TQS: Quality Assurance manual

***Daniel Pinto [89341], Diogo Andrade [89265], João Dias [89236], Rodrigo Oliveira [90514]***

v2020-06-05

[1 Project management 1](#_Toc39440117)

[1.1 Team and roles 1](#_Toc39440118)

[1.2 Agile backlog management and work assignment 1](#_Toc39440119)

[2 Code quality management 2](#_Toc39440120)

[2.1 Guidelines for contributors (coding style) 2](#_Toc39440121)

[2.2 Code quality metrics 2](#_Toc39440122)

[3 Continuous delivery pipeline (CI/CD) 2](#_Toc39440123)

[3.1 Development workflow 2](#_Toc39440124)

[3.2 CI/CD pipeline and tools 2](#_Toc39440125)

[3.3 Artifacts repository [Optional] 2](#_Toc39440126)

[4 Software testing 2](#_Toc39440127)

[1.1 Overall strategy for testing 2](#_Toc39440128)

[1. Functional testing/acceptance 2](#_Toc39440129)

[2. Unit tests 3](#_Toc39440130)

[3. System and integration testing 3](#_Toc39440131)

[4. Performance testing [Optional] 3](#_Toc39440132)

# Project management

## Team and roles

Team manager – João Dias

Responsible for managing the distribution of tasks within the team and promoting meetings between the members.

Product owner – Diogo Andrade

Responsible for laying out the objectives of the projects and what features need to be developed and their priorities

DevOps Master – Daniel Pinto

Responsible for configuring the deployment of the application and ensuring that all the code is being correctly integrated into the project with git

Developer – Rodrigo Oliveira and everyone else

Everyone contributes to the development of the project

## Agile backlog management and work assignment

For this project we decided to use PivotalTracker to keep track of the user stories being implemented as we moved forward with the project, along with who was responsible for implementing each of them.

https://www.pivotaltracker.com/n/projects/2448127

# Code quality management

## Guidelines for contributors (coding style)

N/A

## Code quality metrics

For static code analysis, we decided to use Codacy

https://app.codacy.com/manual/dmatiaspinto/tqs\_project/dashboard

# Continuous delivery pipeline (CI/CD)

## Development workflow

Most new features had new branches associated to them created, where developers would develop the feature fully before communicating that the feature was working and ready to be merged into the main working tree.

Whenever this happened, the other members of the team would go into that branch, verify that everything was working properly and then a merge would occur.

A user story was marked as done when the developed functionality hit all the expected requirements and functions and was approved by the Product Owner.

## CI/CD pipeline and tools

Whenever a new commit was pushed to the master branch, GitLab created a new pipeline for deploying the newly developed features and verifying if everything was working correctly.

## Artifacts repository [Optional]

N/A

# Software testing

## Overall strategy for testing

The test development strategy was BDD.

In our tests we basically tested against possible points of failure in different parts of the application, ensuring that we had employed sufficient coverage against failure in those features.

## Functional testing/acceptance

N/A

## Unit tests

The policy for writing unit tests was open box testing.

## System and integration testing

The policy for writing integration tests was open box testing.

## Performance testing [Optional]

N/A