

ASInt – MEEC
2019/2020
Project (V1)

In this project students should develop a web application used to ease the access of Instituto superior Técnico resources information, by means of web site, mobile applications or QR-Codes.

In an organization such as IST lots of resources (buildings, rooms, services, people, canteen) have attributes information that can be accessed by different ways (mobile application, mobile phone, ...).

The objective of this work is to develop a system that allows such interaction.

In this work students should

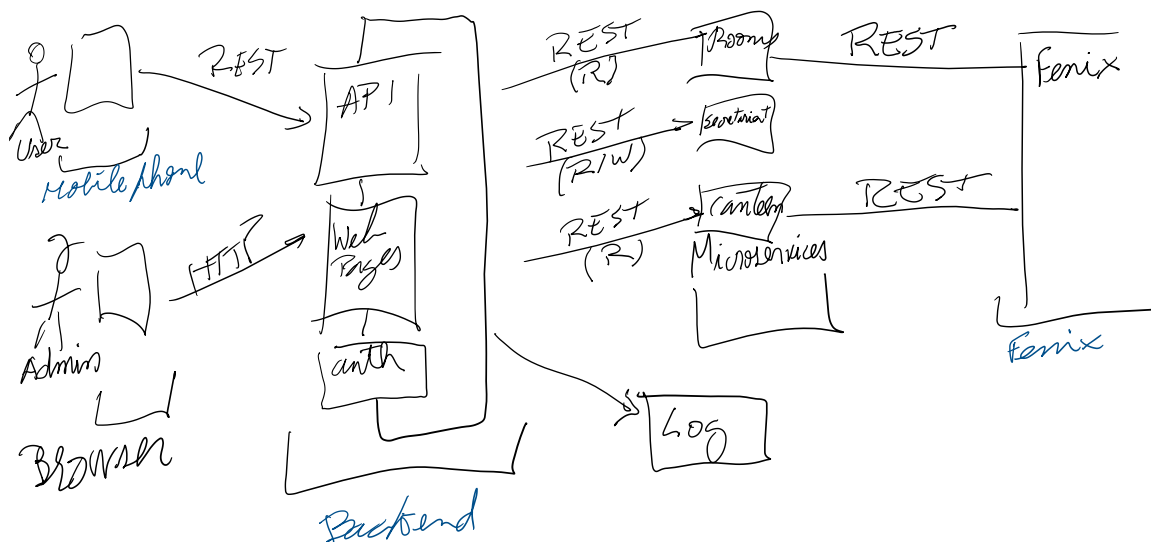
- Define the set of resources to be made available
- Define the relevant information (attributes) of such resources
- Define the interfaces (WEB and REST) to access such resources
- Implement a server prototype that illustrates the access to the information
- Implement a simple web server for access and management of resources
- Implement a simple mobile application

The system will be operated by four different class of users:

- Programmer that implements the mechanism to allow new resources
- Administrator that edit some of the resources information
- Regular users that access the resources
- Authenticated user that use a mobile application

1 Architecture of the system

The system will be implemented following the micro-service architecture, where each functionality is implemented by a simple web-service and will have the following components



List of components:

- FENIX – origin of some of the data that will be made available to users (canteen, rooms)
- Microservices – small web-services that will be responsible for managing the requests corresponding resource. Each webservice will be responsible for handling on resource. Depending on the resource these web-services can store some information.
- Backend – a set of public API and web pages that will allow the access and manipulation of the various

resources

- Admin Web pages – a set of pages that allow the administrator to manage and edit some of the resources
- Mobile access – a simple application that can be executed in a mobile phone that allows users to access the resources information. If the user authenticates in this application he will be able to prove its identity to another user.
- Log – the service will receive and store all the accesses to every of the following components:
 - Backend (web and REST accesses)
 - Rooms microservice
 - Secretariat microservice
 - Canteen microservice
 - Authentication

2 FENIX

The Fenix webservice will provide part of the information necessary by the microservices.

2.1 REST endpoints

The rest endpoint for the Fenix is available in <https://fenixedu.org/dev/api/>

2.2 User authentication

It is possible to use the FENIX istID and password in web applications using the OAuth protocol. A description of the way to integrate on web application with the Técnico authentication is available here: <https://fenixedu.org/dev/tutorials/use-fenixedu-api-in-your-application/>

3 Micro services

In this project students should define a suitable API that will follow the common REST API patterns. This section describes what information should be stored in each of the microservices and that will be exposed by the developed API.

These microservices will need to make some data persistent. To simplify the project students can use the python pickle serialization library.

3.1 Rooms

The **rooms microservices** will provide an API to expose each **room** information. Each room should have an ID and the information exported by each room should be:

- Location: Campi and building
- Timetable: list of events (classes or exams, ...) assigned to the room

3.2 Secretariats

Técnico has several secretariats (for instance Secretaria Académica) they have a certain location and opening hours.

Each Secretariat should have one identifier, and the **secretariat microservice** will provide an API to expose and edit the following information:

- Location

- Name
- Description
- Opening hours (can be text)

3.3 Canteen

The **canteen microservice** will provide an API to expose the canteen menus.

4 Backend

The backend is divided into 3 components. Each component will have one specific set of functionalities.

4.1 API

The API component will export all the REST endpoints necessary by the mobile application. These endpoints should only allow the access in read mode to the information. These endpoints can replicate those in the microservices.

4.2 Web pages

The web pages module will present a set of web pages that will allow different users to access information:

- Administrators will be able to edit the Secretariats information and access logs
- Regular users will be able to query (using the resources identifiers) and see each resource information (see section 6.1).

4.3 Auth

The authentication module will allow users to authenticate themselves using the FENIX ISTid and password. It should implement the OAuth protocol.

This module should implement the algorithm to allow users using the mobile application to prove who they are.

5 Admin web pages

The admin pages will be accessed by the administrator to:

- list the logs
- edit (create and change) the Secretariats information.

6 Mobile access to the information

Users using a mobile phone can access the information about the resources.

One way is taking advantage of QR-Code affixed close to the physical resources, another is using the mobile application.

6.1 QR-Code

Each physical resource (room, or secretariat) will have a QR-Code affixed to its door that will allow any user holding a smartphone (with a qr-code reader and internet access) to access the resource web-page (section 4.2).

In order to access this information, it is not necessary to use the developed mobile application., it is only necessary to use a QR-Code reader and a browser.

6.2 Mobile application

The mobile application will be composed by a set of simple web pages that will be available after the user authenticates.

This application will implement the following functionalities in three different pages:

- Scan a QR-code and access the resource information using the REST API (from section 4.1)
- Present a secret to allow other user to validate his identity. After the secret is used this page shows the identity of the other user.
- Type one secret to verify the other user identify. If the secret is valid it show its name and photo.
- Show the rooms and services available om the current building (the location service will be simulated by a for where the user inputs the building he is in).

6.2.1 Access to resource information

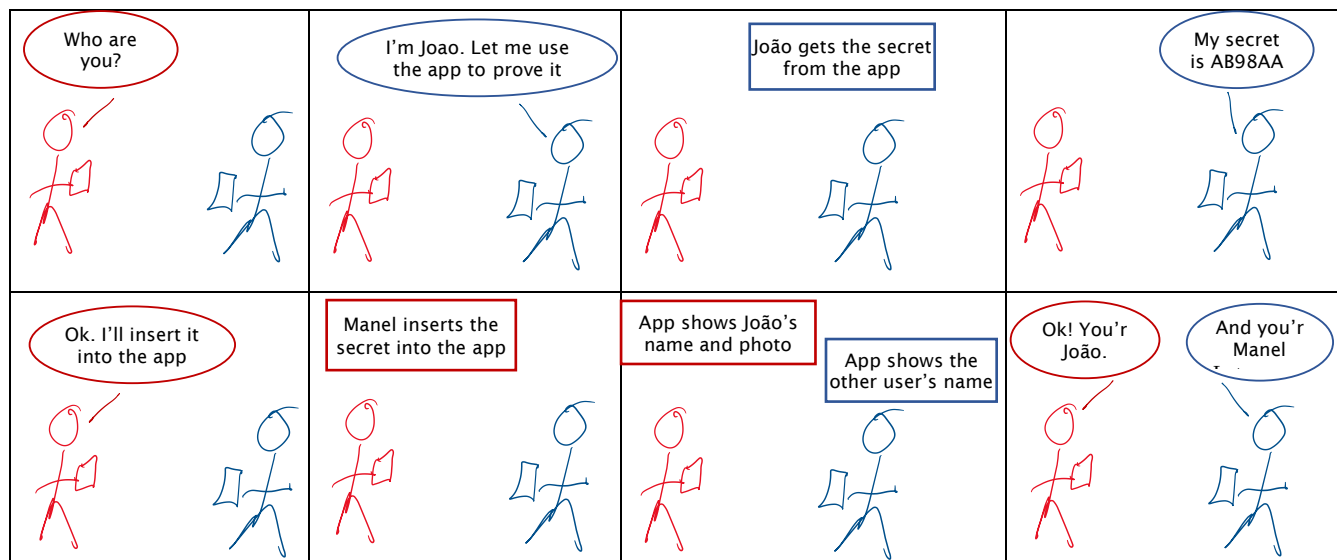
The page of the mobile application that will present the resources information will read a QR-Code (using a javascript widget) and contact the server using the REST API to fetch the corresponding information. The page will update itself, and without reloading will update the resource information on the screen.

6.2.2 User authentication

The previously defined pages should only be accessible after the user enters the FENIX istID and password. This should user the OAuth protocol, implemented on the server.

6.2.3 User validation

It is possible to use the developed application to show the user validation following these steps:



The secret used to identify a user should be random and generated every time it is used. Only authenticated users should access the pages that generate and validate the secret.

When somebody uses the secret to identify other user, the page where the other user got the secret will show the identity of the one that used it.

7 Extensibility

Although this implementation of the system only contains 3 microservices (corresponding to 3 types of resources) the system should be implemented so that it can be easily extended with new resource type/microservices.

8 Implementation

Students can use any technology and programming language (python, php, java, node.js) to develop the system. The selection the programming language will not affect the final grade.

9 Data persistence

In order to simplify implementation and future deployment in the cloud, students can use the simple serialization libraries to perform data persistence.

It is not necessary to use any DBMS.

10 Resource identifiers

Students should decide the format of the identifier of resources and whether they are transparent or not.

11 Evaluation

The detailed evaluation criteria will be posted later, but the evaluation will follow these generic criteria:

- Number of implemented functionalities
- Quality of the suitable resource/endpoint
- Extensibility
- Implementation of the backend
- Implementation of the Mobile application
- Report

12 Report

The students should write a report describing:

- The system architecture
- Implemented REST API
- Used technologies/libraries
- User interface of the system
- FENIX integration

A section should also describe how the proposed implementation could be deployed on the cloud: presenting the selected environment, its advantages and the necessary changes.

13 Submission date

Students should submit the projects before Christmas holiday (until 20th December 2019)