

# Open source e-learning plataforma

Projeto e Seminário

Licenciatura em Engenharia Informática e Computadores

André de Oliveira nº44580

Rodrigo Leal nº44823

917135594 alaço@hotmail.com

961604272 rodrigomfl@hotmail.com

João Santos nº44847

926366577 j.ers.santos@outlook.com

Advisors:

Cátia Vaz, cvaz@cc.isel.ipl.pt

José Simão jsimao@cc.isel.ipl.pt

March 30, 2020

## 1 Introduction

Nowadays there are some platforms that provide an environment for defining algorithms and testing, but not all of them are open source, don't have such an appealing environment or they don't allow multi-language.

Therefore, this project intends to develop an open source e-learning platform, dedicated to the definition and testing of algorithms in a multi-language environment.

This platform will have several challenges which can be solved for study or evaluation purposes. This can be useful in academic environments, professional interviews or programming enthusiast.

## 2 Analysis

To develop the e-learning platform three main modules were identified: UI, Services and Execution Environments.

The UI module is the presentation layer, with which the final user will interact. This

interface will be developed as a single page application.(Fowler, 2002, p. 2)

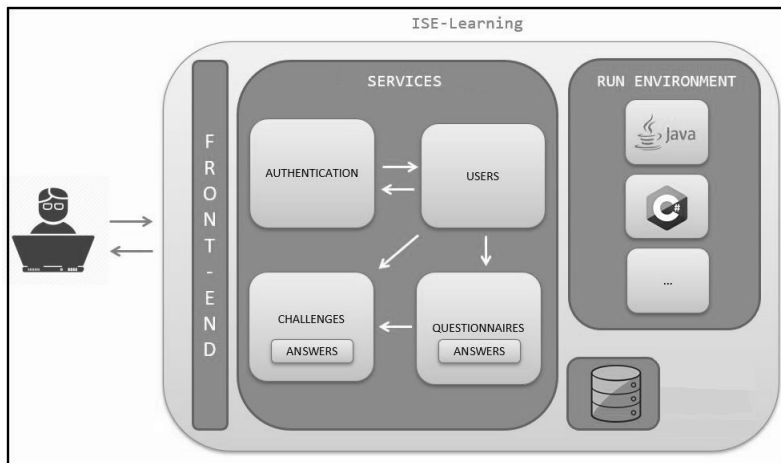


Figure 1: Architecture

vironments, where each application will be developed and hosted on a separate container.Jain (2019)

Some risks were identified during the project analysis: scalability, requires extensive load tests with the solution deployed on cloud environment; lack of diverse userbase for testing, complex solutions required thorough testing to minimize bugs; group members are currently working and studying at same time; the learning process required for new tools has inherent uncertainty; estimates may not meet proposed values.

The major constraint of this project's execution is time management. Taking this into account several strategies were devised as contingency plans if need be: scope reduction, removing questionnaire functionality.

## 2.1 Requirements

After analysing the objectives of this project we identified functional and non-functional requirements.

Functional:

- Multi-Language Provide run environment for multiple programming languages.(eg. Java, Kotlin, C#, JS, Python)
- Execute Solution Any user may write a code solution, run it and get the result through the front-end application.
- Basic Authentication Users can create an account, with a profile information and will be able to get access to more features (eg. keep track of the solved challenges, create/solve Questionnaires to/from other users).

- Challenges are a programming problem that needs to be solve. Every challenge has a built in solution that will be compared with the user submitted answer to determinate its "correctness" through unit tests.
  - To create a Challenge a user must supply its unit tests, and they have to compile/run successfully before being able to submit it. Only its creator can edit the Challenge.
  - Challenges may have tags associated, and they can be searched by them.
  - A logged in user can track the Challenge he/she submitted.
- Questionnaires are a group of Challenges.
  - Questionnaires can only be submitted by logged in users.
  - Only its creator can edit the Questionnaire.
  - A user may choose any number of Challenges to create a Questionnaire and set it a timer. (eg. user A created a Questionnaire with 5 Challenges that has to be solved by user B in one hour).
  - Questionnaires can be shared through a link created by the platform.

#### Non-Functional Requirements:

- Security Executing third party code in a machine raises security concerns.
  - A self-contained run environment limits the impact of malicious code to the container which executes it, protecting the remaining infrastructure.
- Solution Maintenance Maintaining a complex solution requires a balance between many moving parts, as such this project's architecture reflects the principles of loose coupling and modularity which facilitate the solution maintenance.
- Open source Open source software refers to something people can modify and share because its design is publicly accessible.

### 3 Schedule

In the next two tables can be consulted the schedule for the assignments to be completed during the project's implementation (Table 1) and the description of each assignment type (Table 2).

Table 1: Requirements

Date	Assignment	Milestone
09-Mar	Introduction to docker, react, bash, python	
16-Mar	Set up DB, webserver react configuration, E(JVM)	Project Proposal 16 Mar
23-Mar	Set up DB, webserver react configuration, E(JVM)	
30-Mar	E(JVM), E(Node), P(Execute code)	
06-Apr	E(JVM), E(Node), P(Execute code)	
13-Apr	E(JVM), E(Node), P(Home)	
20-Apr	Progress Report and Individual presentation	
27-Apr	Progress Report and Individual presentation	
04-May	S(Challenges), P(Home), S(User)	Progress Report and Individual presentation 4 May
11-May	S(Challenges), S(User), S(Questionnaire)	
18-May	S(Challenges), P(User's Profile), S(Questionnaire)	
25-May	Poster and Beta version, S(Questionnaire)	
01-Jun	S(Challenges), P(User's Profile), S(Questionnaire)	Poster and Beta version 1 June
08-Jun	P(Login and Signin ), P(Fill Questionnaire), S(Authentication)	
15-Jun	P(Login and Signin ), P(Fill Questionnaire), S(Authentication)	
22-Jun	Exams	
29-Jun		
06-Jul		
13-Jul		
20-Jul		
27-Jul	P(Challenge), E(Python), E(CLR)	
03-Aug	P(Challenge), E(Python), E(CLR)	
10-Aug	P(Challenge), E(Python), E(CLR)	
17-Aug	Cloud environment deployment	
24-Aug	Cloud environment deployment	
31-Aug	Wrap up final report	
07-Sep	Wrap up final report	Final delivery 12 Sept

Table 2: Assignment Types

Assignment Type	Description
E(x)	Design, Implementation and deploying to the cloud of a solution capable of running code "remotely"
P(x)	Creation of webpage
S(x)	Design and Implementation of backend service

## References

- Fowler, Martin. 2002. *Patterns of Enterprise Application Architecture*. 1 edn. Addison-Wesley Educational Publishers Inc.
- Jain, Shashank Mohan. 2019. *Linux Containers and Virtualization - A kernel perspective*. 3 edn. Independently published.