

Índice

Índice	1
API para cambio de divisas	2
Descripción de la iniciativa	2
Descripción de la solución	2
Modelo de la solución	3
Diagrama de despliegue	4
Evidencias	5
Lenguaje de programación JAVA	5
Framework Spring Boot	5
Programación reactiva RxJava	6
Base de datos MySQL	6
Dockerización de la base de datos y la API	7
Imagen base para la dockerización Ubuntu 20.04	7
Seguridad de la API usando JWT	8
Consumo de la API mediante Postman	8
Obtención del token	8
Creación de divisa	9
Actualización de divisa (value)	9
Actualización de divisa (name)	10
Obtención de una divisa por ID	10
Obtención de todas las divisas registradas	11
Eliminación de una divisa	11
Ejecución de una solicitud (800 EUR a USD)	12
Uso de las ramas en Github	12
Flujo de pull requests realizados	13

API para cambio de divisas

Descripción de la iniciativa

Disponer de un servicio para el cambio entre las diferentes divisas, manteniendo el registro de las solicitudes hechas por cada persona autenticada. Permitir el mantenimiento de los diferentes tipos de divisas (creación, actualización, obtención y eliminación). Y registrar los campos de auditoría para el mantenimiento de las divisas y las solicitudes de tipo de cambio.

Descripción de la solución

Framework y lenguaje: La implementación de la solución será usando el framework Spring Boot y el lenguaje de programación JAVA. Se aplicarán los principios de la programación reactiva RxJava.

Base de datos: La solución contará con una base de datos MySQL donde se almacenarán datos como las divisas, usuarios y solicitudes. Para los Id's de las divisas se hará uso del estándar ISO 4217.

Dockerización: Se generarán 2 imágenes Docker, una para la base de datos y la otra para la API, con base en el SO Ubuntu 20.04.

Seguridad: Se usará JWT para la seguridad de la API, el cual solicitará un usuario y contraseña, y tras validarlos con la base de datos, devolverá un token. Dicho token será la llave para permitir el acceso al uso de la API. El token tendrá una configuración, cuya vigencia será de 1 minuto.

Modelo de la solución

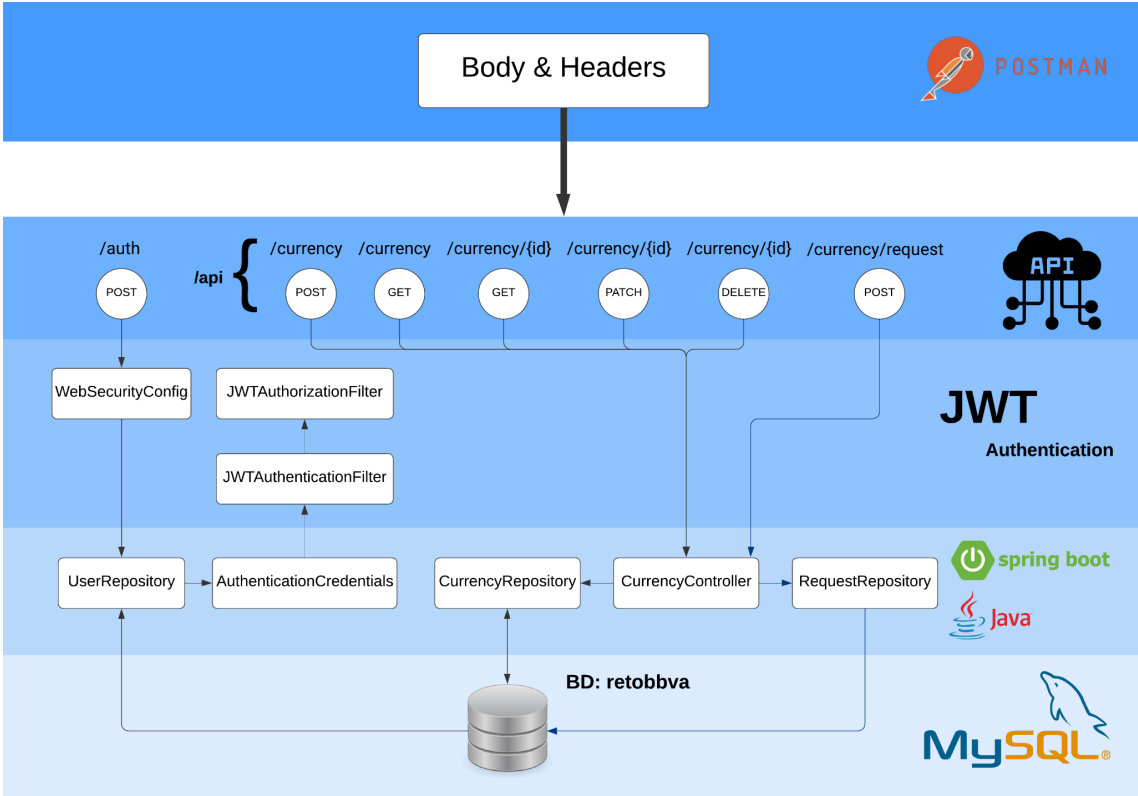
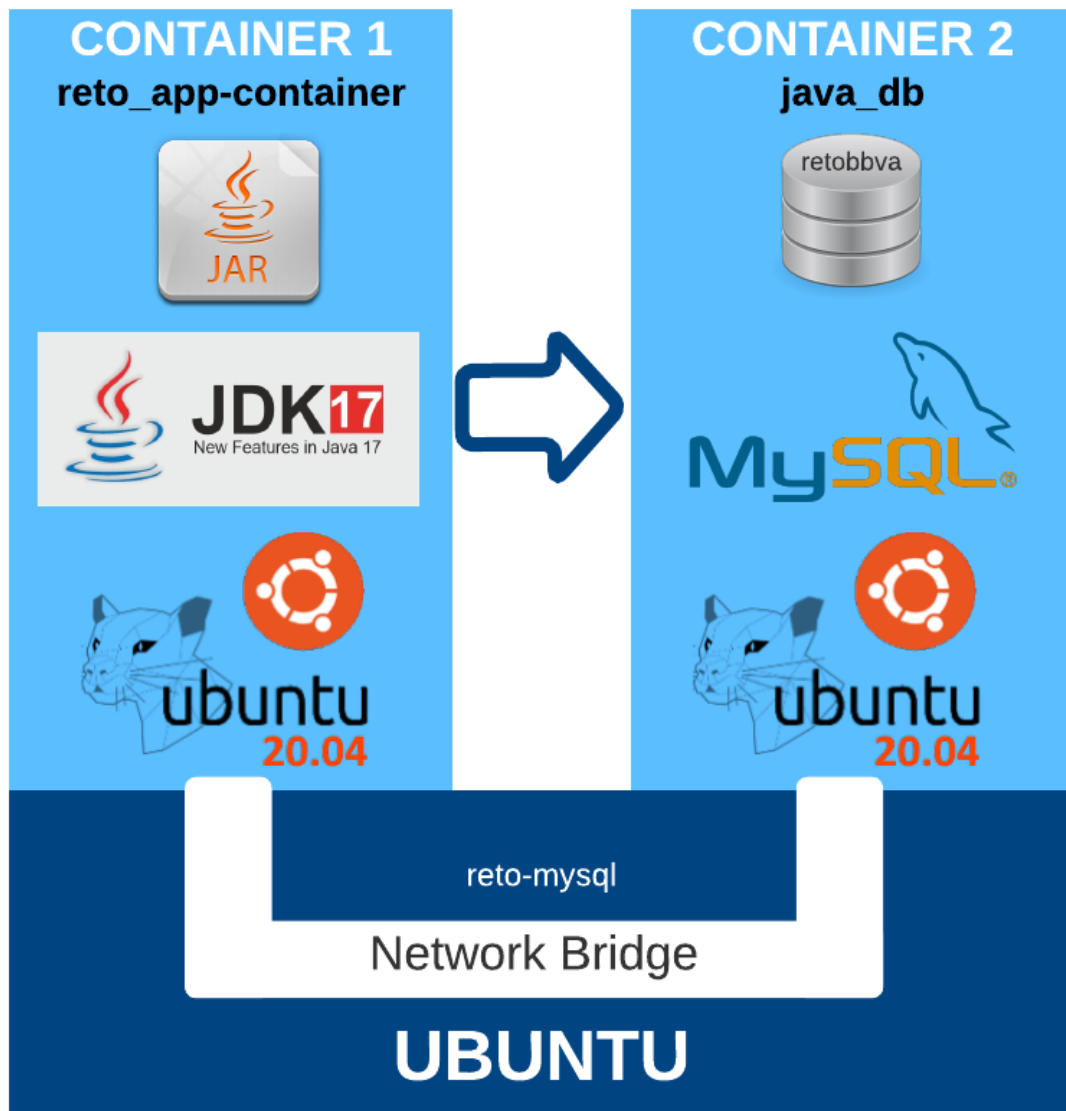
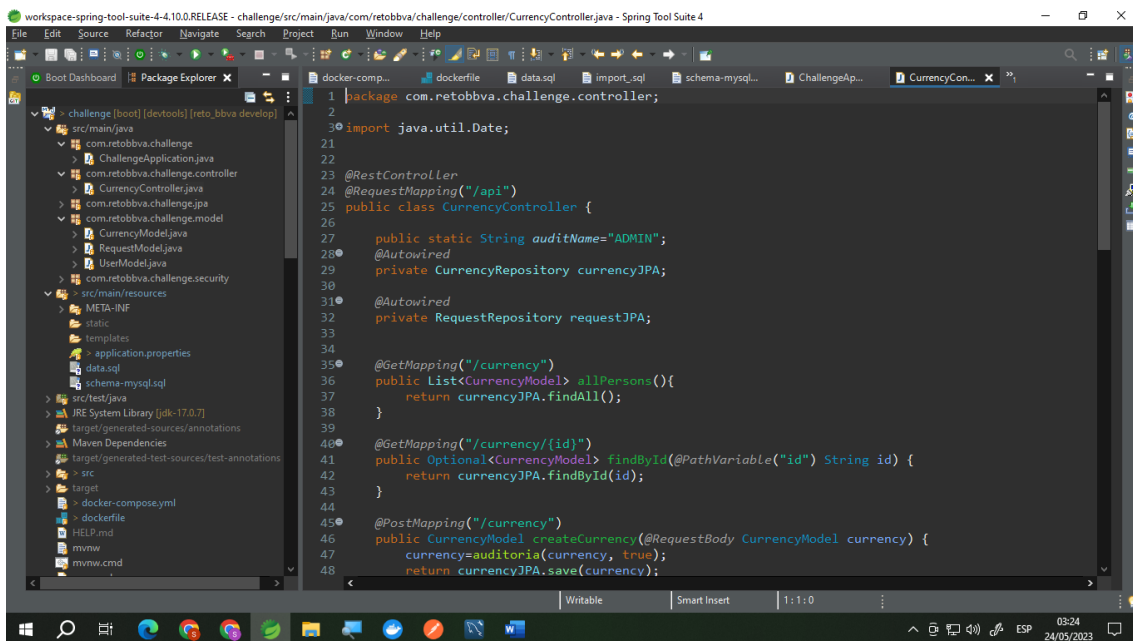


Diagrama de despliegue



Evidencias

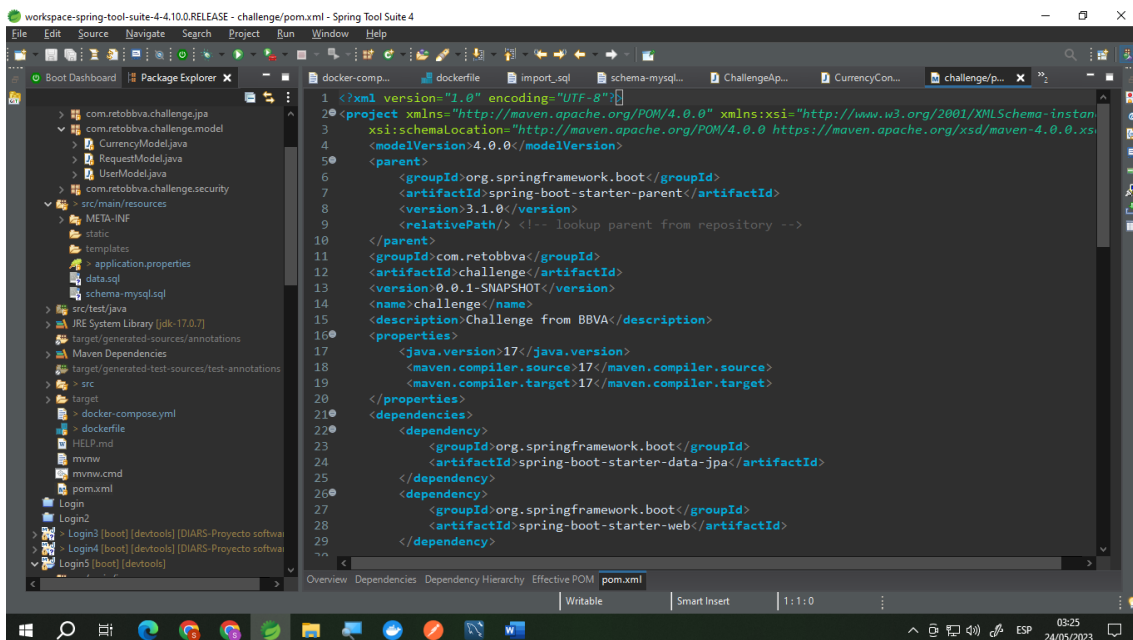
Lenguaje de programación JAVA



The screenshot shows the Spring Tool Suite 4 IDE with the 'CurrencyController.java' file open. The left sidebar displays the project structure, including the 'com.retobbva.challenge' package and its sub-packages. The main editor area shows the following Java code:

```
1 package com.retobbva.challenge.controller;
2
3 import java.util.Date;
4
5 @RestController
6 @RequestMapping("/api")
7 public class CurrencyController {
8
9     public static String auditName="ADMIN";
10    @Autowired
11    private CurrencyRepository currencyJPA;
12
13    @Autowired
14    private RequestRepository requestJPA;
15
16    @GetMapping("/currency")
17    public List<CurrencyModel> allPersons(){
18        return currencyJPA.findAll();
19    }
20
21    @GetMapping("/currency/{id}")
22    public Optional<CurrencyModel> findById(@PathVariable("id") String id) {
23        return currencyJPA.findById(id);
24    }
25
26    @PostMapping("/currency")
27    public CurrencyModel createCurrency(@RequestBody CurrencyModel currency) {
28        currency=auditoria(currency, true);
29        return currencyJPA.save(currency);
30    }
31}
```

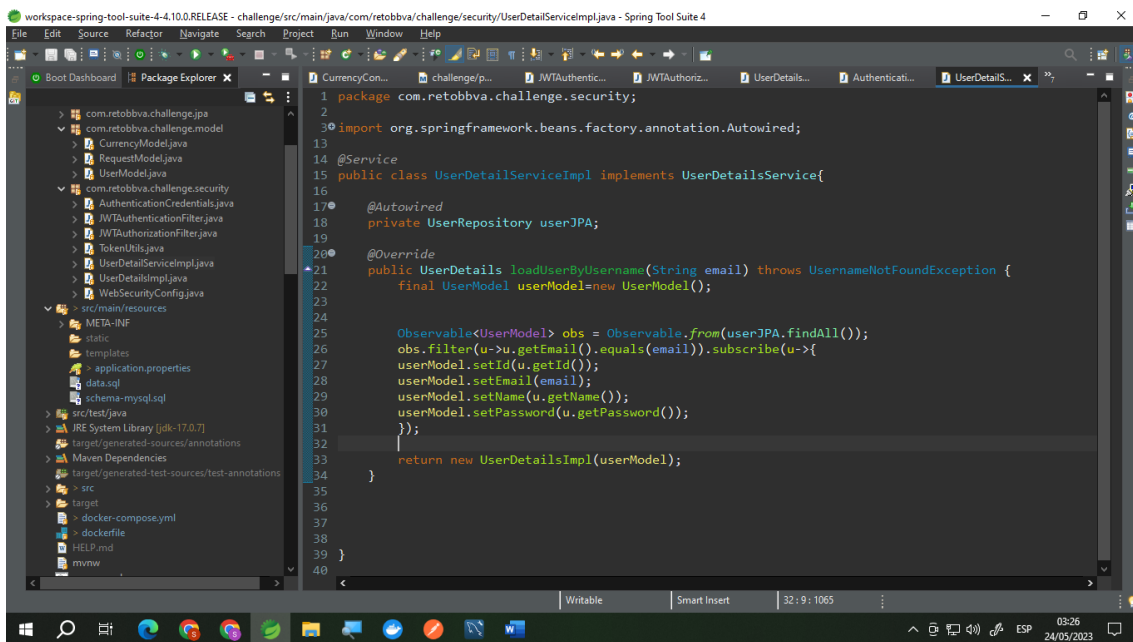
Framework Spring Boot



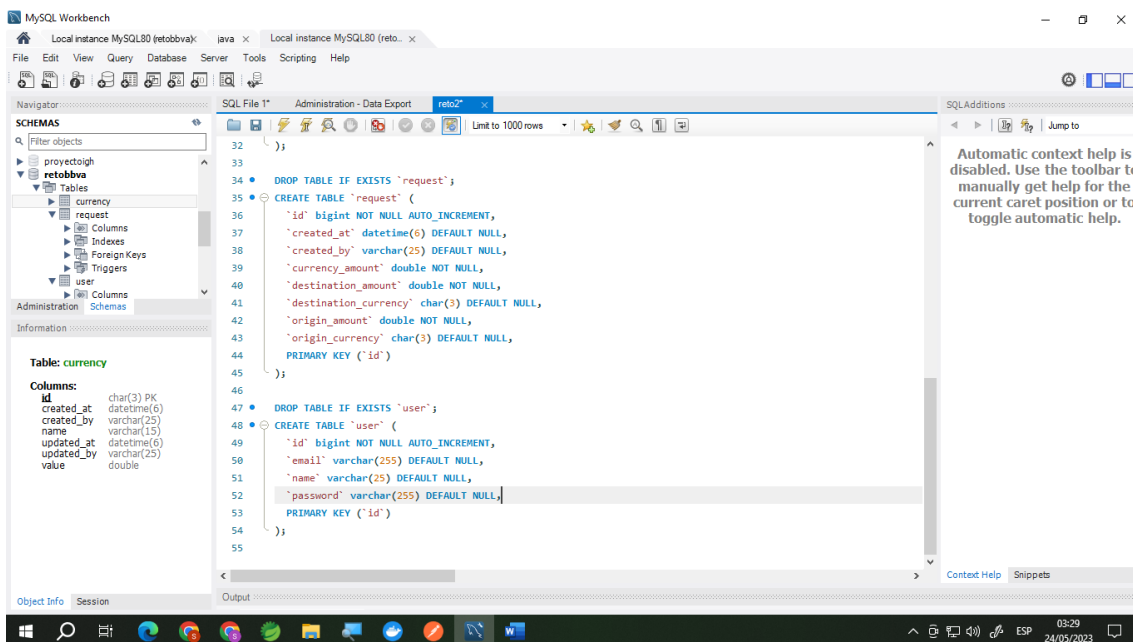
The screenshot shows the Spring Tool Suite 4 IDE with the 'pom.xml' file open. The left sidebar displays the project structure, including the 'com.retobbva.challenge' package and its sub-packages. The main editor area shows the following XML code:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
4   <modelVersion>4.0.0</modelVersion>
5   <parent>
6     <groupId>org.springframework.boot</groupId>
7     <artifactId>spring-boot-starter-parent</artifactId>
8     <version>3.1.0</version>
9     <relativePath><!-- lookup parent from repository -->
10   </parent>
11   <groupId>com.retobbva</groupId>
12   <artifactId>challenge</artifactId>
13   <version>0.0.1-SNAPSHOT</version>
14   <name>challenge</name>
15   <description>Challenge from BBVA</description>
16   <properties>
17     <java.version>17</java.version>
18     <maven.compiler.source>17</maven.compiler.source>
19     <maven.compiler.target>17</maven.compiler.target>
20   </properties>
21   <dependencies>
22     <dependency>
23       <groupId>org.springframework.boot</groupId>
24       <artifactId>spring-boot-starter-data-jpa</artifactId>
25     </dependency>
26     <dependency>
27       <groupId>org.springframework.boot</groupId>
28       <artifactId>spring-boot-starter-web</artifactId>
29     </dependency>
30   </dependencies>
```

Programación reactiva RxJava



Base de datos MySQL



Dockerización de la base de datos y la API

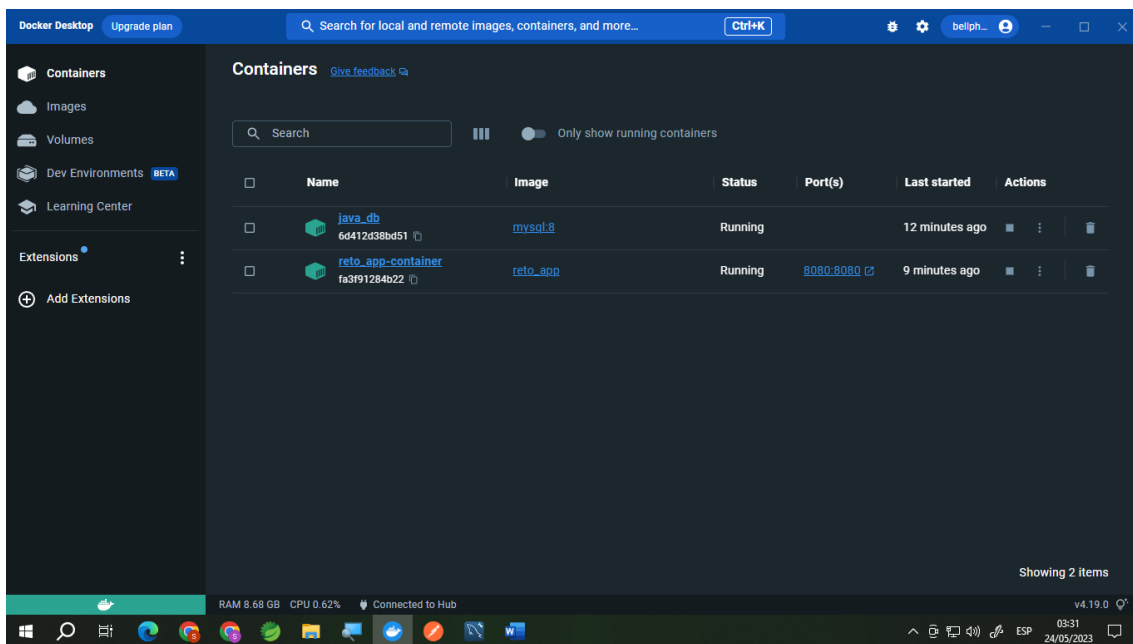
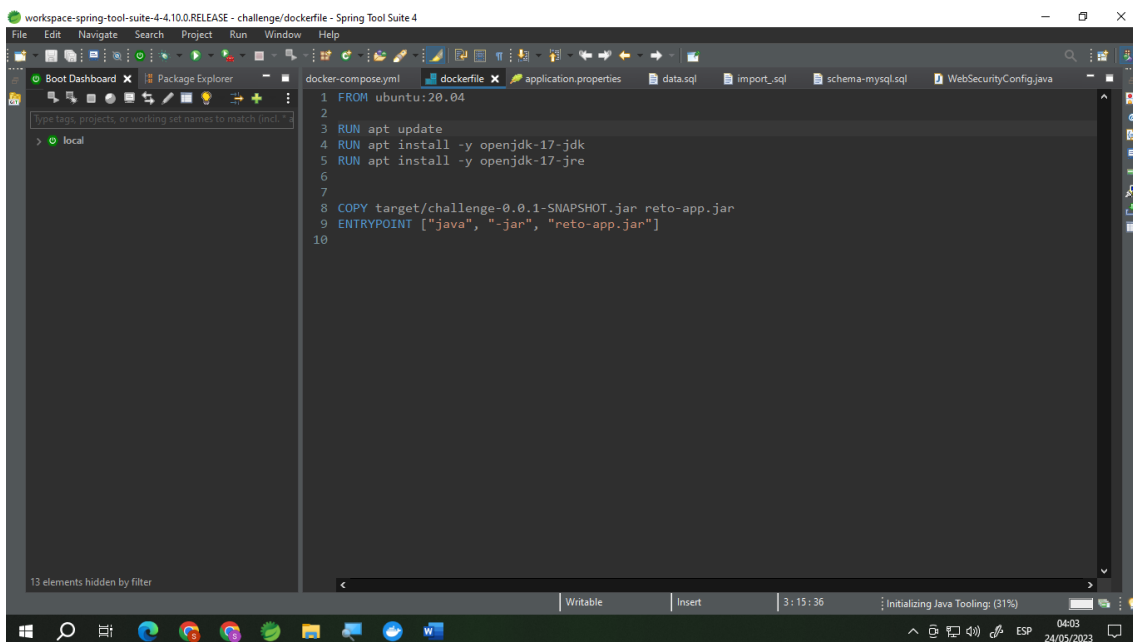
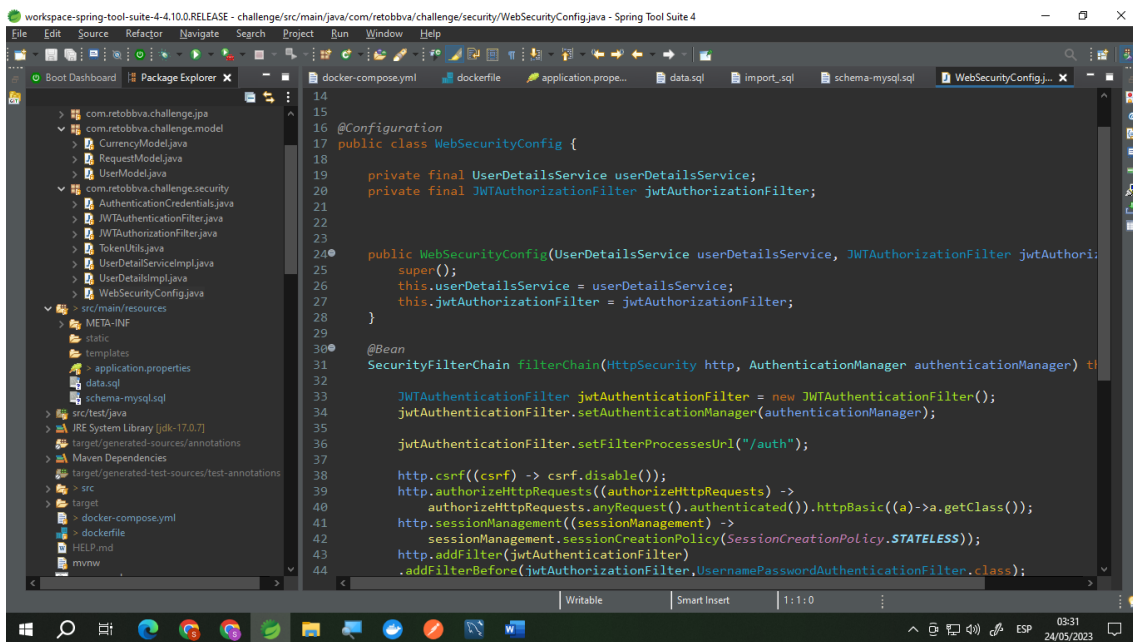


Imagen base para la dockerización Ubuntu 20.04

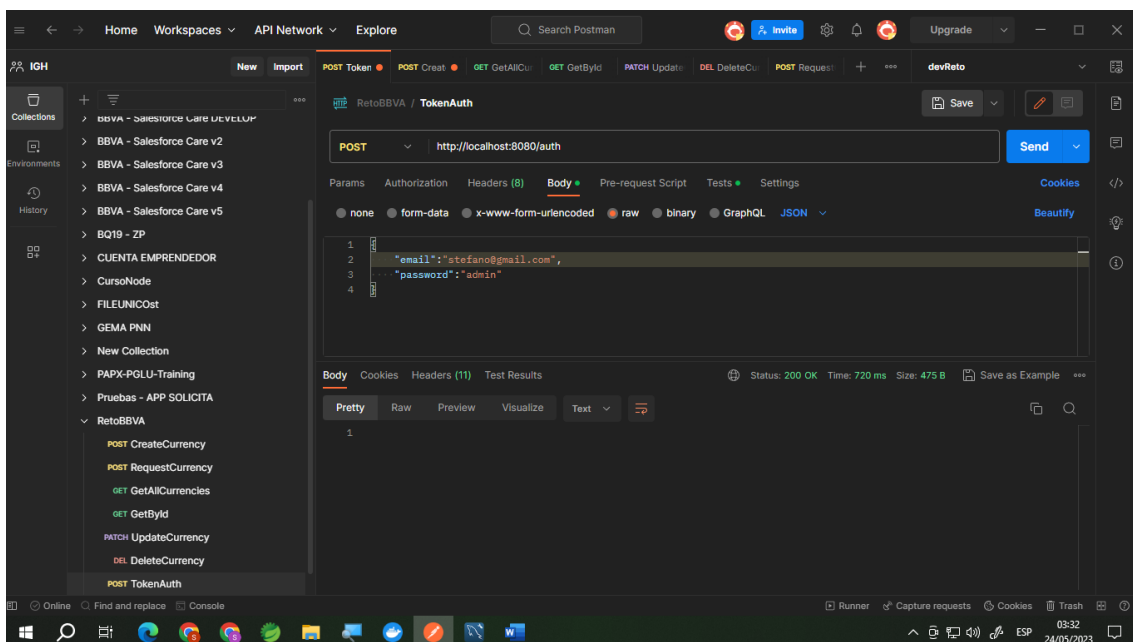


Seguridad de la API usando JWT

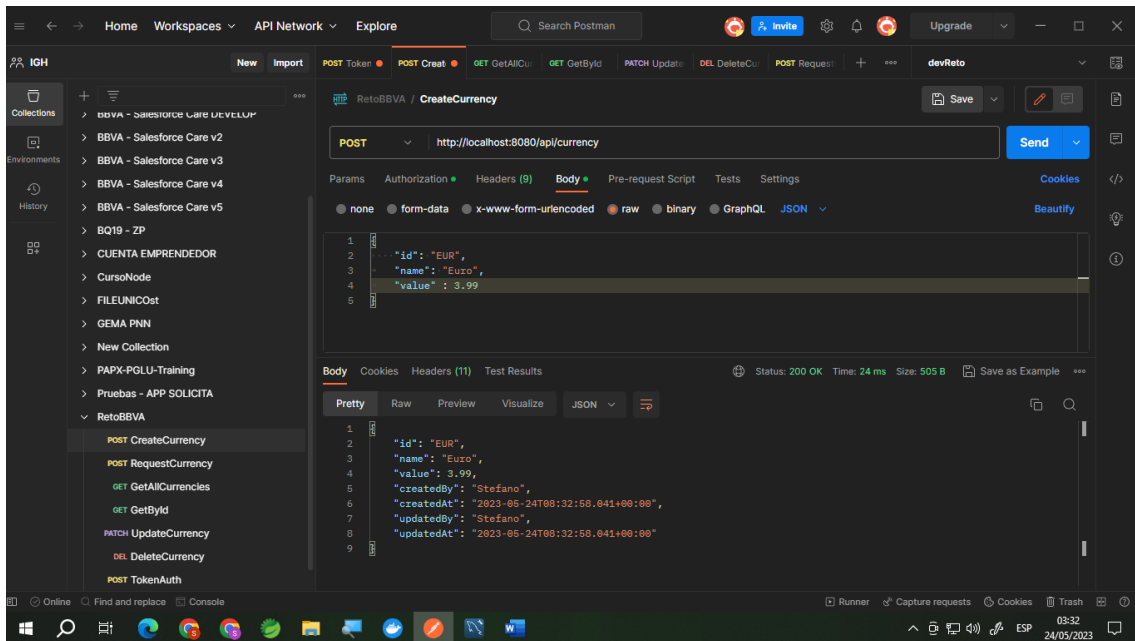


Consumo de la API mediante Postman

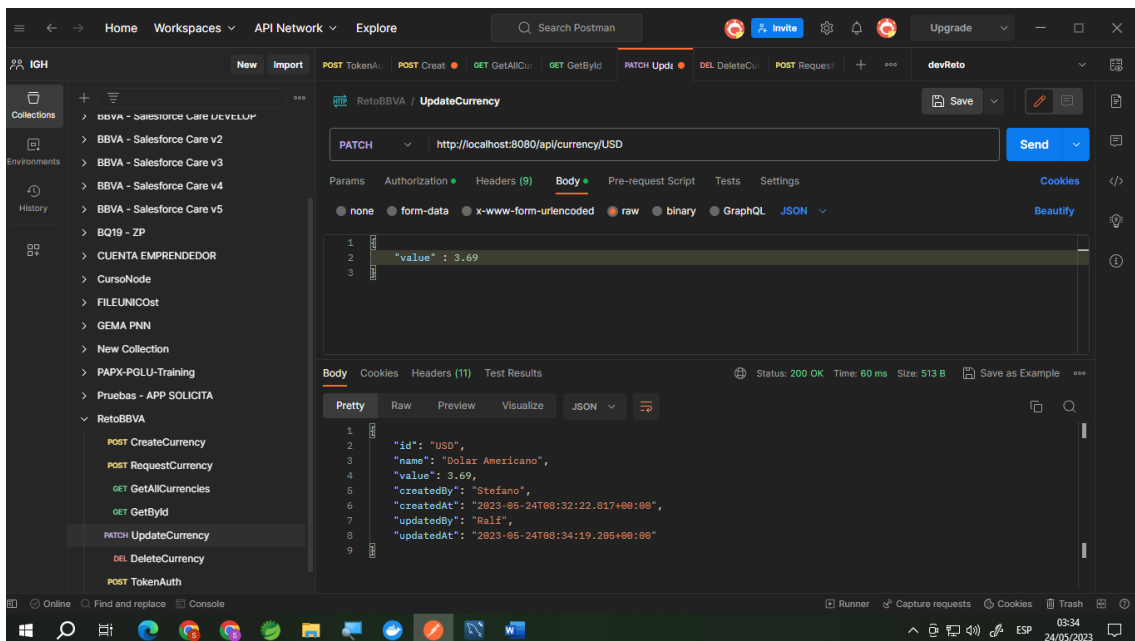
Obtención del token



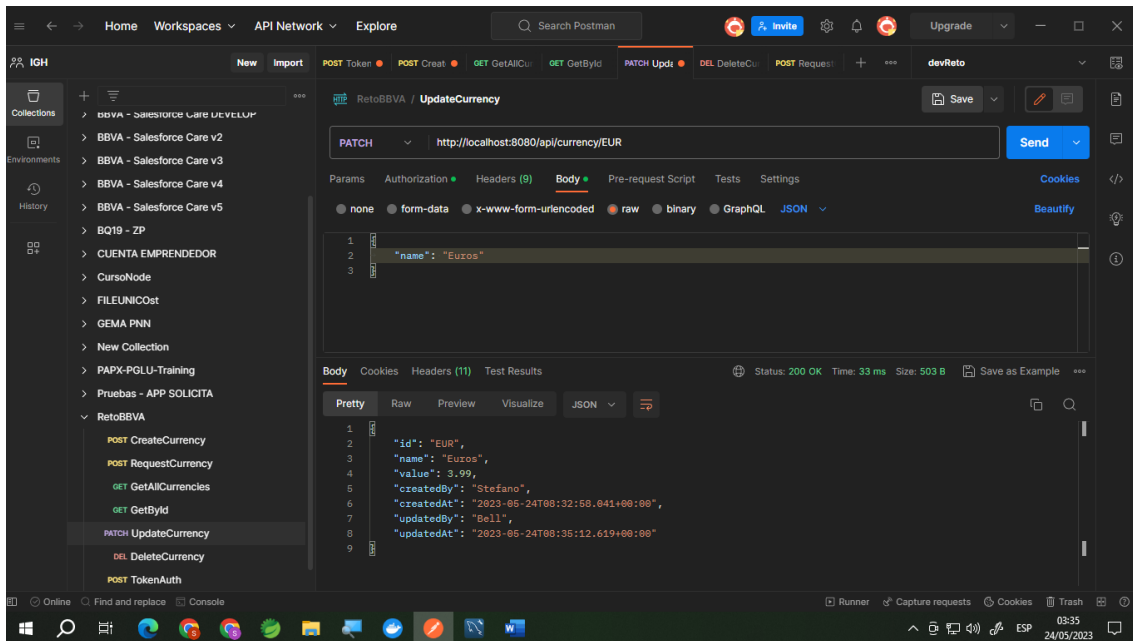
Creación de divisa



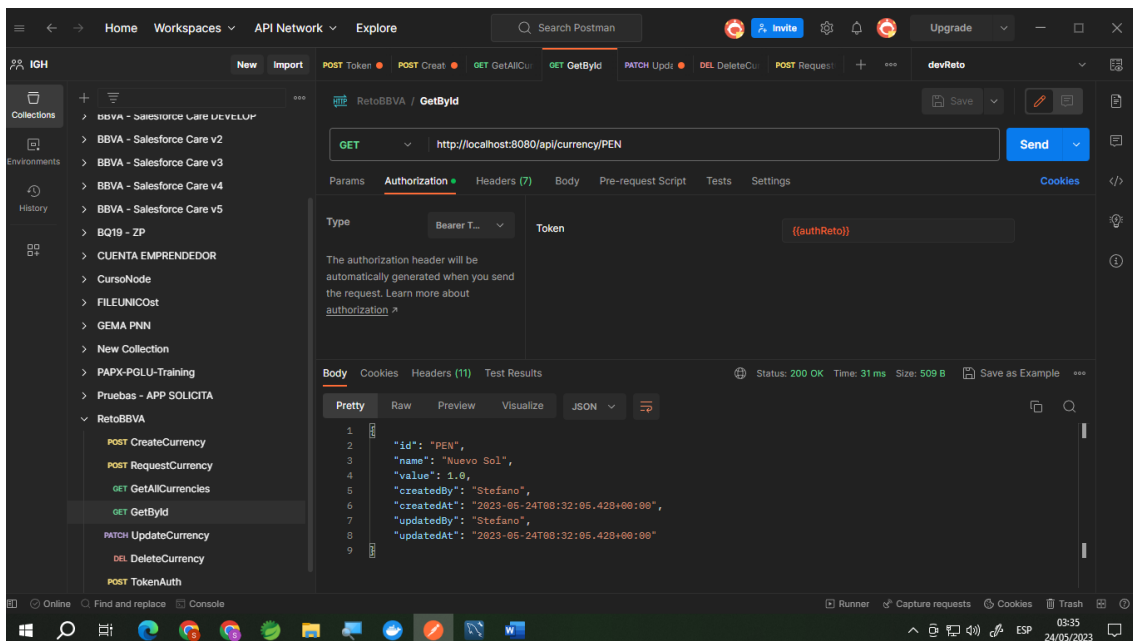
Actualización de divisa (value)



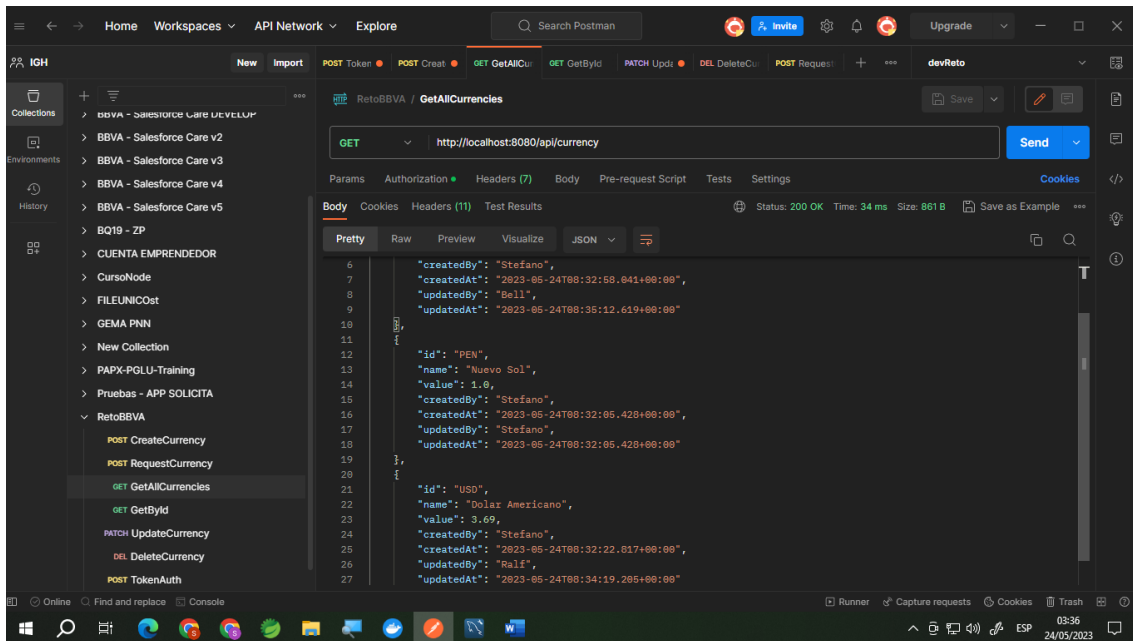
Actualización de divisa (name)



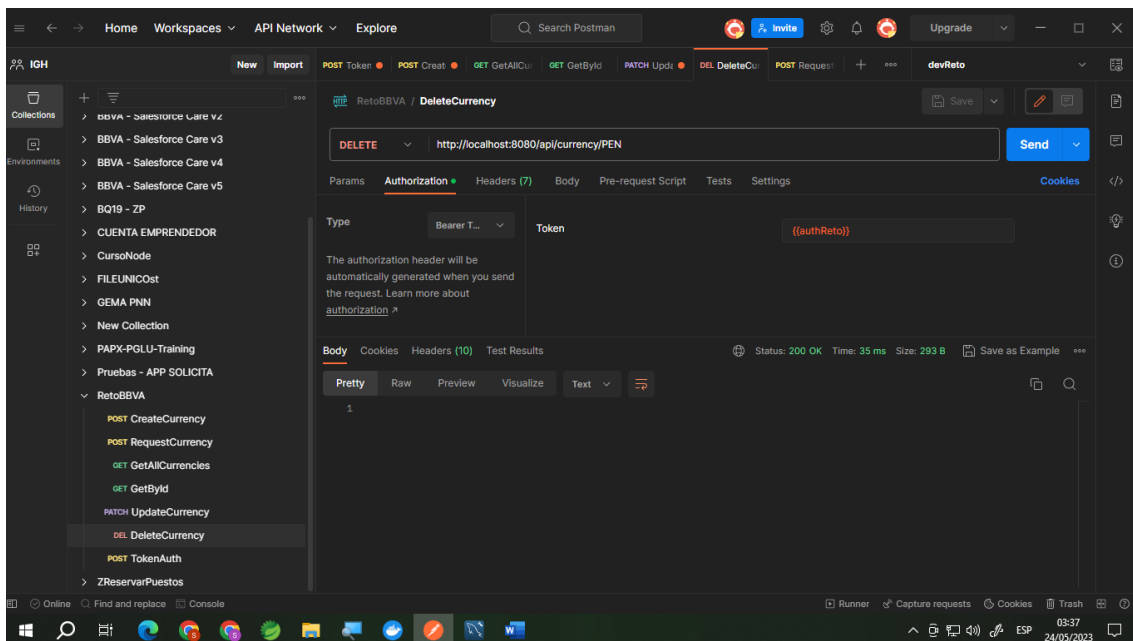
Obtención de una divisa por ID



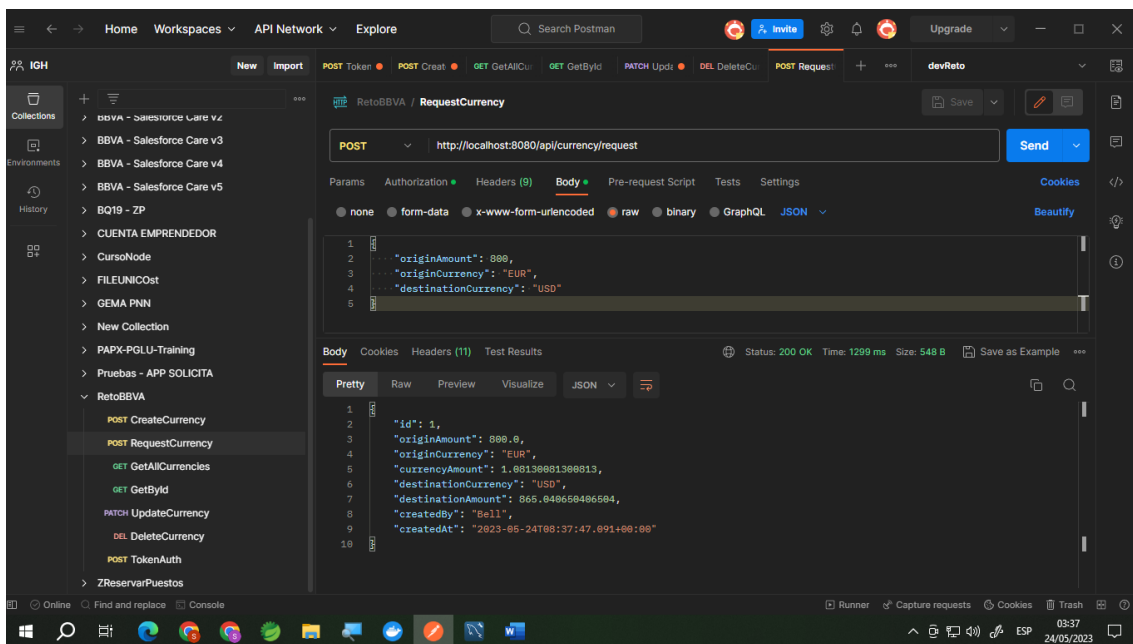
Obtención de todas las divisas registradas



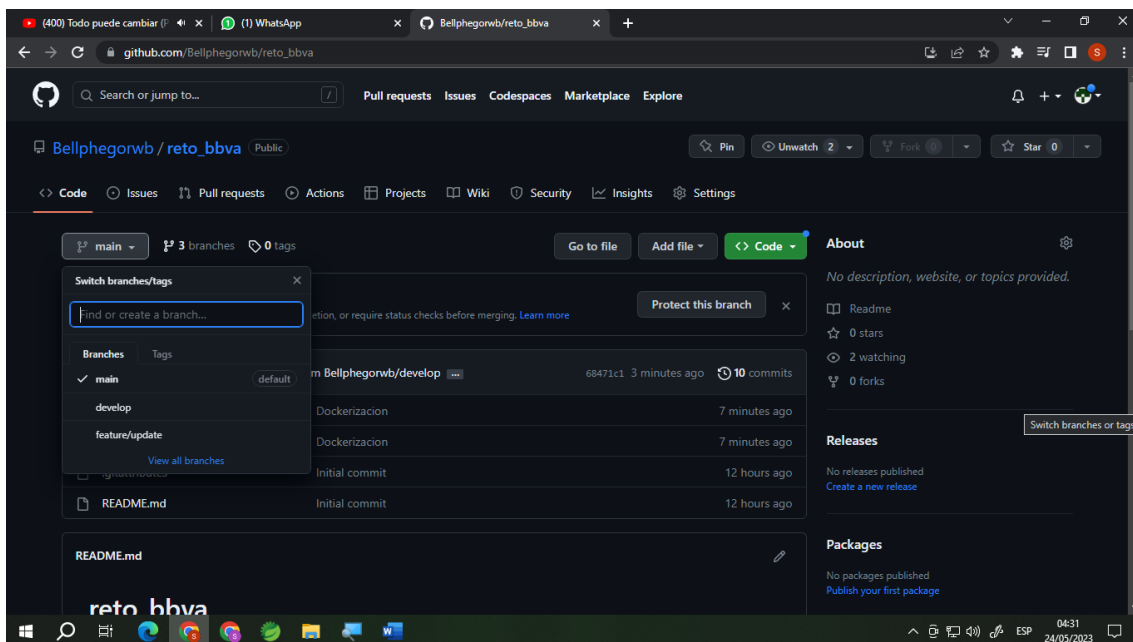
Eliminación de una divisa



Ejecución de una solicitud (800 EUR a USD)



Uso de las ramas en Github



Flujo de pull requests realizados

