

SISTEMAS ORIENTADOS A SERVICIOS

PRÁCTICA DE SERVICIO REST (GEOETSIINF)

NICOLÁS COSSÍO MIRAVALLS (190082)

LUIS ÁLVAREZ OSORIO

TELMO VIÚDEZ ESCOBAR (190084)

Introducción

El objetivo de esta práctica es el diseño e implementación de una API REST, incluyendo un cliente y servidor que hagan uso de la misma.

El propósito de esta API es permitir consultas y operaciones sobre una base de datos orientada al geocaching (**GeoETSIINF**), que almacena datos de usuarios (personas que añaden y buscan tesoros) y de los propios tesoros. Estas consultas y operaciones se realizan mediante peticiones y respuestas HTTP.

Las herramientas utilizadas para el desarrollo de la práctica son:

- Visual Studio Code como entorno de desarrollo
- Swagger Editor, para formalizar el diseño de la API
- Java 8, como lenguaje de programación
- Maven para la gestión de dependencias
- MySQL 8 como SGBD
- Tomcat 9 como servidor web
- Thunder Client (extensión de VSCode) como cliente HTTP externo

Hemos puesto especial atención en la representación como recurso de los tesoros y usuarios, creando un modelo común entre servidor de cliente para la comunicación mediante objetos Java transformados en **JSON**. Por tanto, nuestra API utiliza JSON como formato de representación.

En nuestra implementación de cliente todas las peticiones se envían a URIs referenciadas por los recursos, en vez de ser creadas mediante concatenaciones por el cliente. Esto permite un diseño conforme a **HATEOAS** y asegura la navegabilidad de la API, necesitando sólo conocer la URI del punto de entrada.

Conforme a la especificación de la práctica, además de esta memoria hemos incluido los siguientes ficheros:

- geoetsiinf.war (WAR del servidor con código fuente)
- client.zip (Proyecto Maven/Eclipse con código fuente del cliente)
- swagger.yml (Diseño de la API)
- create.sql (Ficheros para inicialización de la BD)

En los siguientes apartados, describimos el diseño de la API y los resultados de su utilización.

Diseño de API (Swagger)

Hemos usado Swagger Editor para documentar el diseño de nuestra API. Además de adjuntar el archivo YAML correspondiente, adjuntamos capturas de todos los métodos o recursos (algunas tienen las respuestas en la página siguiente debido a su extensión):

api This is the api entry point that allows the navigation to other resources ^

GET / Lists references to users and treasures resources ^

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	(OK) Successful operation	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "users": "http://localhost:8080/geoetsiinf/api/users",
  "treasures": "http://localhost:8080/geoetsiinf/api/treasures"
}
```

treasures

This endpoint is in charge of managing all user-added treasures related resources



POST

/users/{userName}/treasures

The given user adds a treasure

Parameters

Try it out

Name	Description
userName <small>* required</small>	User name of the user who wants to add a treasure
string (path)	<input type="text" value="obi-wan"/>

Request body required

application/json

Example Value

Schema

```
{
  "treasureName": "Lightsaber",
  "coordinateX": 54.023,
  "coordinateY": -84.023,
  "terrainType": "Rocky",
  "difficulty": 6,
  "size": 20.6,
  "hint": "Behind a T-shaped rock"
}
```

Responses

Code	Description	Links						
201	(OK) Successful operation	No links						
	Headers:							
	<table><thead><tr><th>Name</th><th>Description</th><th>Type</th></tr></thead><tbody><tr><td>Location</td><td>URL of the representation. Example: http://localhost:8080/geoetsiinf/api/users/obi-wan</td><td>string</td></tr></tbody></table>	Name	Description	Type	Location	URL of the representation. Example: http://localhost:8080/geoetsiinf/api/users/obi-wan	string	
Name	Description	Type						
Location	URL of the representation. Example: http://localhost:8080/geoetsiinf/api/users/obi-wan	string						
400	(BadRequest) Request body is not valid.	No links						
404	(NotFound) Requested resource was not found.	No links						

GET

/users/{userName}/treasures

Lists all treasures added by an user

^

Parameters

Try it out

Name	Description
userName * required	user of the user whose added treasures to be listed
string (path)	<input type="text" value="obi-wan"/>
before	filter by elements added before a given date
string(\$date) (query)	<input type="text" value="2021-03-17"/>
start	initial element to list
integer (query)	Default value : 10 <input type="text" value="1"/>
amount	number of elements to list
integer (query)	Default value : 10 <input type="text" value="5"/>
difficulty	Difficulty of the treasure
string (query)	<input type="text" value="Medium"/>

Responses

Code	Description	Links						
200	<div>(OK) Successful operation</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "items": [{ "href": "http://localhost:8080/geoetsiinf/api/treasures/1", "treasureName": "Lightsaber", "coordinateX": 54.023, "coordinateY": -84.023, "treasureId": 504, "terrainType": "Rocky", "difficulty": 6, "size": 20.6, "hint": "Behind a T-shaped rock", "dateAdded": "2021-03-17" }] }</pre></div> <div>Headers:</div> <table><tr><th>Name</th><th>Description</th><th>Type</th></tr><tr><td>Content-Location</td><td>URL of the representation. Example: http://localhost:8080/geoetsiinf/api/users/obi-one/treasures</td><td>string</td></tr></table>	Name	Description	Type	Content-Location	URL of the representation. Example: http://localhost:8080/geoetsiinf/api/users/obi-one/treasures	string	No links
Name	Description	Type						
Content-Location	URL of the representation. Example: http://localhost:8080/geoetsiinf/api/users/obi-one/treasures	string						
404	<div>(NotFound) Requested resource was not found.</div>	No links						

PUT

/users/{userName}/treasures/{treasureId}

Updates a treasure

Parameters

Try it out

Name	Description
userName * required	User name of the user
string (path)	Example : obi-wan
	<input type="text" value="obi-wan"/>
treasureId * required	number that identifies a treasure
integer (path)	<input type="text" value="500"/>

Request body required

application/json

Example Value | Schema

```
{
  "treasureName": "Lightsaber",
  "coordinateX": 54.023,
  "coordinateY": -84.023,
  "terrainType": "Rocky",
  "difficulty": 6,
  "size": 20.6,
  "hint": "Behind a T-shaped rock"
}
```

Responses

Code	Description	Links						
200	(OK) Succesful operation	No links						
	Headers:							
	<table><thead><tr><th>Name</th><th>Description</th><th>Type</th></tr></thead><tbody><tr><td>Content-Location</td><td>URL of the representation. Example: http://localhost:8080/geoetsiinf/api/users/obi-wan/treasures/1</td><td>string</td></tr></tbody></table>	Name	Description	Type	Content-Location	URL of the representation. Example: http://localhost:8080/geoetsiinf/api/users/obi-wan/treasures/1	string	
Name	Description	Type						
Content-Location	URL of the representation. Example: http://localhost:8080/geoetsiinf/api/users/obi-wan/treasures/1	string						
400	(BadRequest) Request body is not valid.	No links						
404	(NotFound) Requested resource was not found.	No links						

DELETE

/users/{userName}/treasures/{treasureId}

Deletes a treasure

^

Parameters

Try it out

Name	Description
userName * required	User name of the user
string	Example : obi-wan
(path)	<input type="text" value="obi-wan"/>
treasureId * required	number that identifies a treasure
integer	
(path)	<input type="text" value="500"/>

Responses

Code	Description	Links
204	(NoContent) Succesful operation with no content in reponse	No links
404	(NotFound) Requested resource was not found.	No links

GET**/treasures** Lists all treasures added by an user which are closed to a location**Parameters****Try it out****Name****Description****coordinateX** * required

filter by coordinateX

number

(query)

54.023

coordinateY * required

filter by coordinateY

number

(query)

54.023

before

filter by elements added before a given date

string(\$date)

(query)

2021-03-17

start

initial element to list

integer

(query)

Default value : 10

1

amount

number of elements to list

integer

(query)

Default value : 10

5

difficulty

Difficulty of the treasure

string

(query)

Medium

Responses

Code	Description	Links
200	(OK) Successful operation	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "items": [
    {
      "href": "http://localhost:8080/geoetsiinf/api/treasures/1",
      "treasureName": "Lightsaber",
      "coordinateX": 54.023,
      "coordinateY": -84.023,
      "treasureId": 504,
      "terrainType": "Rocky",
      "difficulty": 6,
      "size": 20.6,
      "hint": "Behind a T-shaped rock",
      "dateAdded": "2021-03-17"
    }
  ]
}
```

Headers:

Name	Description	Type
Content-Location	URL of the representation. Example: http://localhost:8080/geoetsiinf/api/users/treasures	string

users

This endpoint is in charge of managing all users related resources

^

POST

/users Registers an user in the system

^

Parameters

Try it out

No parameters

Request body required

application/json

Example Value

Schema

```
{
  "firstName": "Obi-Wan",
  "lastName": "Kenovi",
  "userName": "obiwan",
  "email": "obiwankenobi@jediorder.org",
  "postalCode": 28015
}
```

Responses

Code	Description	Links						
201	(OK) Succesful operation	No links						
Headers:								
	<table><thead><tr><th>Name</th><th>Description</th><th>Type</th></tr></thead><tbody><tr><td>Location</td><td>URL of the representation. Example: http://localhost:8080/geoetsiinf/api/users/obi-wan</td><td>string</td></tr></tbody></table>	Name	Description	Type	Location	URL of the representation. Example: http://localhost:8080/geoetsiinf/api/users/obi-wan	string	
Name	Description	Type						
Location	URL of the representation. Example: http://localhost:8080/geoetsiinf/api/users/obi-wan	string						
400	(BadRequest) Request body is not valid.	No links						

GET

/users List all registered users by default

Parameters

Try it out

Name	Description
firstName	filters by first name or pattern.
string (query)	Example : Ana -> returns Analuz, Anaking

Ana -> returns Analuz, Anaking

Responses

Code	Description	Links
200	(OK) Succesful operation	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{  "items": [    {      "firstName": "Obi-Wan",      "href": "http://localhost:8080/geoetsiinf/api/users/obiwan",      "friends": "http://localhost:8080/geoetsiinf/api/users/obiwan/friends",      "treasures": "http://localhost:8080/geoetsiinf/api/users/obiwan/treasures",      "foundTreasures": "http://localhost:8080/geoetsiinf/api/users/obiwan/foundTreasures",      "lastName": "Kenovi",      "userName": "obiwan",      "email": "obiwankenobi@jedior.org",      "postalCode": 28015    }  ]}
```

Headers:

Name	Description	Type
Content-Location	URL of the representation. Example: http://localhost:8080/geoetsiinf/api/users	string

GET

/users/{userName} Returns the info of an user

Parameters

Try it out

Name	Description
userName <small>* required</small>	User name of the user
string (path)	Example : obi-wan
	<div>obi-wan</div>

Responses

Code	Description	Links						
200	(OK) Succesful operation	No links						
	<div>Media type</div> <div><div>application/json</div></div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "firstName": "Obi-Wan", "href": "http://localhost:8080/geoetsiinf/api/users/obiwan", "friends": "http://localhost:8080/geoetsiinf/api/users/obiwan/friends", "treasures": "http://localhost:8080/geoetsiinf/api/users/obiwan/treasures", "foundTreasures": "http://localhost:8080/geoetsiinf/api/users/obiwan/foundTreasures", "lastName": "Kenovi", "userName": "obiwan", "email": "obiwankenobi@jediorder.org", "postalCode": 28015 }</pre></div> <div>Headers:</div> <table><thead><tr><th>Name</th><th>Description</th><th>Type</th></tr></thead><tbody><tr><td>Content-Location</td><td>URL of the representation. Example: http://localhost:8080/geoetsiinf/api/users/obi-one</td><td>string</td></tr></tbody></table>	Name	Description	Type	Content-Location	URL of the representation. Example: http://localhost:8080/geoetsiinf/api/users/obi-one	string	
Name	Description	Type						
Content-Location	URL of the representation. Example: http://localhost:8080/geoetsiinf/api/users/obi-one	string						
404	(NotFound) Requested resource was not found.	No links						

PUT

/users/{userName} Updates an user

Parameters

Try it out

Name	Description
userName * required	User name of the user to be updated.
string (path)	<input type="text" value="obi-wan"/>

Request body required

application/json

Example Value | Schema

```
{  "firstName": "Obi-Wan",  "lastName": "Kenovi",  "userName": "obiwan",  "email": "obiwankenobi@jediorder.org",  "postalCode": 28015}
```

Responses

Code	Description	Links						
200	(OK) Successful operation	No links						
	Headers:							
	<table><thead><tr><th>Name</th><th>Description</th><th>Type</th></tr></thead><tbody><tr><td>Content-Location</td><td>URL of the representation. Example: http://localhost:8080/geoetsiinf/api/users/obi-wan</td><td>string</td></tr></tbody></table>	Name	Description	Type	Content-Location	URL of the representation. Example: http://localhost:8080/geoetsiinf/api/users/obi-wan	string	
Name	Description	Type						
Content-Location	URL of the representation. Example: http://localhost:8080/geoetsiinf/api/users/obi-wan	string						
400	(BadRequest) Request body is not valid.	No links						
404	(NotFound) Requested resource was not found.	No links						

DELETE

/users/{userName} Deletes an user

Parameters

Try it out

Name	Description
userName <small>* required</small>	User name of the user to be deleted.
string (path)	<input type="text" value="obi-wan"/>

Responses

Code	Description	Links
204	(NoContent) Succesful operation with no content in reponse	No links
404	(NotFound) Requested resource was not found.	No links

GET

/users/{userName}/foundTreasures

Lists all found treasures by an user

^

Parameters

Try it out

Name	Description
userName * required	User name of the user
string (path)	<input type="text" value="obi-wan"/>
before	filter by elements added before a given date
string(\$date) (query)	<input type="text" value="2021-03-17"/>
start	initial element to list
integer (query)	Default value : 10 <input type="text" value="1"/>
amount	number of elements to list
integer (query)	Default value : 10 <input type="text" value="5"/>
difficulty	Difficulty of the treasure
string (query)	<input type="text" value="Medium"/>

Responses

Code	Description	Links
------	-------------	-------

200	(OK) Successful operation	No links
-----	---------------------------	----------

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "items": [
    {
      "href": "http://localhost:8080/geoetsiinf/api/treasures/1",
      "treasureName": "Lightsaber",
      "coordinateX": 54.023,
      "coordinateY": -84.023,
      "treasureId": 504,
      "terrainType": "Rocky",
      "difficulty": 6,
      "size": 20.6,
      "hint": "Behind a T-shaped rock",
      "dateAdded": "2021-03-17"
    }
  ]
}
```

Headers:

Name	Description	Type
Content-Location	URL of the representation. Example: http://localhost:8080/geoetsiinf/api/users/obi-one/foundTreasures	string

404	(NotFound) Requested resource was not found.	No links
-----	--	----------

PUT

/users/{userName}/foundTreasures/{treasureId}

Adds a treasure to an user's found treasures list

Parameters

Try it out

Name	Description
userName * required	User name of the user
string (path)	Example : obi-wan
	<input type="text" value="obi-wan"/>
treasureId * required	number that identifies a treasure
integer (path)	
	<input type="text" value="500"/>

Responses

Code	Description	Links						
200	(OK) Successful operation	No links						
	Headers:							
	<table><thead><tr><th>Name</th><th>Description</th><th>Type</th></tr></thead><tbody><tr><td>Content-Location</td><td>URL of the representation. Example: http://localhost:8080/geoetsiinf/api/users/obi-wan/treasures/1</td><td>string</td></tr></tbody></table>	Name	Description	Type	Content-Location	URL of the representation. Example: http://localhost:8080/geoetsiinf/api/users/obi-wan/treasures/1	string	
Name	Description	Type						
Content-Location	URL of the representation. Example: http://localhost:8080/geoetsiinf/api/users/obi-wan/treasures/1	string						
400	(BadRequest) Request body is not valid.	No links						
404	(NotFound) Requested resource was not found.	No links						

GET

/users/{userName}/friends

Lists all friends of an user

^

Parameters

Try it out

Name	Description
userName * required	User name of the user
string (path)	<input type="text" value="obi-wan"/>
name	filters by first name or pattern.
string (query)	Example : Ana -> returns Analuz, Anakin <input type="text" value="Ana -> returns Analuz, Anakin"/>
start	initial element to list
integer (query)	Default value : 10 <input type="text" value="1"/>
amount	number of elements to list
integer (query)	Default value : 10 <input type="text" value="5"/>

Responses

Code	Description	Links
200	(OK) Successful operation	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "items": [
    {
      "firstName": "Obi-Wan",
      "href": "http://localhost:8080/geoetsiinf/api/users/obiwan",
      "friends": "http://localhost:8080/geoetsiinf/api/users/obiwan/friends",
      "treasures": "http://localhost:8080/geoetsiinf/api/users/obiwan/treasures",
      "foundTreasures": "http://localhost:8080/geoetsiinf/api/users/obiwan/foundTreasures",
      "lastName": "Kenobi",
      "userName": "obiwan",
      "email": "obiwankenobi@jediorder.org",
      "postalCode": 28015
    }
  ]
}
```

Headers:

Name	Description	Type
Content-Location	URL of the representation. Example: http://localhost:8080/geoetsiinf/api/user/obi-one/friends	string

404	(NotFound) Requested resource was not found.	No links
-----	--	----------

PUT**/users/{userName}/friends/{friendUserName}** Adds a friend to an user's friend list**Parameters**[Try it out](#)**Name****Description****userName** * required

User name of the user

string

Example : obi-wan

(path)

obi-wan

friendUserName * required

User name of the user to be added as a friend

string

(path)

luke

Responses**Code****Description****Links**

200

(OK) Successful operation

*No links***Headers:****Name****Description****Type**Content-
LocationURL of the representation. Example:
<http://localhost:8080/geoetsiinf/api/users/obi-wan/friends/obi-wan>

string

400


(BadRequest) Request body is not valid.

No links

404

(NotFound) Requested resource was not found.

No links

DELETE `/users/{userName}/friends/{friendUserName}` Deletes a friend from an user's friend list 

Parameters

[Try it out](#)

Name	Description
userName * required string (path)	User name of the user <i>Example</i> : obi-wan <input type="text" value="obi-wan"/>
friendUserName * required integer (path)	User name of the user to be deleted <input type="text" value="500"/>

Responses

Code	Description	Links
204	(NoContent) Succesful operation with no content in reponse	<i>No links</i>
404	(NotFound) Requested resource was not found.	<i>No links</i>

GET

/users/{userName}/profile

Profile of an user. Contains the user's personal info, the amount of found treasures, the last 5 found treasures, the amount of friends and the amount of added treasures by the user

Parameters

Try it out

Name	Description
userName * required	User name of the user
string (path)	<input type="text" value="obi-wan"/>

Responses

Code	Description	Links
200	(OK) Successful operation	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "userInfo": {
    "firstName": "Obi-Wan",
    "href": "http://localhost:8080/geoetsiinf/api/users/obiwan",
    "friends": "http://localhost:8080/geoetsiinf/api/users/obiwan/friends",
    "treasures": "http://localhost:8080/geoetsiinf/api/users/obiwan/treasures",
    "foundTreasures": "http://localhost:8080/geoetsiinf/api/users/obiwan/foundTreasures",
    "lastName": "Kenobi",
    "userName": "obiwan",
    "email": "obiwankenobi@jediorder.org",
    "postalCode": 28015
  },
  "foundTreasuresAmount": 67,
  "foundTreasuresList": {
    "items": [
      {
        "href": "http://localhost:8080/geoetsiinf/api/treasures/1",
        "treasureName": "Lightsaber",
        "coordinateX": 54.023,
        "coordinateY": -84.023,
        "treasureId": 504,
      }
    ]
  }
}
```

Headers:

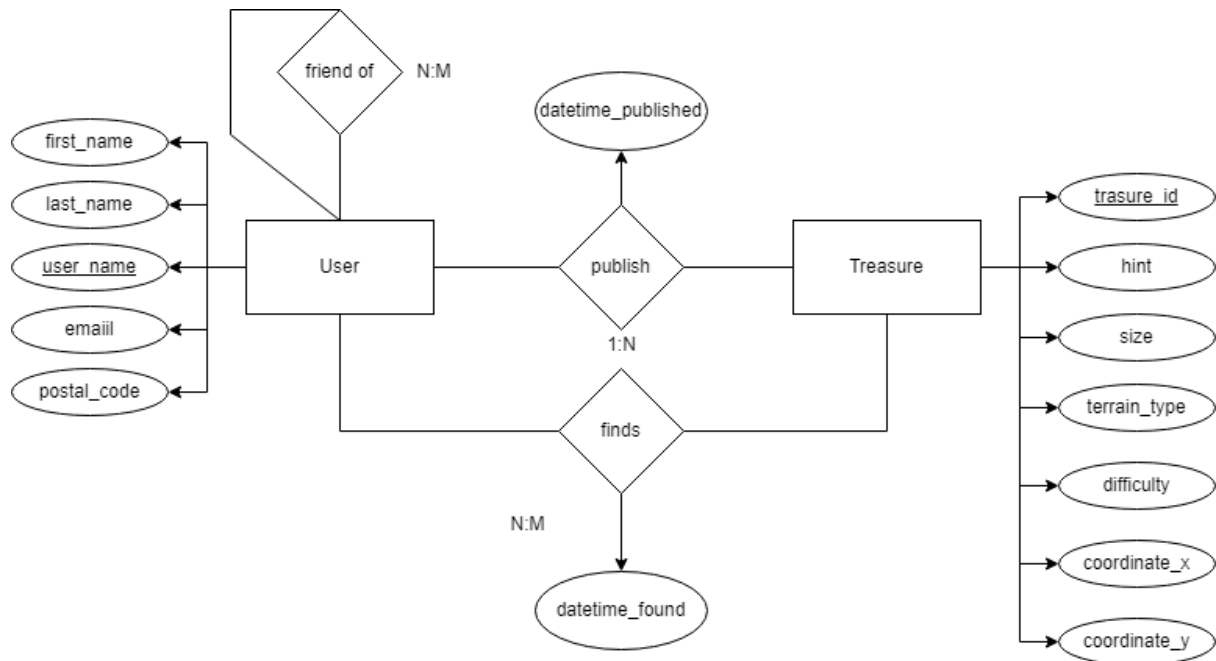
Name	Description	Type
Content-Location	URL of the representation. Example: http://localhost:8080/geoetsiinf/api/users/obi-one/profile	string

404	(NotFound) Requested resource was not found.	No links
-----	--	----------

Persistencia con base de datos

Nuestra implementación de servidor se apoya en una base de datos relacional utilizando el sistema de gestión de bases de datos **MySQL 8.0.26**.

Hemos utilizado el siguiente diagrama E/R para crear nuestra base de datos:



El servidor se autentica con los siguientes parámetros:

- Usuario: **geoetsiinf**
- Contraseña: **1234**

Para crear tanto este usuario con los permisos necesarios, como la base de datos y su estructura, hemos preparado un fichero **create.sql** con el siguiente contenido:

```

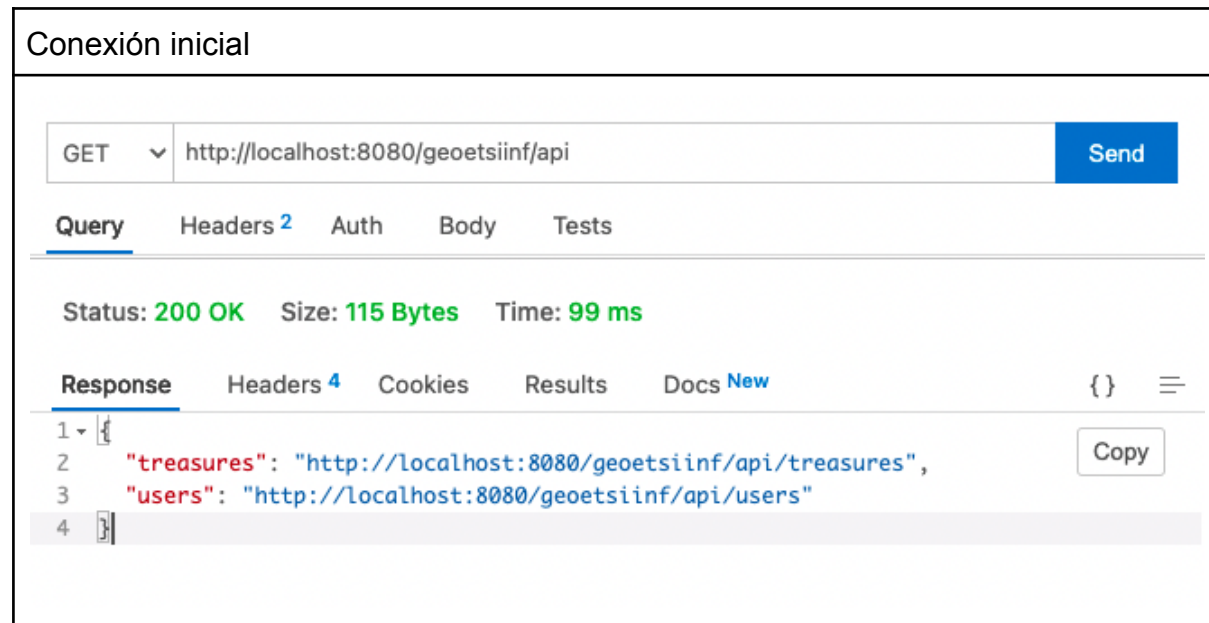
CREATE USER 'geoetsiinf'@'localhost' IDENTIFIED BY '1234';
CREATE DATABASE geoetsiinfdb;
GRANT ALL PRIVILEGES on geoetsiinfdb.* to 'geoetsiinf'@'localhost';

USE geoetsiinfdb;
CREATE TABLE users (
    first_name VARCHAR(255) NOT NULL,
    last_name VARCHAR(255) NOT NULL,
    user_name VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL UNIQUE,
    postal_code INT NOT NULL,
    user_number int NOT NULL AUTO_INCREMENT,
    KEY (user_number),
    PRIMARY KEY(user_name)
);
CREATE TABLE treasures (
    treasure_name VARCHAR(255) NOT NULL,
    coordinate_x FLOAT NOT NULL,
    coordinate_y FLOAT NOT NULL,
    terrain_type VARCHAR(225) NOT NULL,
    difficulty INT NOT NULL,
    size FLOAT NOT NULL,
    hint VARCHAR(255) NOT NULL,
    author_user_name VARCHAR(255) NOT NULL,
    date_added TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    treasure_id INT AUTO_INCREMENT,
    FOREIGN KEY (author_user_name) REFERENCES users(user_name) ON UPDATE CASCADE ON
DELETE CASCADE,
    PRIMARY KEY(treasure_id)
);
CREATE TABLE found_treasures (
    user_name VARCHAR(255),
    treasure_id INT,
    FOREIGN KEY (user_name) REFERENCES users(user_name) ON UPDATE CASCADE ON DELETE
CASCADE,
    FOREIGN KEY (treasure_id) REFERENCES treasures(treasure_id) ON UPDATE CASCADE ON
DELETE CASCADE,
    PRIMARY KEY(user_name, treasure_id)
);
CREATE TABLE friends (
    user_name_1 VARCHAR(255),
    user_name_2 VARCHAR(255),
    FOREIGN KEY (user_name_1) REFERENCES users(user_name) ON UPDATE CASCADE ON
DELETE CASCADE,
    FOREIGN KEY (user_name_2) REFERENCES users(user_name) ON UPDATE CASCADE ON
DELETE CASCADE,
    PRIMARY KEY(user_name_1, user_name_2)
);

```

Ejemplos de ejecución con Thunder Client (cliente HTTP)

Conexión inicial



GET ⌵ http://localhost:8080/geoetsiinf/api Send

Query Headers ² Auth Body Tests

Status: **200 OK** Size: **115 Bytes** Time: **99 ms**

Response Headers ⁴ Cookies Results Docs ^{New} {} ≡

```
1 {  
2   "treasures": "http://localhost:8080/geoetsiinf/api/treasures",  
3   "users": "http://localhost:8080/geoetsiinf/api/users"  
4 }
```

Copy

Añadiendo un usuario

POST

http://localhost:8080/geoetsiinf/api/users

Send

Query

Headers 2

Auth

Body 1

Tests

Json

Xml

Text

Form

Form-encode

GraphQL

Binary

Json Content

1

{

2

"firstName": "Obi-Wan",

3

"lastName": "Kenovi",

4

"userName": "obiwan",

5

"email": "obiwankenobi@jediorder.org",

6

"postalCode": 28015

7

}

Format

Status: 201 Created

Size: 0 Bytes

Time: 320 ms

Response

Headers 4

Cookies

Results

Docs New

{ }

≡

Response Headers

Header	Value
location	http://localhost:8080/geoetsiinf/api/users/obiwan
content-length	0
date	Fri, 15 Apr 2022 18:23:50 GMT
connection	close

Obteniendo una lista con todos los usuarios añadidos.

GET

http://localhost:8080/geoetsiinf/api/users

Send

Query

Headers 2

Auth

Body 1

Tests

Json

Xml

Text

Form

Form-encode

Graphql

Binary

Json Content

1

2

3

4

5

6

7

```
{
  "firstName": "Obi-Wan",
  "lastName": "Kenovi",
  "userName": "obiwan",
  "email": "obiwankenobi@jediorder.org",
  "postalCode": 28015
}
```

Format

Status: 200 OK Size: 478 Bytes Time: 44 ms

Response

Headers 4

Cookies

Results

Docs New

{}

≡

1

2

3

4

5

6

7

8

9

10

11

12

13

14

```
{
  "email": "obiwankenobi@jediorder.org",
  "firstName": "Obi-Wan",
  "foundTreasures": "http://localhost:8080/geoetsiinf/api/users/obiwan/foundTreasures",
  "friends": "http://localhost:8080/geoetsiinf/api/users/obiwan/friends",
  "href": "http://localhost:8080/geoetsiinf/api/users/obiwan",
  "lastName": "Kenovi",
  "postalCode": 28015,
  "profile": "http://localhost:8080/geoetsiinf/api/users/obiwan/profile",
  "treasures": "http://localhost:8080/geoetsiinf/api/users/obiwan/treasures",
  "userName": "obiwan"
}
```

Copy

Obteniendo un usuario particular usando su nombre de usuario

GET

http://localhost:8080/geoetsiinf/api/users/obiwan

Send

Query

Headers 2

Auth

Body 1

Tests

Json

Xml

Text

Form

Form-encode

Graphql

Binary

Json Content

1

2

3

4

5

6

7

"firstName": "Obi-Wan",

"lastName": "Kenovi",

"userName": "obiwan",

"email": "obiwankenobi@jedioder.org",

"postalCode": 28015

Format

Status: 200 OK

Size: 476 Bytes

Time: 15 ms

Response

Headers 4

Cookies

Results

Docs New

{}

≡

1

2

3

4

5

6

7

8

9

10

11

12

"email": "obiwankenobi@jedioder.org",

"firstName": "Obi-Wan",

"foundTreasures": "http://localhost:8080/geoetsiinf/api/users/obiwan/foundTreasures",

"friends": "http://localhost:8080/geoetsiinf/api/users/obiwan/friends",

"href": "http://localhost:8080/geoetsiinf/api/users/obiwan",

"lastName": "Kenovi",

"postalCode": 28015,

"profile": "http://localhost:8080/geoetsiinf/api/users/obiwan/profile",

"treasures": "http://localhost:8080/geoetsiinf/api/users/obiwan/treasures",

"userName": "obiwan"

Copy

Cambiando datos de un usuario ya existente

PUT

http://localhost:8080/geoetsiinf/api/users/obiwan

Send

Query

Headers 2

Auth

Body 1

Tests

Json

Xml

Text

Form

Form-encode

GraphQL

Binary

Json Content

```
1 {
2   "firstName": "Obi-Wan",
3   "lastName": "Kenobi",
4   "userName": "obiwan",
5   "email": "nuevoemail@jediorder.org",
6   "postalCode": 12345
7 }
```

Format

Status: 200 OK

Size: 474 Bytes

Time: 29 ms

Response

Headers 4

Cookies

Results

Docs New

{}

≡

```
1 {
2   "email": "nuevoemail@jediorder.org",
3   "firstName": "Obi-Wan",
4   "foundTreasures": "http://localhost:8080/geoetsiinf/api/users/obiwan/foundTreasures",
5   "friends": "http://localhost:8080/geoetsiinf/api/users/obiwan/friends",
6   "href": "http://localhost:8080/geoetsiinf/api/users/obiwan",
7   "lastName": "Kenobi",
8   "postalCode": 12345,
9   "profile": "http://localhost:8080/geoetsiinf/api/users/obiwan/profile",
10  "treasures": "http://localhost:8080/geoetsiinf/api/users/obiwan/treasures",
11  "userName": "obiwan"
12 }
```

Borrando un usuario particular usando usando su nombre de usuario

DELETE

http://localhost:8080/geoetsiinf/api/users/obiwan

Send

Query

Headers 2

Auth

Body 1

Tests

Json

Xml

Text

Form

Form-encode

Graphql

Binary

Json Content

Status: 204 No Content

Size: 0 Bytes

Time: 6 ms

Response

Headers 2

Cookies

Results

Docs New

{ }

Response Headers

Header	Value
date	Fri, 15 Apr 2022 18:29:24 GMT
connection	close

Un usuario creando un tesoro

POST

http://localhost:8080/geoetsiinf/api/users/obiwan/treasures

Send

Query

Headers 2

Auth

Body 1

Tests

Json

Xml

Text

Form

Form-encode

GraphQL

Binary

Json Content

```
1 {
2   "treasureName": "Lightsaber",
3   "coordinateX": 54.023,
4   "coordinateY": -84.023,
5   "terrainType": "Rocky",
6   "difficulty": 6,
7   "size": 20.6,
8   "hint": "Behind a T-shaped rock"
9 }
```

Format

Status: 201 Created Size: 0 Bytes Time: 55 ms

Response

Headers 4

Cookies

Results

Docs New

{}

≡

Response Headers

Header	Value
location	http://localhost:8080/geoetsiinf/api/treasures/obiwan/1
content-length	0
date	Fri, 15 Apr 2022 18:31:13 GMT
connection	close

Obteniendo una lista con todos los tesoros añadidos por un usuario

GET ⌵ http://localhost:8080/geoetsiinf/api/users/obiwan/treasures Send

Query Headers ² Auth Body ¹ Tests

Status: 200 OK Size: 635 Bytes Time: 9 ms

Response Headers ⁴ Cookies Results Docs ^{New} {} ≡

1 ⌵ ⌵

2 ⌵ {

3 "authorUserName": "obiwan",

4 "coordinateX": 54.02299880981445,

5 "coordinateY": -84.02300262451172,

6 "dateAdded": "2022-04-15 20:34:10",

7 "difficulty": 6,

8 "hint": "Behind a T-shaped rock",

9 "href": "http://localhost:8080/geoetsiinf/api/treasures/2",

10 "size": 20.600000381469727,

11 "terrainType": "Rocky",

12 "treasureId": 2,

13 "treasureName": "Lightsaber"

14 },

15 ⌵ {

16 "authorUserName": "obiwan",

17 "coordinateX": 123.02300262451172,

18 "coordinateY": -92.02300262451172,

19 "dateAdded": "2022-04-15 20:36:02",

20 "difficulty": 10,

21 "hint": "Where the eyes can't see",

22 "href": "http://localhost:8080/geoetsiinf/api/treasures/3",

23 "size": 11.5,

24 "terrainType": "Flat",

25 "treasureId": 3,

26 "treasureName": "The force"

27 }

28 ⌵

Copy

Obteniendo una lista con todos los tesoros que cumplen las condiciones dadas y ordenados según cercanía a las coordenadas dadas,

GET

http://localhost:8080/geoetsiinf/api/treasures?coordinateX=0.0&c

Send

Query

Headers 2

Auth

Body 1

Tests

Query Parameters

☒ coordinateX

0.0

☒ coordinateY

0.0

☒ difficulty

10

☒ before

2022-4-16

☒ start

4

☒ amount

8

☐ parameter

value

Status: 200 OK

Size: 829 Bytes

Time: 122 ms

Response

Headers 4

Cookies

Results

Docs New

1

{

2

{

3

"authorUserName": "obiwan",

4

"coordinateX": -2.0,

5

"coordinateY": 6.5,

6

"dateAdded": "2022-04-15 20:49:24",

7

"difficulty": 10,

8

"hint": "near a tree",

9

"href": "http://localhost:8080/geoetsiinf/api/treasures/7",

10

"size": 100.0,

11

"terrainType": "forest",

12

"treasureId": 7,

13

"treasureName": "tomb"

14

},

15

{

16

"authorUserName": "anakin",

17

"coordinateX": 100.5,

18

"coordinateY": -100.5,

19

"dateAdded": "2022-04-15 20:48:39",

20

"difficulty": 10,

21

"hint": "below a rock",

22

"href": "http://localhost:8080/geoetsiinf/api/treasures/6",

23

"size": 10.0,

24

"terrainType": "mountain",

25

"treasureId": 6,

26

"treasureName": "tomb"

27

},

28

{

29

"authorUserName": "obiwan",

30

"coordinateX": 122.0,

31

"coordinateY": 143.5,

32

"dateAdded": "2022-04-15 20:51:09",

33

"difficulty": 10,

34

"hint": "near a river",

35

"href": "http://localhost:8080/geoetsiinf/api/treasures/8",

36

"size": 12.199999809265137,

37

"terrainType": "mountain",

38

"treasureId": 8,

39

"treasureName": "tesoro"

40

}

41

}

Añadiendo un tesoro encontrado por un usuario particular

PUT

http://localhost:8080/geoetsiinf/api/users/obiwan/foundTreasures/4

Send

Query

Headers 2

Auth

Body 1

Tests

Status: 201 Created

Size: 0 Bytes

Time: 25 ms

Response

Headers 3

Cookies

Results

Docs New

1

|

Copy

Borrando un tesoro añadido por un usuario particular

DELETE [Send](#)

[Query](#) [Headers 2](#) [Auth](#) [Body 1](#) [Tests](#)

Status: **204 No Content** Size: **0 Bytes** Time: **307 ms**

[Response](#) [Headers 2](#) [Cookies](#) [Results](#) [Docs New](#) [{} ☰](#)

1 [Copy](#)

Obteniendo una lista con los tesoros encontrados por un usuario

GET [Send](#)

[Query](#) [Headers 2](#) [Auth](#) [Body 1](#) [Tests](#)

Status: **200 OK** Size: **312 Bytes** Time: **24 ms**

[Response](#) [Headers 4](#) [Cookies](#) [Results](#) [Docs New](#) [{} ☰](#)

```
1   
2 {  
3   "authorUserName": "obiwan",  
4   "coordinateX": 123.02300262451172,  
5   "coordinateY": -92.02300262451172,  
6   "dateAdded": "2022-04-15 20:37:51",  
7   "difficulty": 10,  
8   "hint": "Where the eyes can't see",  
9   "href": "http://localhost:8080/geoetsiinf/api/treasures/4",  
10  "size": 11.5,  
11  "terrainType": "Flat",  
12  "treasureId": 4,  
13  "treasureName": "The force"  
14 }  
15  Copy
```

Añadiendo un amigo

PUT

http://localhost:8080/geoetsiinf/api/users/obiwan/friends/anakin

Send

QueryHeaders 2AuthBody 1Tests

Status: 201 CreatedSize: 0 BytesTime: 31 ms

ResponseHeaders 3CookiesResultsDocs New

1

Obteniendo una lista con los amigos de un usuario

GET

http://localhost:8080/geoetsiinf/api/users/obiwan/friends

Send

QueryHeaders 2AuthBody 1Tests

Status: 200 OKSize: 474 BytesTime: 26 ms

ResponseHeaders 4CookiesResultsDocs New

1
2 {
3 "email": "anakin@jediorder.org",
4 "firstName": "Anakin",
5 "foundTreasures": "http://localhost:8080/geoetsiinf/api/users/anakin
6 /foundTreasures",
7 "friends": "http://localhost:8080/geoetsiinf/api/users/anakin/friends",
8 "href": "http://localhost:8080/geoetsiinf/api/users/anakin",
9 "lastName": "Skywalker",
10 "postalCode": 28017,
11 "profile": "http://localhost:8080/geoetsiinf/api/users/anakin/profile",
12 "treasures": "http://localhost:8080/geoetsiinf/api/users/anakin/treasures",
13 "userName": "anakin"
14 }

Borrando una amistad por parte de un usuario

DELET ▾

http://localhost:8080/geoetsiinf/api/users/obiwan/friends/anakin

Send

Query

Headers ²

Auth

Body ¹

Tests

Status: **204 No Content**

Size: **0 Bytes**

Time: **22 ms**

Response

Headers ²

Cookies

Results

Docs ^{New}

{ } ≡

1 |

Copy

Obteniendo el perfil de un usuario

GET ⌵ http://localhost:8080/geoetsiinf/api/users/obiwan/profile Send

Query Headers 2 Auth Body 1 Tests

Status: **200 OK** Size: **876 Bytes** Time: **42 ms**

Response Headers 4 Cookies Results Docs New {} ≡

Copy

```
1 {
2   "lastFoundTreasures": [
3     {
4       "authorUserName": "obiwan",
5       "coordinateX": 123.02300262451172,
6       "coordinateY": -92.02300262451172,
7       "dateAdded": "2022-04-15 20:37:51",
8       "difficulty": 10,
9       "hint": "Where the eyes can't see",
10      "href": "http://localhost:8080/geoetsiinf/api/treasures/4",
11      "size": 11.5,
12      "terrainType": "Flat",
13      "treasureId": 4,
14      "treasureName": "The force"
15    }
16  ],
17  "numAddedTreasures": 1,
18  "numFoundTreasures": 1,
19  "numFriends": 0,
20  "user": {
21    "email": "nuevoemail@jediorder.org",
22    "firstName": "Obi-Wan",
23    "foundTreasures": "http://localhost:8080/geoetsiinf/api/users/obiwan
24    /foundTreasures",
25    "friends": "http://localhost:8080/geoetsiinf/api/users/obiwan/friends",
26    "href": "http://localhost:8080/geoetsiinf/api/users/obiwan",
27    "lastName": "Kenobi",
28    "postalCode": 12345,
29    "profile": "http://localhost:8080/geoetsiinf/api/users/obiwan/profile",
30    "treasures": "http://localhost:8080/geoetsiinf/api/users/obiwan/treasures",
31    "userName": "obiwan"
32  }
}
```

Cliente REST

Como parte de la práctica hemos programado un cliente Java que proporcionamos como parte de la entrega en formato zip. Este zip contiene un proyecto Maven que se puede importar en Eclipse.

La clase que contiene el método principal para ejecución del cliente es MainMenu.

Al iniciar el cliente, se pregunta al usuario por la URI del punto de entrada a la API. En caso de que no se proporcione, se asume que el servidor se ubica en localhost:8080. Las opciones de edición siguen un patrón similar: si un dato no se proporciona, se mantiene el dato actual indicado en paréntesis.

El cliente utiliza un menú basado en consola que tiene una opción por cada requisito detallado en la especificación de la práctica. Algunas operaciones requieren la entrada de datos por parte del usuario, que se realiza a través de la entrada estándar. En esta sección, proporcionamos capturas del cliente en su menú principal y ejecutando las operaciones de la API:

```
Initial setup OK
--- GeoETSIINF REST Client ---
Please select an option...
0. Exit the program
1. Add new user
2. Get a user
3. Edit a user
4. Delete a user
5. Get all users
6. Add a treasure
7. Edit or delete a treasure
8. Get treasures added by user
9. Register treasure as found
A. Get treasures found by user
B. Add a friend
C. List a user's friends
D. Delete a friend
E. Get treasures closest to a spot
F. Get user's profile
```



```
1
Username:
tommy
First name:
Tommy
Last name:
Test
E-mail:
tt@tt.com
Postal code:
56076
Status code: 201, info: Created
--- GeoETSIINF REST Client ---
Please select an option...
```

```
2
Username:
tommy
User: tommy, Name: Tommy Test, E-mail: tt@tt.com, Postcode: 56076
```

```
3
Username to edit:
tommy
First name (Tommy):
Thomas
Last name (Test):

E-mail (tt@tt.com):

Postal code (56076):

Status code: 200, info: OK
--- GeoETSIINF REST Client ---
Please select an option...
```

```
4
Username to delete:
ana
Status code: 204, info: No Content
```

```
5
Listing all users...
Name Filter (leave blank if none):
an
User: anakin, Name: Anakin Skywalker, E-mail: anakin@jediorder.org, Postcode: 28017
User: anthony, Name: Anthony Anthony, E-mail: none, Postcode: 59696
--- GeoETSIINF REST Client ---
Please select an option...
```

```
6
Author:
tommy
Treasure name:
Chocolate
X coordinate:
55
Y coordinate:
67
Terrain type:
Desert
Difficulty:
8
Size:
3.3
Hint:
On the dune
Status code: 201, info: Created
--- GeoETSIINF REST Client ---
Please select an option...
```

```
7
Login (treasure's owner):
tommy
Treasure ID to modify (to delete, type - before it, e.g. -125 instead of 125):
10
Editing treasure...
Treasure name (Chocolate):
Milk chocolate
X coordinate (55.0):

Y coordinate (67.0):

Terrain type (Desert):

Difficulty out of 10 (8):

Size (3.3):

Hint (On the dune):

Status code: 200, info: OK
--- GeoETSIINF REST Client ---
Please select an option...
```

```
8
Author:
tommy
Treasure [authorUserName=tommy, coordinateX=12.0, coordinateY=13.0, difficulty=6,
Treasure [authorUserName=tommy, coordinateX=66.0, coordinateY=66.0, difficulty=6,
--- GeoETSIINF REST Client ---
Please select an option...
```

```
9|
Finder's username:
tommy
Treasure ID:
8
8
Status code: 201, info: Created
--- GeoETSIINF REST Client ---
Please select an option...
```

```
A
Author:
tommy
Treasure [authorUserName=tommy, coordinateX=12.0, coordinateY=13.0, difficulty=6, hint=Up, size=5.5,
--- GeoETSIINF REST Client ---
```

```
B
Login (user to add friend to):|
tommy
Friend to add:
anakin
Status code: 201, info: Created
--- GeoETSIINF REST Client ---
Please select an option...
```

```
C
Login (user to get friends of):
tommy
User: anakin, Name: Anakin Skywalker, E-mail: anakin@jediorder.org, Postcode: 28017
--- GeoETSIINF REST Client ---
Please select an option...
```

```
D
Login (username to delete friend from):
tommy
Friend to delete:
anakin
Status code: 204, info: No Content
--- GeoETSIINF REST Client ---
Please select an option...
```

```
F
Username to obtain profile from:
tommy
User: tommy, Name: Thomas Test, E-mail: tt@tt.com, Postcode: 56076
Total found treasures: 1
Number of friends: 0
Number of added treasures: 2
Last found treasures:
Treasure [authorUserName=tommy, coordinateX=12.0, coordinateY=13.0,
--- GeoETSIINF REST Client ---
Please select an option...
```

Conclusiones

Hemos conseguido realizar consultas y operaciones sobre la base de datos de GeoETSIINF mediante métodos HTTP.. Con esta práctica como ejemplo, entendemos mejor la comunicación entre cliente y servidor REST.

El mayor desafío que hemos encontrado durante la práctica ha sido la depuración de errores del servidor, que recalca la importancia de propagar correctamente la información contenida en las excepciones de Java.

Separar el código del servidor en modelo, recursos y DAO (interfaz con la base de datos MySQL) nos permite, mediante excepciones específicas, averiguar si los errores se producen al procesar el cuerpo de petición o al lanzar consultas MySQL.

Una posible expansión de esta práctica sería sustituir el cliente de línea de comandos por un cliente Android, algo ya presente en la práctica de la asignatura “Programming for Mobile Devices”.

En resumen, durante la elaboración de esta práctica nos hemos familiarizado con métodos y herramientas para el diseño e implementación de una API REST, tecnología clave en el ámbito de las aplicaciones Web interactivas.