

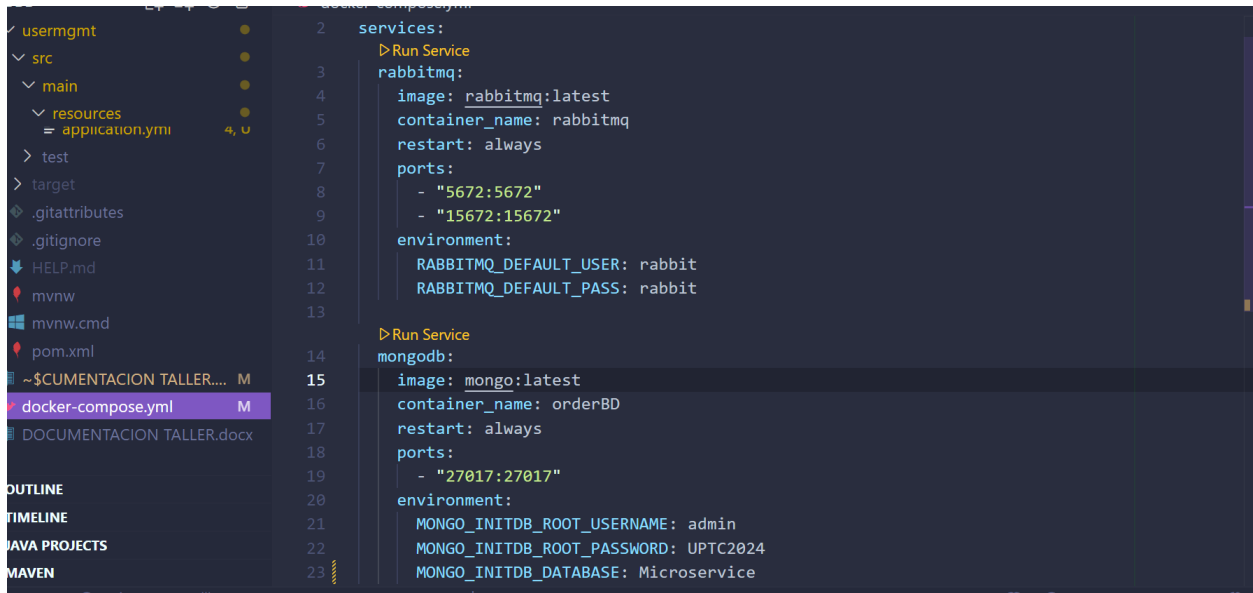
DOCUMENTACION TALLER

LUIS HERNANDEZ

JULIAN BAYONA

DEVID CRUZ

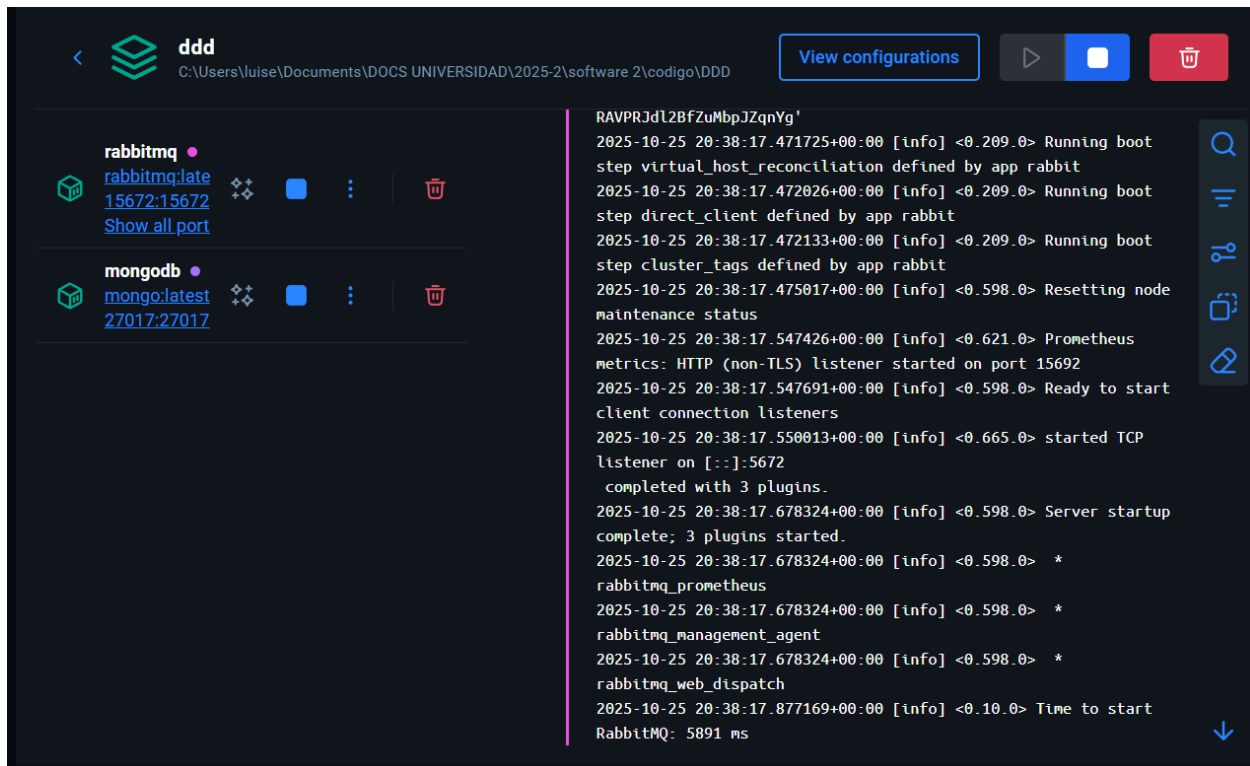
LINK REPO <https://github.com/luisDeveloper2002/DDD.git>



The screenshot shows an IDE with a file explorer on the left and a code editor on the right. The file explorer lists files including `usermgmt`, `src`, `main`, `resources`, `application.yml`, `test`, `target`, `.gitattributes`, `.gitignore`, `HELP.md`, `mvnw`, `mvnw.cmd`, `pom.xml`, `~$DOCUMENTACION TALLER.... M`, `docker-compose.yml`, and `DOCUMENTACION TALLER.docx`. The code editor displays the `docker-compose.yml` file with the following content:

```
services:
  rabbitmq:
    image: rabbitmq:latest
    container_name: rabbitmq
    restart: always
    ports:
      - "5672:5672"
      - "15672:15672"
    environment:
      RABBITMQ_DEFAULT_USER: rabbit
      RABBITMQ_DEFAULT_PASS: rabbit
  mongodb:
    image: mongo:latest
    container_name: orderBD
    restart: always
    ports:
      - "27017:27017"
    environment:
      MONGO_INITDB_ROOT_USERNAME: admin
      MONGO_INITDB_ROOT_PASSWORD: UPTC2024
      MONGO_INITDB_DATABASE: Microservice
```

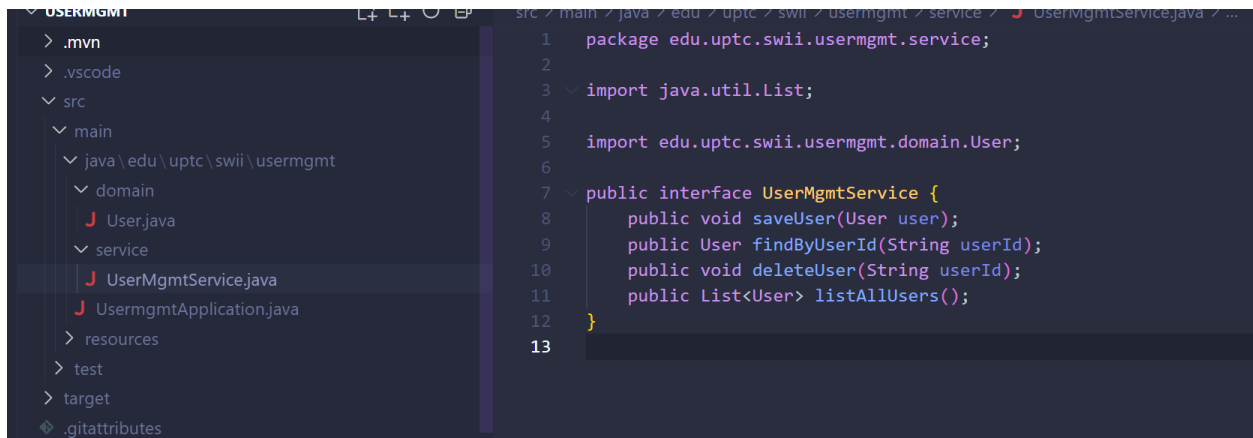
Validad que al hacer el compose up esten bien los contenedores



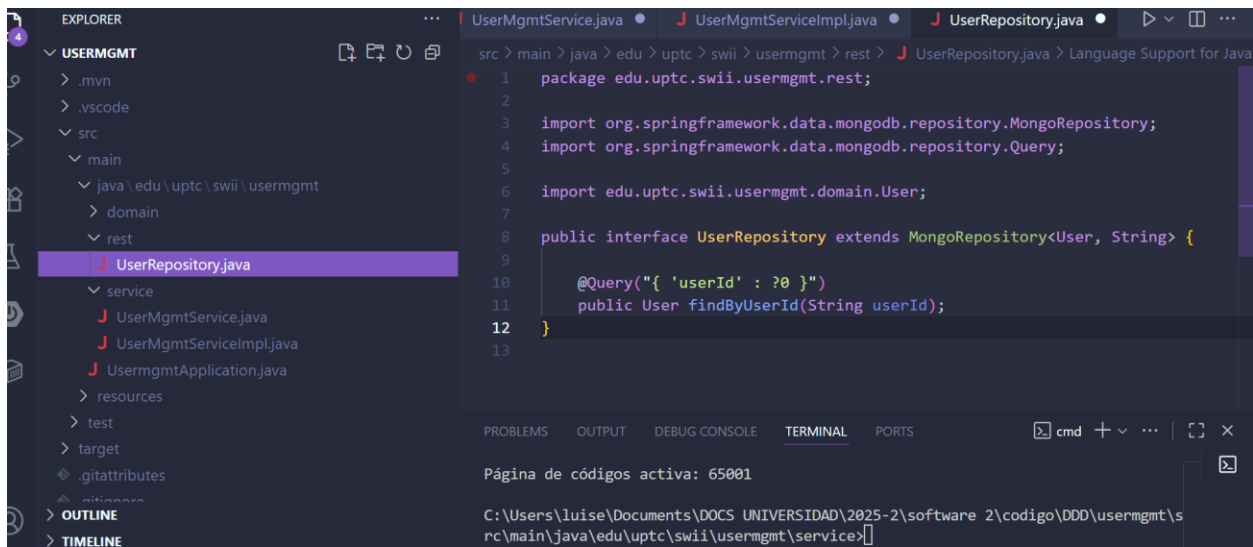
Se crea la clase users en domain



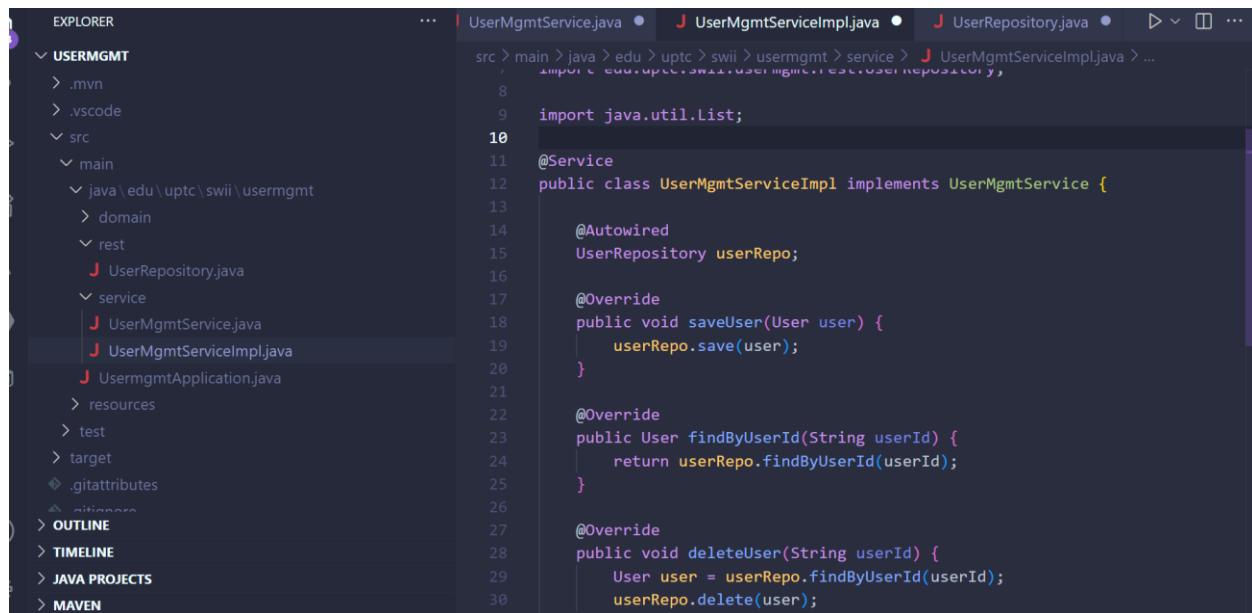
Se crea el service



Se crea el repository



Se crea el service implement



The screenshot shows the VS Code editor with the Explorer sidebar on the left. The Explorer shows a project structure for 'USERMGMT' with a 'rest' directory containing 'UserRepository.java' and a 'service' directory containing 'UserMgmtService.java', 'UserMgmtServiceImpl.java', and 'UsermgmtApplication.java'. The main editor displays the code for 'UserMgmtServiceImpl.java'. The code is as follows:

```
src > main > java > edu > uptc > swii > usermgmt > service > J UserMgmtServiceImpl.java > ...
import edu.uptc.swii.usermgmt.service.UserMgmtService;
import edu.uptc.swii.usermgmt.domain.User;
import java.util.List;

@Service
public class UserMgmtServiceImpl implements UserMgmtService {

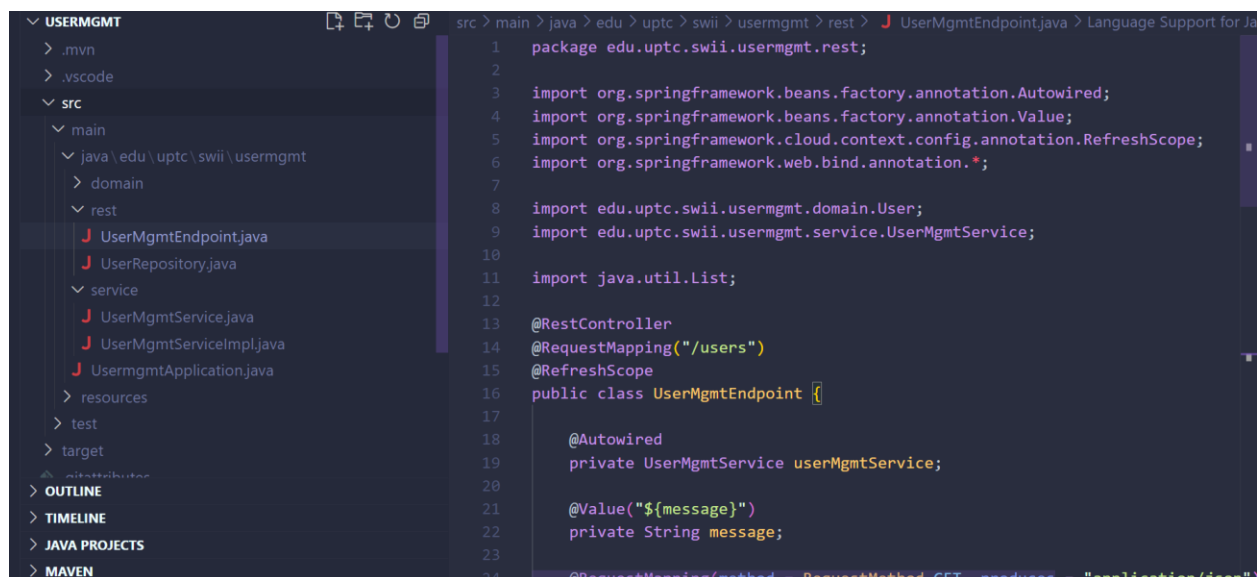
    @Autowired
    UserRepository userRepo;

    @Override
    public void saveUser(User user) {
        userRepo.save(user);
    }

    @Override
    public User findById(String userId) {
        return userRepo.findById(userId);
    }

    @Override
    public void deleteUser(String userId) {
        User user = userRepo.findById(userId);
        userRepo.delete(user);
    }
}
```

Se crea el endpoint



The screenshot shows the VS Code editor with the Explorer sidebar on the left. The Explorer shows the same project structure as the previous screenshot, but now with 'UserMgmtEndpoint.java' added to the 'rest' directory. The main editor displays the code for 'UserMgmtEndpoint.java'. The code is as follows:

```
src > main > java > edu > uptc > swii > usermgmt > rest > J UserMgmtEndpoint.java > Language Support for Java
package edu.uptc.swii.usermgmt.rest;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.cloud.context.config.annotation.RefreshScope;
import org.springframework.web.bind.annotation.*;

import edu.uptc.swii.usermgmt.domain.User;
import edu.uptc.swii.usermgmt.service.UserMgmtService;

import java.util.List;

@RestController
@RequestMapping("/users")
@RefreshScope
public class UserMgmtEndpoint {

    @Autowired
    private UserMgmtService userMgmtService;

    @Value("${message}")
    private String message;

    @RequestMapping(method = RequestMethod.GET, produces = "application/json")
}
```

Se adiciona estas dos notaciones en el aplicacion

The screenshot shows the VS Code interface with the project structure on the left and the `UsermgmtApplication.java` file open in the editor. The project structure includes:

- USERMGMT
  - .mvn
  - .vscode
  - src
    - main
      - java\edu\uptc\swii\usermgmt
        - domain
        - rest
          - UserMgmtEndpoint.java
          - UserRepository.java
        - service
          - UserMgmtService.java
          - UserMgmtServiceImpl.java
        - UsermgmtApplication.java
      - resources
      - test
      - target

The `UsermgmtApplication.java` file contains the following code:

```
src > main > java > edu > uptc > swii > usermgmt > J UsermgmtApplication.java > ...
1 package edu.uptc.swii.usermgmt;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5 import org.springframework.cloud.client.discovery.EnableDiscoveryClient;
6 import org.springframework.context.annotation.ComponentScan;
7
8 @EnableDiscoveryClient
9 @SpringBootApplication
10 @ComponentScan(basePackages = {"edu.uptc.swii.usermgmt"})
11 edu.uptc.swii.usermgmt.UsermgmtApplication
12 public class UsermgmtApplication {
13
14     Run|Debug
15     public static void main(String[] args) {
16         SpringApplication.run(UsermgmtApplication.class, args);
17     }
18 }
19
```

The screenshot shows the VS Code interface with the project structure on the left and the `application.properties` file open in the editor. The project structure is the same as the previous screenshot. The `application.properties` file contains the following code:

```
usermgmt > src > main > resources > application.properties
1 message=Bienvenido al servicio de usuarios
2 spring.application.name=usermgmt
3 spring.data.mongodb.uri=mongodb://admin:UPTC2024@localhost:27017/microservice?authSource=admin
4
5
6
```

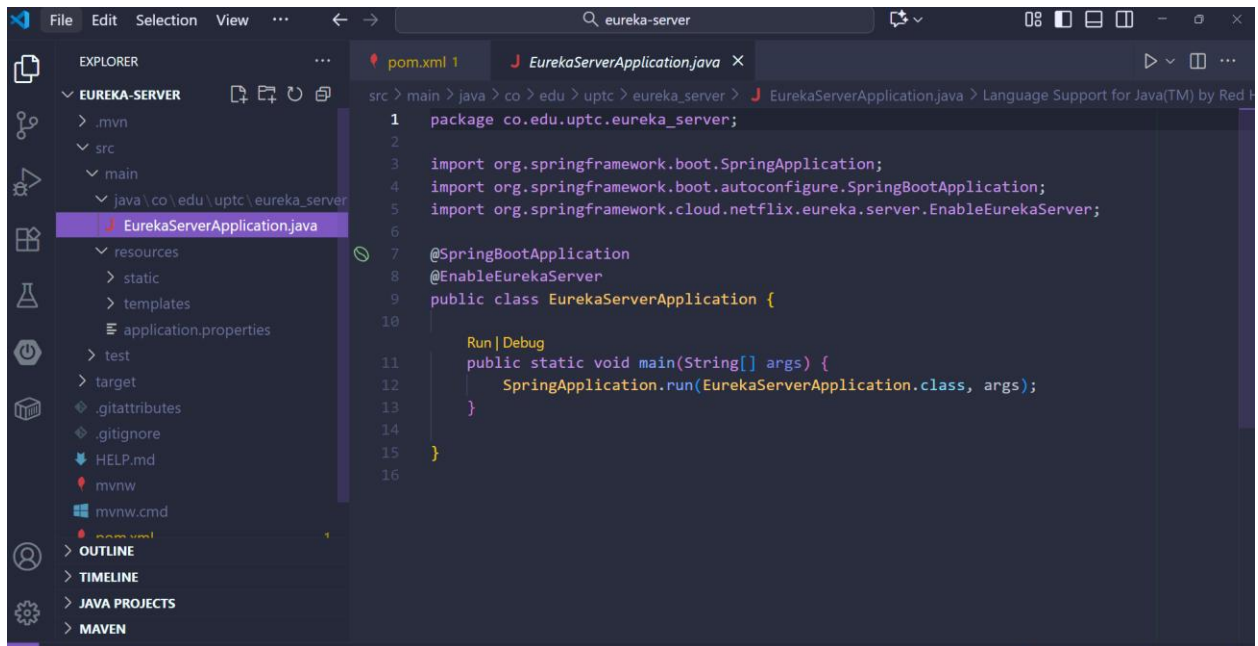
The screenshot shows the VS Code interface with the project structure on the left and the `application.yml` file open in the editor. The project structure is the same as the previous screenshots. The `application.yml` file contains the following code:

```
usermgmt > src > main > resources > application.yml
1
2 spring:
3   cloud:
4     config:
5       name: microservices-config
6       uri: http://localhost:8888/
7       username: config_user
8       password: config_user
9   rabbitmq:
10     host: localhost
11     port: 5672
12     virtual-host: /
13     username: guest
14     password: guest
15   profiles:
16     active: DEV
17   data:
18     mongodb:
19       database: Microservice
20       host: localhost
21       port: 27017
22       repositories:
23         type: auto
24       uri: mongodb://admin:UPTC2024@localhost:27017/Microservice?authSource=admin
25
26
```

Desarrollo TALLER

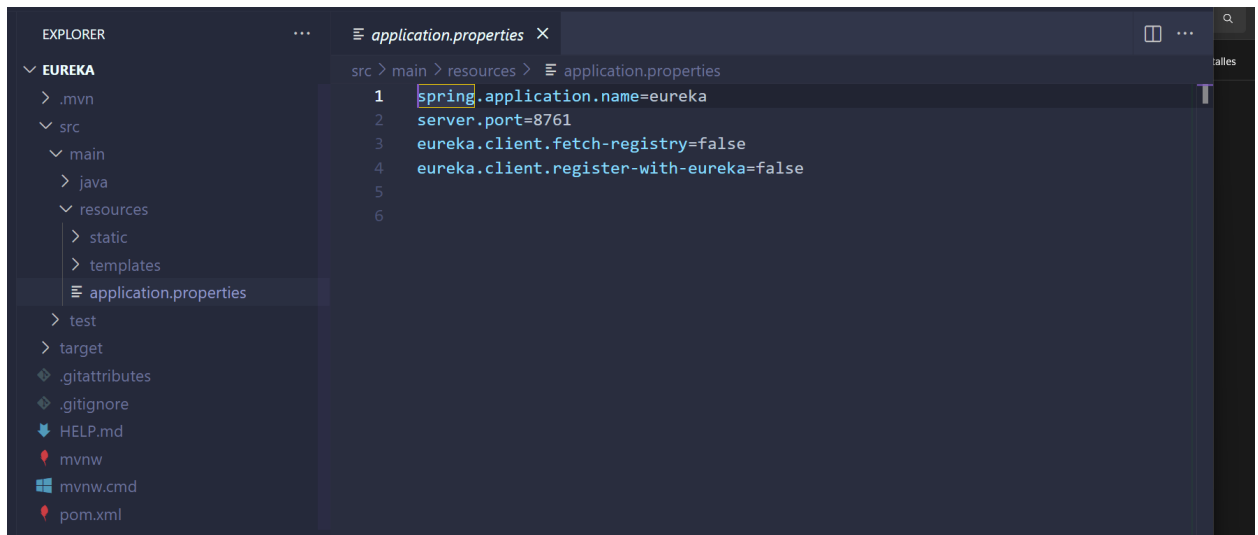
1PUNTO

Inicie un servidor de Eureka como se observó en pasados talleres



The screenshot shows an IDE window titled "eureka-server". The Explorer panel on the left shows the project structure: EUREKA-SERVER > src > main > java > co > edu > uptc > eureka\_server > EurekaServerApplication.java. The main editor displays the code for EurekaServerApplication.java. The code includes imports for SpringApplication, SpringBootApplication, and EnableEurekaServer. It defines a public class EurekaServerApplication with a main method.

```
1 package co.edu.uptc.eureka_server;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5 import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;
6
7 @SpringBootApplication
8 @EnableEurekaServer
9 public class EurekaServerApplication {
10
11     Run | Debug
12     public static void main(String[] args) {
13         SpringApplication.run(EurekaServerApplication.class, args);
14     }
15 }
16
```



The screenshot shows the same IDE window with the Explorer panel showing the project structure: EUREKA > src > main > resources > application.properties. The main editor displays the content of the application.properties file, which contains configuration properties for the Eureka server.

```
1 spring.application.name=eureka
2 server.port=8761
3 eureka.client.fetch-registry=false
4 eureka.client.register-with-eureka=false
5
6
```

spring Eureka

HOME LAST 1000 SINCE STARTUP

System Status

Environment	test	Current time	2025-10-25T17:58:03 -0500
Data center	default	Uptime	00:00
		Lease expiration enabled	false
		Renews threshold	3
		Renews (last min)	0

DS Replicas

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
USERMGMT	n/a (1)	(1)	UP (1) - localhost:usermgmt

General Info

Name	Value
total-avail-memory	88mb

## 2 PUNTO

Cree la base de datos denominada “microservice” y las tablas correspondientes en MongoDB

My Queries

Data Modeling

CONNECTIONS (1)

Search connections

localhost:27017

Microservice

appusers

admin

config

local

localhost:27017 > Microservice > appusers

Documents 1 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

Explain Reset Find </> Options

ADD DATA EXPORT DATA UPDATE DELETE

25 1 - 1 of 1

```
{
  "_id": "12346"
  "userId": "12346"
  "firstName": "Andres"
  "lastName": "Sierra"
  "age": 25
  "_class": "edu.uptc.swii.usermgmt.domain.User"
}
```

## 3 PUNTO

CON ESTAS CLASES EL MICROSERVICIO PERSISTE EN MONGO DB

```
DDD
├── usermgmt
│   ├── src
│   │   ├── main
│   │   │   ├── java \ edu \ uptc \ swii \ userm...
│   │   │   │   ├── domain
│   │   │   │   │   ├── rest
│   │   │   │   │   │   ├── UserMgmtEndpoint.java
│   │   │   │   │   │   ├── UserRepository.java
│   │   │   │   │   │   ├── service
│   │   │   │   │   │   │   ├── UserMgmtService.java
│   │   │   │   │   │   │   ├── UserMgmtServiceImpl.java
│   │   │   │   │   │   │   ├── UsermgmtApplication.java
│   │   │   │   │   │   │   ├── resources
│   │   │   │   │   │   │   ├── static
│   │   │   │   │   │   │   ├── templates
│   │   │   │   │   │   │   └── application.properti... 1, M
│   │   │   │   │   │   └── OUTLINE
│   │   │   │   │   │   └── TIMELINE
│   │   │   │   │   └── JAVA PROJECTS
│   │   │   └── ...
│   │   └── ...
│   └── ...
└── ...
```

```
12
13 }RestController
14 @RequestMapping("/users")
15 @RefreshScope
16 public class UserMgmtEndpoint {
17
18     @Autowired
19     private UserMgmtService userMgmtService;
20
21     @Value("${message}")
22     private String message;
23
24     @RequestMapping(method = RequestMethod.GET, produces = "application/json")
25     public String welcome() {
26         return message;
27     }
28
29     @RequestMapping(value = "/{userId}", method = RequestMethod.GET, produces = "application/json")
30     public User findUserById(@PathVariable("userId") String userId) {
31         User user = userMgmtService.findUserById(userId);
32         return user;
33     }
34 }
```

```
DDD
├── usermgmt
│   ├── src
│   │   ├── main
│   │   │   ├── java \ edu \ uptc \ swii \ userm...
│   │   │   │   ├── domain
│   │   │   │   │   ├── rest
│   │   │   │   │   │   ├── UserMgmtEndpoint.java
│   │   │   │   │   │   ├── UserRepository.java
│   │   │   │   │   │   ├── service
│   │   │   │   │   │   │   ├── UserMgmtService.java
│   │   │   │   │   │   │   ├── UserMgmtServiceImpl.java
│   │   │   │   │   │   │   ├── UsermgmtApplication.java
│   │   │   │   │   │   │   ├── resources
│   │   │   │   │   │   │   ├── static
│   │   │   │   │   │   │   ├── templates
│   │   │   │   │   │   │   └── application.properti... 1, M
│   │   │   │   │   │   └── OUTLINE
│   │   │   │   │   │   └── TIMELINE
│   │   │   │   └── JAVA PROJECTS
│   │   └── ...
│   └── ...
└── ...
```

```
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.stereotype.Service;
5
6 import edu.uptc.swii.usermgmt.domain.User;
7 import edu.uptc.swii.usermgmt.rest.UserRepository;
8
9 import java.util.List;
10
11 @Service
12 public class UserMgmtServiceImpl implements UserMgmtService {
13
14     @Autowired
15     UserRepository userRepo;
16
17     @Override
18     public void saveUser(User user) {
19         userRepo.save(user);
20     }
21
22     @Override
23     public User findById(String userId) {
24         return userRepo.findById(userId);
25     }
26 }
```

```
EXPLORE
├── DDD
│   ├── usermgmt
│   │   ├── src
│   │   │   ├── main
│   │   │   │   ├── java \ edu \ uptc \ swii \ userm...
│   │   │   │   │   ├── domain
│   │   │   │   │   │   ├── rest
│   │   │   │   │   │   │   ├── UserMgmtEndpoint.java
│   │   │   │   │   │   │   ├── UserRepository.java
│   │   │   │   │   │   │   ├── service
│   │   │   │   │   │   │   │   ├── UserMgmtService.java
│   │   │   │   │   │   │   │   ├── UserMgmtServiceImpl.java
│   │   │   │   │   │   │   │   ├── UsermgmtApplication.java
│   │   │   │   │   │   │   │   ├── resources
│   │   │   │   │   │   │   │   ├── static
│   │   │   │   │   │   │   │   ├── templates
│   │   │   │   │   │   │   └── application.properti... 1, M
│   │   │   │   │   │   └── OUTLINE
│   │   │   │   │   │   └── TIMELINE
│   │   │   │   └── JAVA PROJECTS
│   │   └── ...
│   └── ...
└── ...
```

```
1 package edu.uptc.swii.usermgmt.rest;
2
3 import org.springframework.data.mongodb.repository.MongoRepository;
4 import org.springframework.data.mongodb.repository.Query;
5
6 import edu.uptc.swii.usermgmt.domain.User;
7
8 public interface UserRepository extends MongoRepository<User, String> {
9
10     @Query("{ 'userId' : ?0 }")
11     public User findById(String userId);
12 }
13 }
```



#### 4 PUNTO

Valide la configuración de RabbitMq para que funcione bajo el puerto 5672.

Con esta configuración se logra que el RabbitMq funcione bajo el puerto 5672

Run Service

```
rabbitmq:
  image: rabbitmq:latest
  container_name: rabbitmq
  restart: always
  ports:
    - "5672:5672"
    - "15672:15672"
  environment:
    RABBITMQ_DEFAULT_USER: rabbit
    RABBITMQ_DEFAULT_PASS: rabbit
```

```
0  rabbitmq:
1      host: localhost
2      port: 5672
3      virtual-host: /
4      username: guest
5      password: guest
```

```

C:\Users\luise>netstat | findstr 5672
^C^C
C:\Users\luise>netstat -ano | findstr 5672
TCP    0.0.0.0:5672          0.0.0.0:0             LISTENING        11248
TCP    0.0.0.0:15672         0.0.0.0:0             LISTENING        11248
TCP    [::]:5672            [::]:0                LISTENING        11248
TCP    [::]:15672           [::]:0                LISTENING        11248
TCP    [::1]:5672           [::]:0                LISTENING        14324
TCP    [::1]:15672          [::]:0                LISTENING        14324

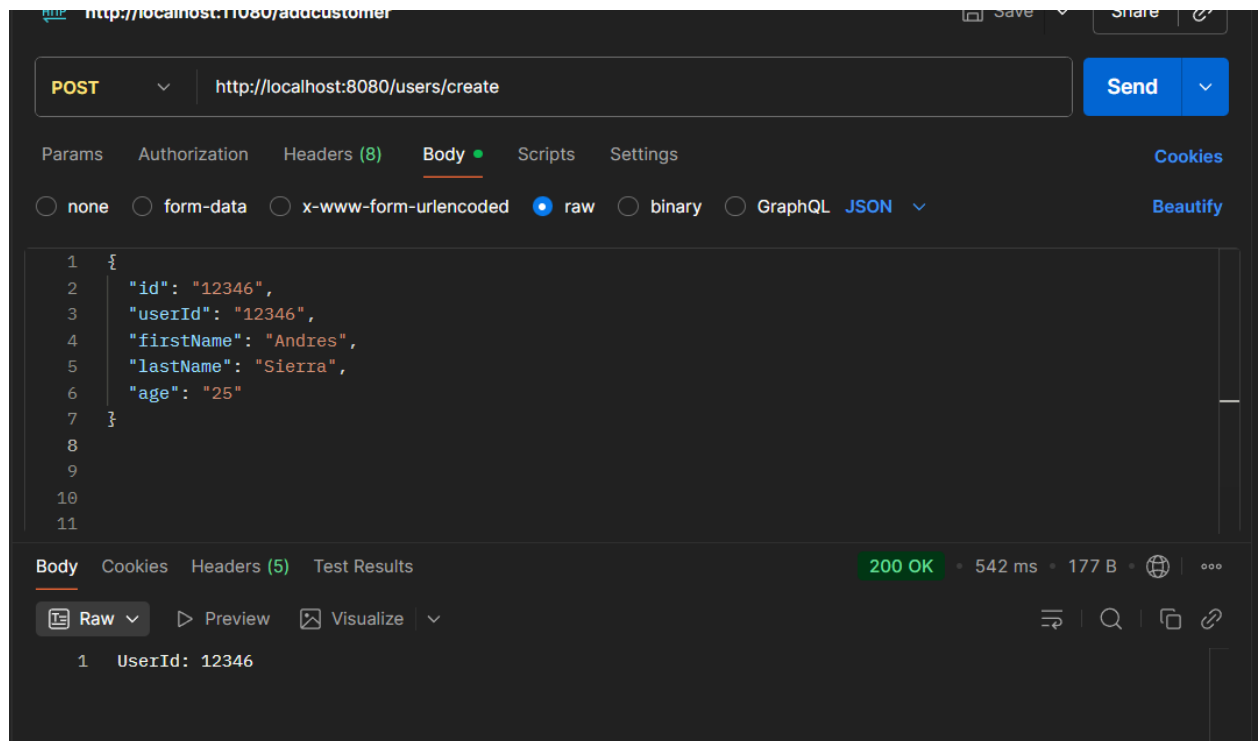
C:\Users\luise>tasklist | findstr 14324
wslrelay.exe             14324 Console             10      2.768 KB


C:\Users\luise>tasklist | findstr 11248
com.docker.backend.exe    11248 Console             10     126.700 KB



```


## 5 PUNTO


Pruebe que la aplicación trabaja correctamente por medio de postman a los endpoints expuestos por la aplicación.




 **http://localhost:11080/addcustomer**

 Save 

Share 

GET 

http://localhost:8080/users

Send 

Params

Authorization

Headers (6)

**Body**

Scripts

Settings

Cookies

☒ none

☐ form-data

☐ x-www-form-urlencoded

☐ raw

☐ binary

☐ GraphQL

This request does not have a body

**Body**

Cookies

Headers (5)

Test Results

200 OK • 30 ms • 190 B •  

 JSON 

 Preview

 Visualize 

1

Bienvenido al servicio de usuarios

HTTP

http://localhost:11080/addcustomer

Save

Share

GET

http://localhost:8080/users/12346

Send

ParamsAuthorizationHeaders (6)BodyScriptsSettingsCookies

☒ none

☐ form-data

☐ x-www-form-urlencoded

☐ raw

☐ binary

☐ GraphQL

This request does not have a body

BodyCookiesHeaders (5)Test Results

200 OK • 258 ms • 245 B •

{}

JSON

Preview

Visualize

1

{

2

"id": "12346",

3

"userId": "12346",

4

"firstName": "Andres",

5

"lastName": "Sierra",

6

"age": 25

7

}

HTTP <http://localhost:11080/addcustomer> Save Share

**GET** <http://localhost:8080/users/listAll> Send

Params Authorization Headers (6) **Body** Scripts Settings Cookies

☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

This request does not have a body

**Body** Cookies Headers (5) Test Results 200 OK 11 ms 325 B 🌐 ⋮

{} **JSON** ▶ Preview 🖼️ Visualize ⌵ ↔️ ≡ 🔍 📄 🔗

```
2 {
3   "id": "12346",
4   "userId": "12346",
5   "firstName": "Andres",
6   "lastName": "Sierra",
7   "age": 25
8 },
9 {
10  "id": "12348",
11  "userId": "12348",
12  "firstName": "PPPP",
13  "lastName": "XXXX",
```

🔗 My Queries 🔗 Data Modeling

**CONNECTIONS (1)** ✕ + ⋮

Search connections 🔍

localhost:27017 📁 Microservice 📁 **appusers** ⋮

admin 📁 config 📁 local

localhost:27017 > Microservice > appusers 🔗 Open MongoDB shell ⛶

**Documents** 1 Aggregations Schema Indexes 1 Validation

🔔 Type a query: { field: 'value' } or [Generate query](#) ⚡ 🔍 Explain 🔄 Reset 🔍 Find ⌵ 🔗 Options ▶

➕ ADD DATA 📄 EXPORT DATA ✎ UPDATE 🗑️ DELETE 25 1 - 2 of 2 🔄 ⏪ ⏩ ☰ 🔍 📄 🔗

```
{
  "_id": "12346",
  "userId": "12346",
  "firstName": "Andres",
  "lastName": "Sierra",
  "age": 25,
  "_class": "edu.uptc.swii.usermgmt.domain.User"
}
```

```
{
  "_id": "12348",
  "userId": "12348",
  "firstName": "PPPP",
  "lastName": "XXXX",
  "age": 12,
  "_class": "edu.uptc.swii.usermgmt.domain.User"
}
```

