HW1: Mid-term assignment report

Óscar Fernández Sánchez (101631)

1 Introduction

1.1 Overview of the work

This report presents the midterm individual project required for TQS, covering both the software product features and the adopted quality assurance strategy.
The project is a spring-boot application who reports the air pollutant of a city that you want. The measure is provided in µg/m3, and the data of the cities cames from the (API) and then saved in the in-disk database that is placed in the project directory and temporally in the cache memory. The project also allow to fetch data from its own api, who show the data that is saved from past fetchs to the external api, allowing to search the cities by id, by country prefix or directly listing all the cities that are there.

1.2 Current limitations

For now, the app only can to show the air pollutant of a certain city, but it can (evolucionar) to a most complete app adding some features, for example, list the countries by their air quality or for example implementing an own measurement system using sensors, like those that a microcontroller like Arduino can provide. It should also be noted that that the own rest api doesn´t allow to add or modify data, with the purpose of avoid conflicts with the external api.

Also the technology that I used to CI/CD(Heroku), if it detects inactivity in the project it flushes the entire database.

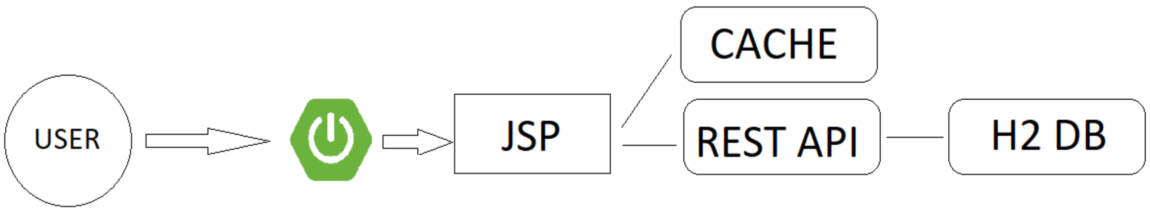
2 Product specification

2.1 Functional scope and supported interactions

The main form the user will interact with the app is through the pages that are provided. The user will fill the form with his city and the app will take the data of that city from the external api or from the internal cache. If the data is taked from the external api, the app “clean” the data and stored in his own api, that can be reached using the URLS specified below.
If the user enters a city that´s not saved in the external api, it will be redirected to an error page, which will give the user the opportunity to get back to the form.

2.2 System architecture

The application is based on Spring Boot, and JSP files are used to render the views, previously established the path in the application.properties file. For the REST API, the JPA technology and the H2 database are used, which is saved on disk within the project in the database.mv.db file .



2.3 API for developers

Base URL: http://www.localhost:8080/api || www.tqs1.herokuapp.com/api		
Action	URL	Objective
GET	/cities	List all the cities stored
GET	/city/name/{name}	Get the records of a given city by its name
GET	/city/id/{locationId}	Get a certain record of a city
GET	/city/ranking	Get the ranking of most contaminated cities
GET	/city/random	Get the latest record of a city
DELETE	/flush	Delete all the database of the api
DELETE	/flush/id/{locationId}	Delete city by its id

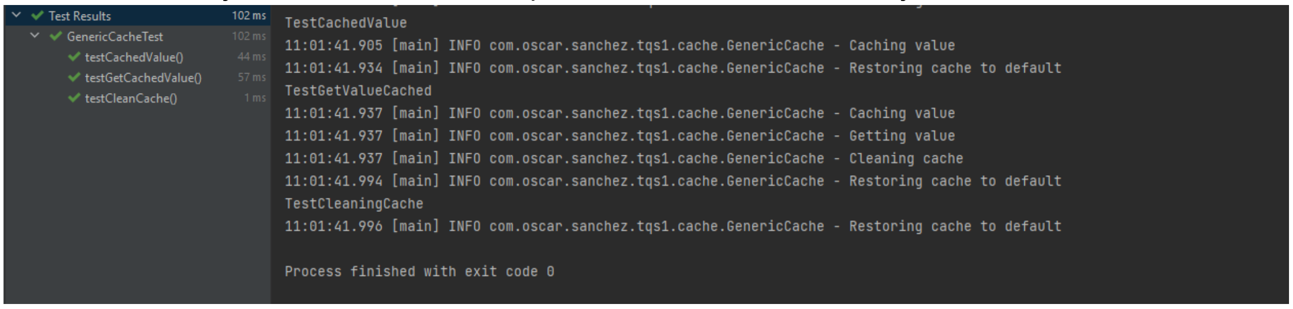
3 Quality assurance

3.1 Overall strategy for testing

Due to the simplicity of the project, not many tools were required to perform the tests. JUnit 5 has been used for the cache memory, to check the operation of both the controller and the JPA repository, MockMvc has been the chosen option. Finally, to verify that the views work, Selenium WebDriver has been used.

3.2 Unit and integration testing

Junit testing have been used to test if the cache memory runs as it expects. For the cache, three main scenarios have been raised: cach a value, get a cached value and finally clean the entire cache. As you can see in the next photo, it returns a satisfactory result.



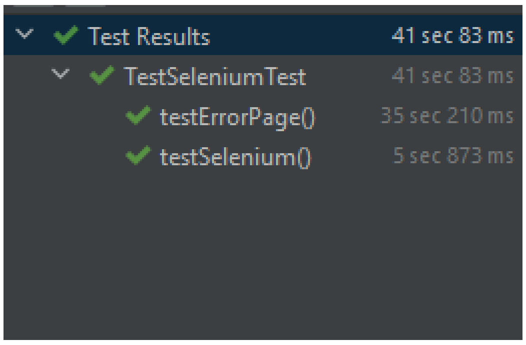
```
Test Results             102 ms
  GenericCacheTest       102 ms
    testCachedValue()     44 ms
    testGetCachedValue()  57 ms
    testCleanCache()       1 ms
```

```
TestCachedValue
11:01:41.905 [main] INFO com.oscar.sanchez.tqs1.cache.GenericCache - Caching value
11:01:41.934 [main] INFO com.oscar.sanchez.tqs1.cache.GenericCache - Restoring cache to default
TestGetValueCached
11:01:41.937 [main] INFO com.oscar.sanchez.tqs1.cache.GenericCache - Caching value
11:01:41.937 [main] INFO com.oscar.sanchez.tqs1.cache.GenericCache - Getting value
11:01:41.937 [main] INFO com.oscar.sanchez.tqs1.cache.GenericCache - Cleaning cache
11:01:41.994 [main] INFO com.oscar.sanchez.tqs1.cache.GenericCache - Restoring cache to default
TestCleaningCache
11:01:41.996 [main] INFO com.oscar.sanchez.tqs1.cache.GenericCache - Restoring cache to default

Process finished with exit code 0
```

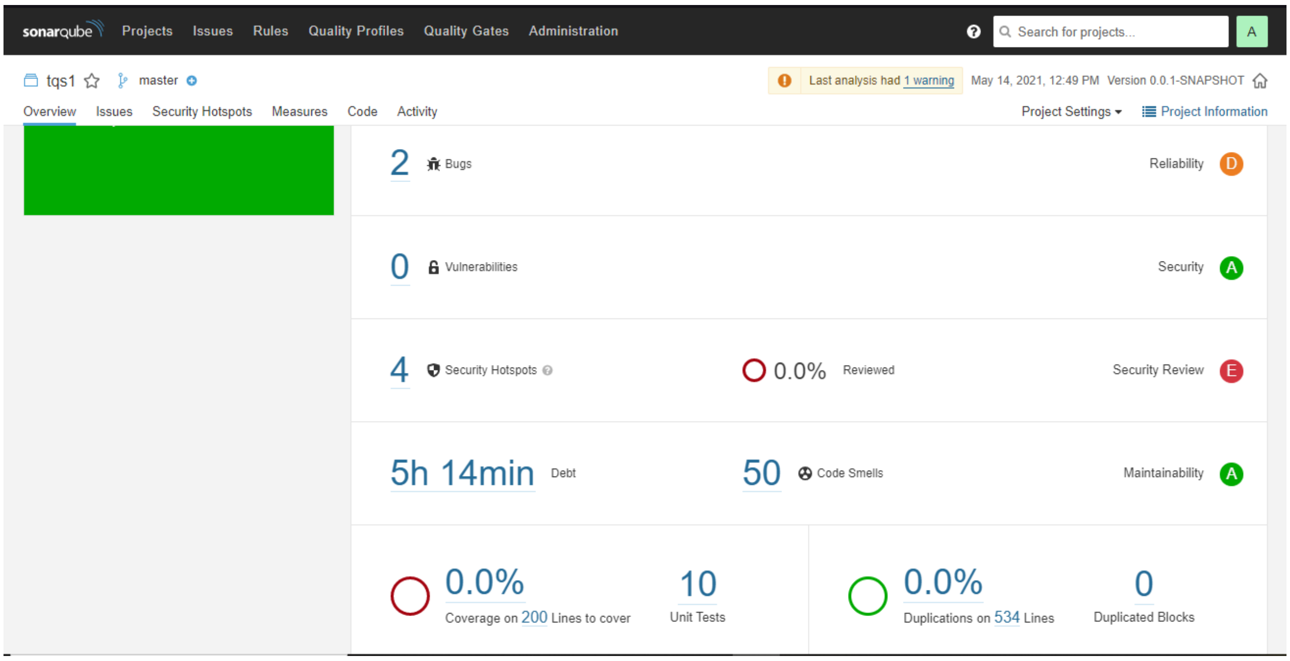
3.3 Functional testing

For the functional testing I used Selenium Web Driver, testing the app through Firefox.
As the application for now is very simple, not many scenarios have been considered. The 2 main scenarios are successful research, and not success and go to error page. As you can see in the screenshots, both test have passed in a satisfactory way, so It´s visible that the front-end is stable.



```
Test Results             41 sec 83 ms
  TestSeleniumTest       41 sec 83 ms
    testErrorPage()      35 sec 210 ms
    testSelenium()        5 sec 873 ms
```

3.4 Static code analysis



Last analysis had 1 warning May 14, 2021, 12:49 PM Version 0.0.1-SNAPSHOT

2 Bugs — Reliability D

0 Vulnerabilities — Security A

4 Security Hotspots — 0.0% Reviewed — Security Review E

5h 14min Debt — 50 Code Smells — Maintainability A

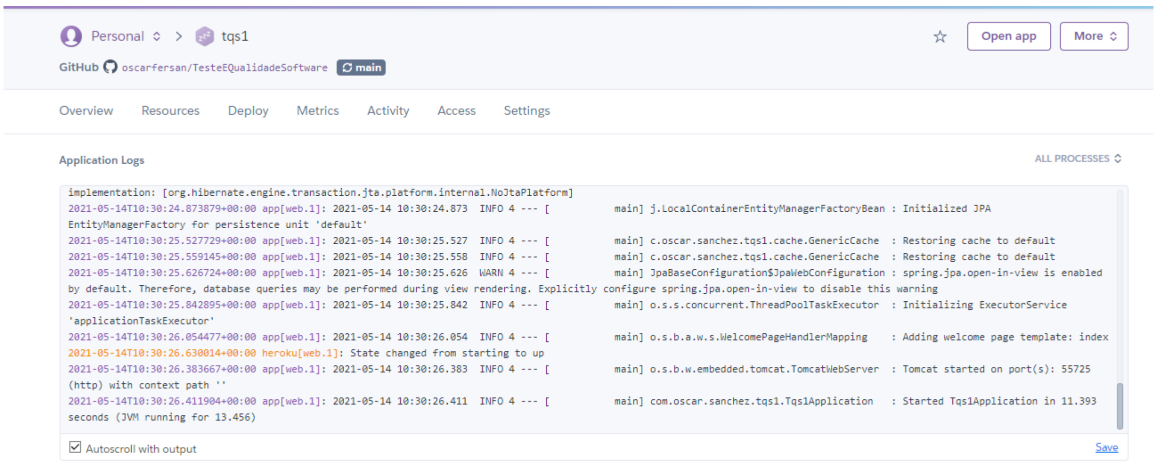
0.0% Coverage on 200 Lines to cover — 10 Unit Tests

0.0% Duplications on 534 Lines — 0 Duplicated Blocks

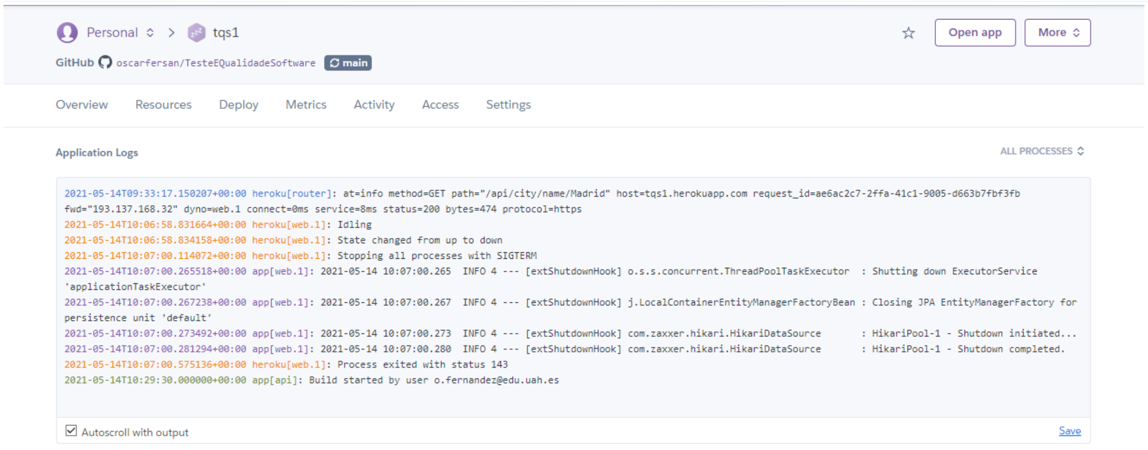
In the exposed image you can see that Sonar Qube notifies that there are 2 bugs, but those bugs are irrelevant, since they have to do with the declaration of the Random class in the method that belongs to the /api/city/random path.
The vulnerabilities that appear are due to the low security provided in the application routes, but it is not worrying as it is not sensitive information such as passwords or other personal information

3.5 Continuous integration pipeline [optional]

For CI/CD I used Heroku, who allows CI/CD if you connect the github project to the page. Every time I push the repository subdirectory, Heroku detects it and builds the project.



```
implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
2021-05-14T10:30:24.873879+00:00 app[web.1]: 2021-05-14 10:30:24.873  INFO 4 --- [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA
EntityManagerFactory for persistence unit 'default'
2021-05-14T10:30:25.527729+00:00 app[web.1]: 2021-05-14 10:30:25.527  INFO 4 --- [           main] c.oscar.sanchez.tqs1.cache.GenericCache  : Restoring cache to default
2021-05-14T10:30:25.559145+00:00 app[web.1]: 2021-05-14 10:30:25.558  INFO 4 --- [           main] c.oscar.sanchez.tqs1.cache.GenericCache  : Restoring cache to default
2021-05-14T10:30:25.626724+00:00 app[web.1]: 2021-05-14 10:30:25.626  WARN 4 --- [           main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled
by default. Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
2021-05-14T10:30:25.842895+00:00 app[web.1]: 2021-05-14 10:30:25.842  INFO 4 --- [           main] o.s.s.concurrent.ThreadPoolTaskExecutor  : Initializing ExecutorService
'applicationTaskExecutor'
2021-05-14T10:30:26.054477+00:00 app[web.1]: 2021-05-14 10:30:26.054  INFO 4 --- [           main] o.s.b.a.w.s.WelcomePageHandlerMapping    : Adding welcome page template: index
2021-05-14T10:30:26.630014+00:00 heroku[web.1]: State changed from starting to up
2021-05-14T10:30:26.383667+00:00 app[web.1]: 2021-05-14 10:30:26.383  INFO 4 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 55725
(http) with context path ''
2021-05-14T10:30:26.411904+00:00 app[web.1]: 2021-05-14 10:30:26.411  INFO 4 --- [           main] com.oscar.sanchez.tqs1.Tqs1Application   : Started Tqs1Application in 11.393
seconds (JVM running for 13.456)
```



```
2021-05-14T09:33:17.150207+00:00 heroku[router]: at=info method=GET path="/api/city/name/Madrid" host=tqs1.herokuapp.com request_id=ae6ac2c7-2ffa-41c1-9005-d663b7fbf3fb
fwd="193.137.168.32" dyno=web.1 connect=0ms service=8ms status=200 bytes=474 protocol=https
2021-05-14T10:06:58.831664+00:00 heroku[web.1]: Idling
2021-05-14T10:06:58.834158+00:00 heroku[web.1]: State changed from up to down
2021-05-14T10:07:00.114072+00:00 heroku[web.1]: Stopping all processes with SIGTERM
2021-05-14T10:07:00.265518+00:00 app[web.1]: 2021-05-14 10:07:00.265  INFO 4 --- [extShutdownHook] o.s.s.concurrent.ThreadPoolTaskExecutor  : Shutting down ExecutorService
'applicationTaskExecutor'
2021-05-14T10:07:00.267238+00:00 app[web.1]: 2021-05-14 10:07:00.267  INFO 4 --- [extShutdownHook] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for
persistence unit 'default'
2021-05-14T10:07:00.273492+00:00 app[web.1]: 2021-05-14 10:07:00.273  INFO 4 --- [extShutdownHook] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Shutdown initiated...
2021-05-14T10:07:00.281294+00:00 app[web.1]: 2021-05-14 10:07:00.280  INFO 4 --- [extShutdownHook] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Shutdown completed.
2021-05-14T10:07:00.575136+00:00 heroku[web.1]: Process exited with status 143
2021-05-14T10:29:30.000000+00:00 app[api]: Build started by user o.fernandez@edu.uah.es
```

4 References & resources

Project resources

- Video demo
  https://github.com/oscarfersan/TesteEQualidadeSoftware/blob/main/tqs1VideoDemo.mp4
- Ready to use application: https://tqs1.herokuapp.com

Reference materials

External API: https://docs.openaq.org/
CacheMemory tutorial:
Spring:https://www.youtube.com/watch?v=kFIvslQQZ9k&list=PLU8oAIHdN5BIq85GIxtKjIXdfHPksV_Hm
Heroku: https://devcenter.heroku.com