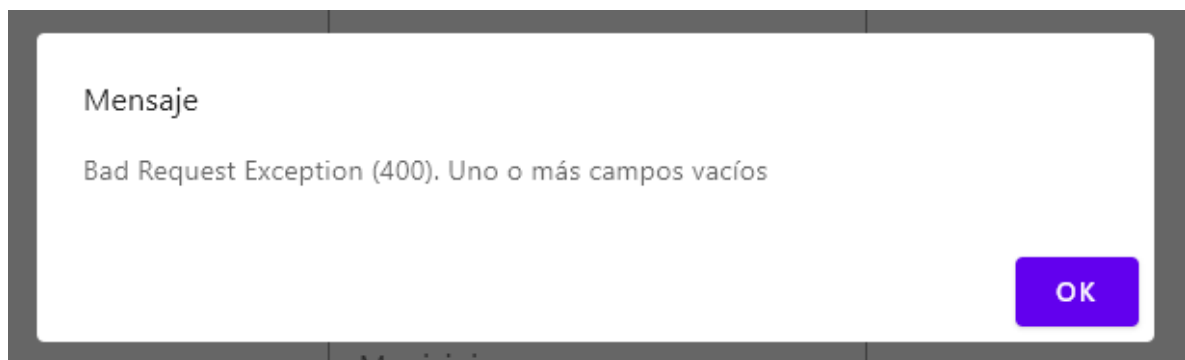
Registrarte

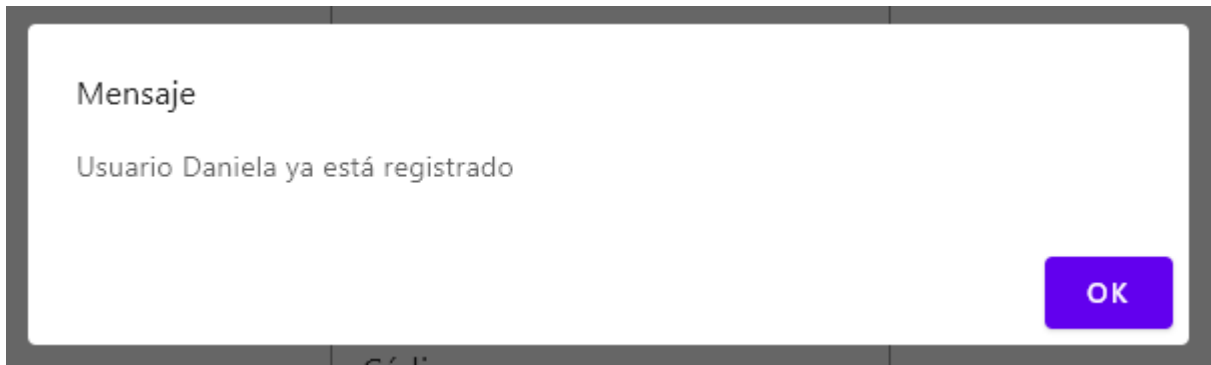


```
_id: ObjectId('67b86d8f4de1851ddb1286db')
username : "Daniela"
password : "$2a$10$VwDKRJwRnLPV0UtMmxVvA.U49BP5mrWpSuBp6xNmbbJzSRPRJwnZa"
roles : "USER"
▸ direccion : Object
  _class : "com.es.aplicacion.modelo.Usuario"
```

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

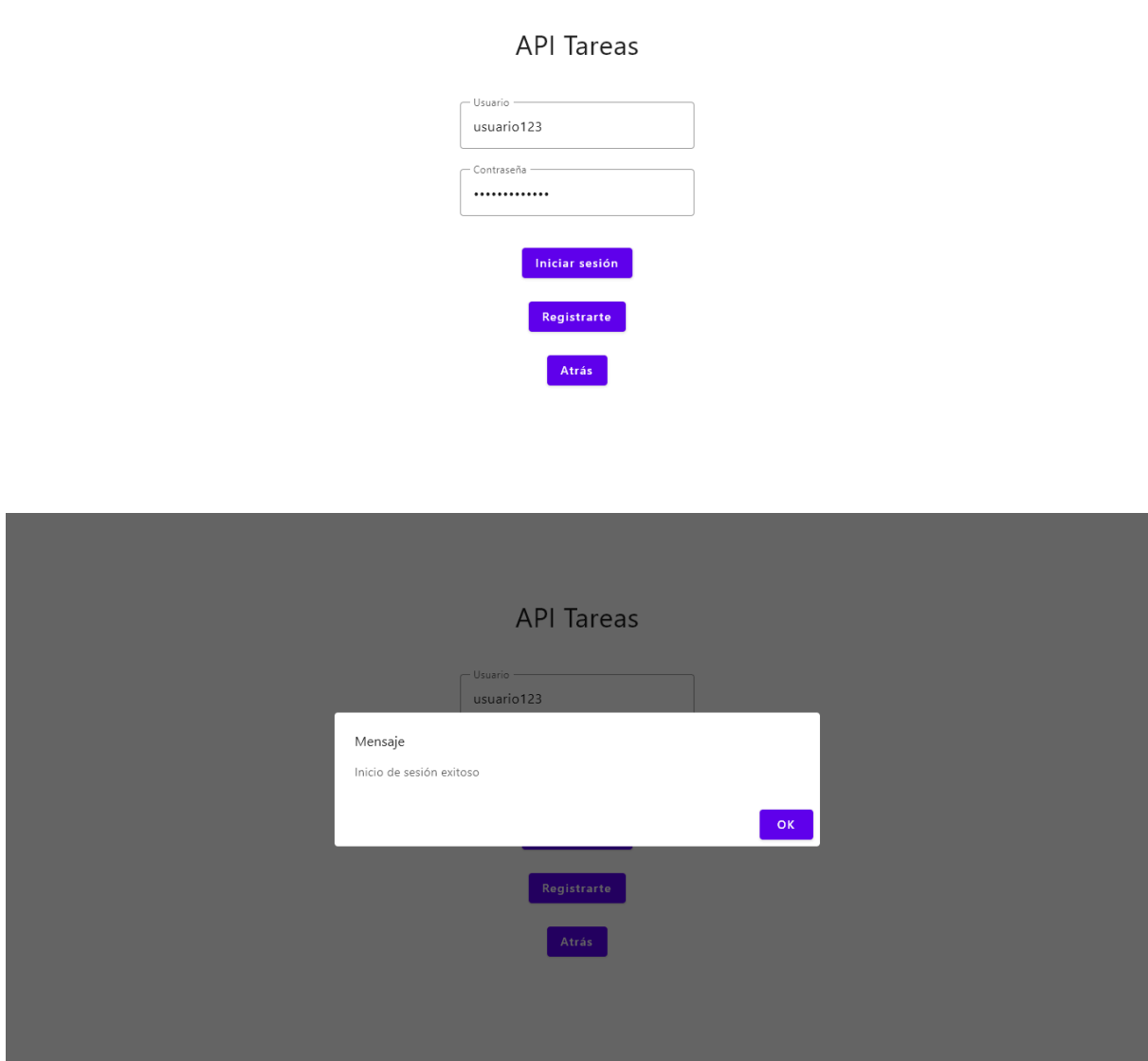
A screenshot of a web browser window titled "API Tareas". The page has a header "Registro de Usuario". Below the header, there is a vertical stack of seven input fields: "Usuario", "Contraseña", "Calle", "Número", "Municipio", "Provincia", and "Código Postal". At the bottom of the form is a blue button labeled "Registrarte".



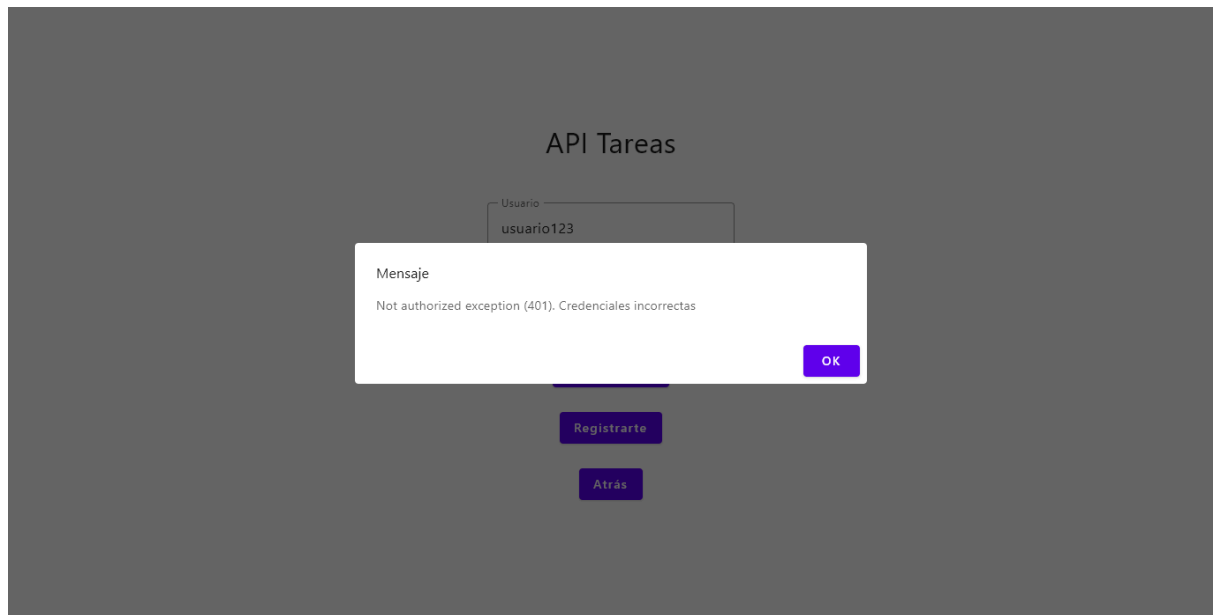


Login

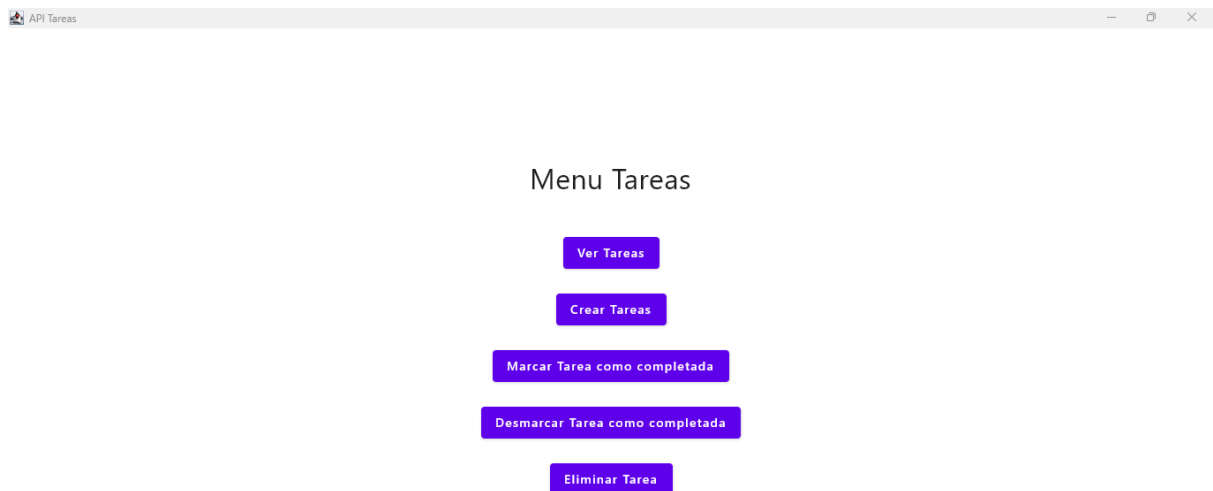
Un ejemplo de que el login funciona:



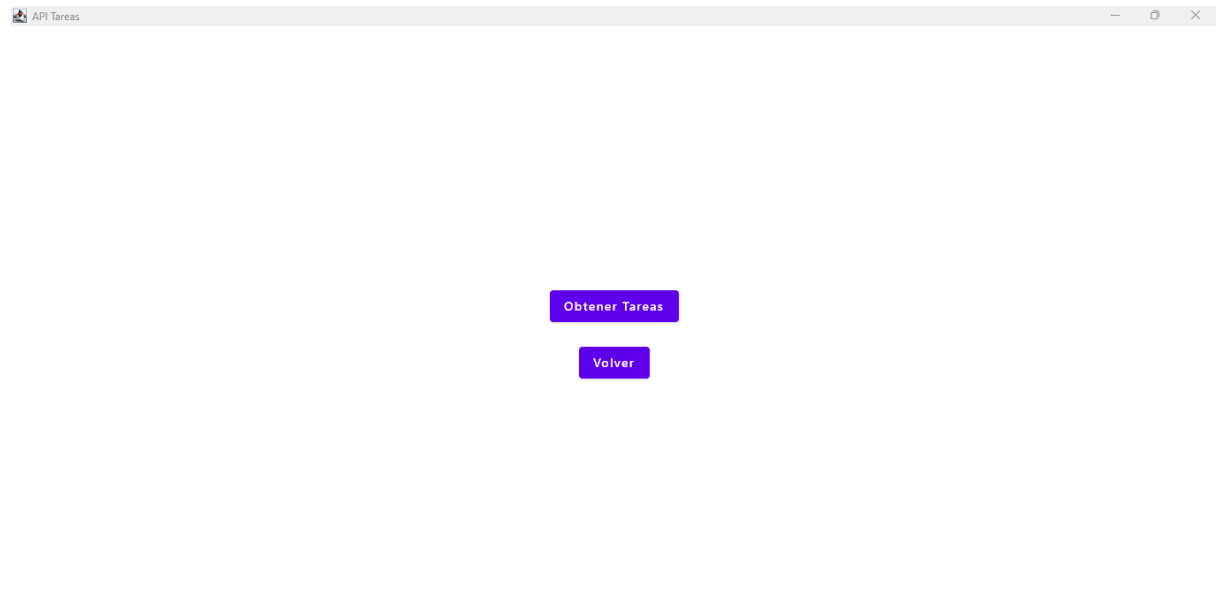
Ejemplo de que comunica cuando no funciona y te avisa por qué:



Menú Tareas



GET Tareas



Resultado al recibir las tareas



Create Tareas

API Tareas

Crear Nueva Tarea

Fecha
2025-03-05T09:00:00.000

Objetivo
Terminar Documentación

Crear Tarea

Volver

Resultado al crear la tarea

API Tareas

Crear Nueva Tarea

Fecha
2025-03-05T09:00:00.000

Objetivo
Terminar Documentación

Crear Tarea
201 Created

OK

Volver

Marcar como completada

API Tareas

Completar Tarea

ID de Tarea

67c1b09d5752f14749e82a5d

Marcar como Completada

Si encuentra la tareas

API Tareas

ID de Tarea

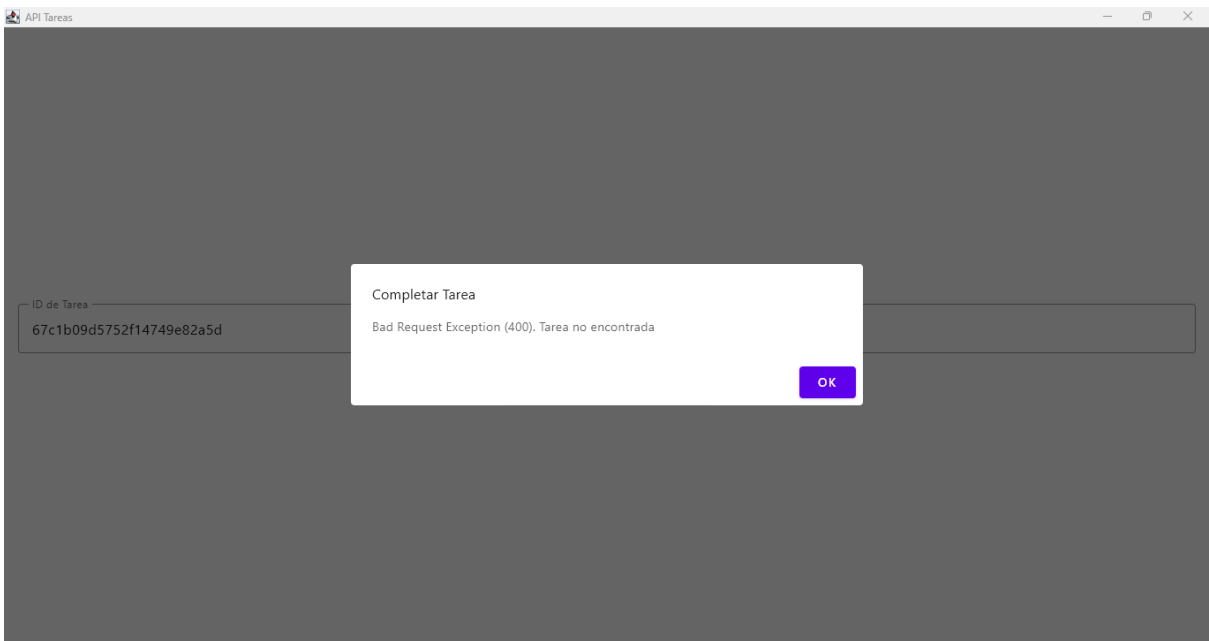
678e455defb2fa38c70d6f14

Completar Tarea

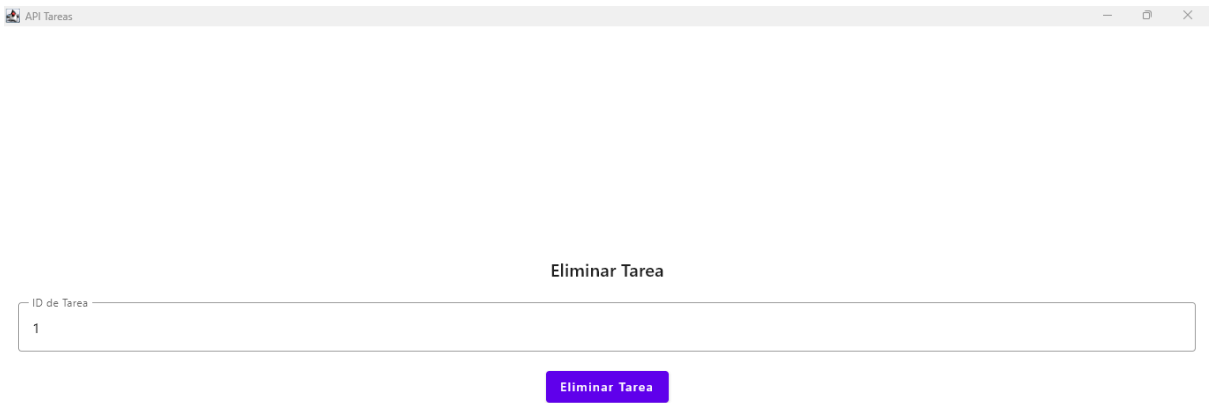
200 OK

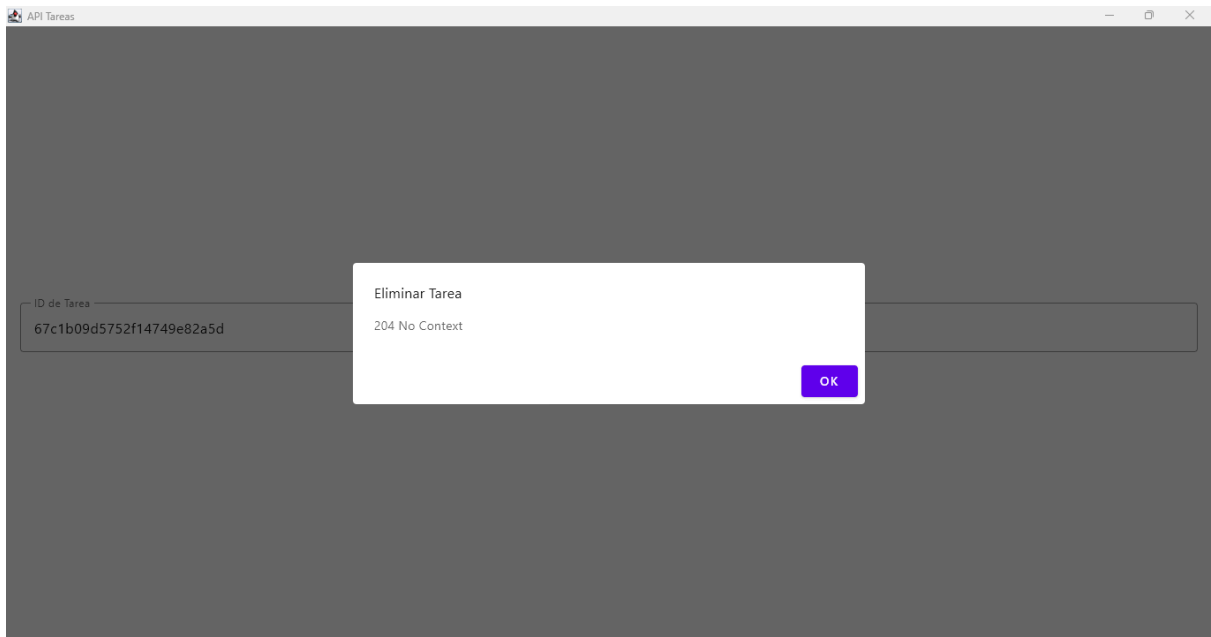
OK

Si no encuentra la tarea



Delete Tareas





Cuando no lo encuentra:

