

# HW1: Mid-term assignment report

## Testing and Software Quality

Rafael Remígio 102435

April 7, 2023

Departamento de Electrónica, Telecomunicações e Informática  
Universidade de Aveiro  
Year 2022/2023

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Overview of the work . . . . .	1
1.2	Current limitations . . . . .	1
<b>2</b>	<b>Product specification</b>	<b>1</b>
2.1	Functional scope and supported interactions . . . . .	1
2.2	System architecture . . . . .	2
2.3	API for developers . . . . .	3
<b>3</b>	<b>Quality assurance</b>	<b>3</b>
3.1	Logging . . . . .	3
3.2	Overall strategy for testing . . . . .	3
3.3	Unit and integration testing . . . . .	3
3.4	Functional testing . . . . .	3
3.5	Code quality analysis . . . . .	3
3.6	Continuous integration pipeline . . . . .	3
<b>4</b>	<b>References &amp; resources</b>	<b>3</b>
4.1	Reference materials . . . . .	3

## 1 Introduction

### 1.1 Overview of the work

In this assignment the web application developed provides information about parameters related to air quality of a certain location.

### 1.2 Current limitations

## 2 Product specification

### 2.1 Functional scope and supported interactions

The WebApp provides two use cases:

1. search for a specific location and receive Air Quality information. This information consists of the specific parameters:
  - Particulate Matter PM2.5
  - Particulate Matter PM10

- Carbon Monoxide CO
- Nitrogen Dioxide NO2

2. access and view statistics of the cache's usage:

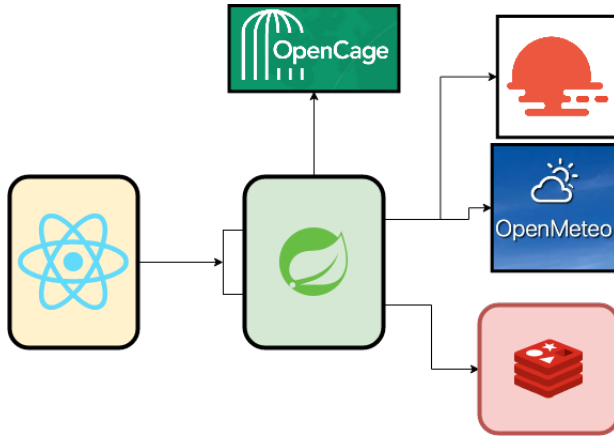
- Misses
- Hits
- Accesses

## 2.2 System architecture

### 2.2.1 Architecture



### 2.2.2 Technologies and API's Used



## 2.3 API for developers

# 3 Quality assurance

## 3.1 Logging

Logging is a crucial aspect of software development that plays a vital role in supporting production and debugging activities.

Logging was conducted using the slf4j Logger class as it provides an easy integration with the Spring Boot Application.

Each log message contains a timestamp, method invocation details, custom log messages, and other contextual data. Additionally, each log entry is assigned a logging level identifier, providing a way to categorize and prioritize log entries based on their significance.

I followed logging principles and best practices - such as: logging at the correct level, providing meaningful messages, etc - in order to provide meaningful logs.

## 3.2 Overall strategy for testing

## 3.3 Unit and integration testing

## 3.4 Functional testing

## 3.5 Code quality analysis

SonarQube

### **3.6 Continuous integration pipeline**

## **4 References & resources**

GitRepository can be found at my github:

### **4.1 Reference materials**