**ARQUITETURA**

**DE**

**SISTEMAS**

**DE**

**INTERNET**

1º semestre 2016/2017

PROJETO

João Beirão nº 75988

Inês Lourenço nº 75637

O trabalho proposto tinha como objetivo o desenvolvimento de uma Web Application para gerir a ocupação e disponibilidade das várias salas do Instituto Superior Técnico.

**Arquitetura do sistema**

O sistema está implementado na Cloud da Google, Google App Engine. A Datastore é a base de dados usada para o armazenamento dos dados. O ficheiro do servidor alojado na Cloud tem as classes que criam as várias APIs, e as que chamam a página html no browser para a interface com o utilizador, tendo-se optado por Python como linguagem de programação. A interação com a API do fénix é também aqui feita, através de pedidos REST. A interação com o browser é feita tanto através de pedidos HTTP como pedidos REST para o JavaScript, que chama as APIs previamente criadas e têm a grande vantagem de poderem ser chamadas por qualquer cliente, em diferentes tipos de aplicação.

A modularidade do sistema está garantida quando as classes responsáveis por criar a API no ficheiro do servidor se chamam umas às outras como meio de criar novas maneiras de interação com a API.

Foram implementadas duas estruturas de classes de dados: ‘Rooms’ – é uma classe que tem como key o id de cada room, e tem 3 campos fixos: O id, o name, o type; e mais um parâmetro adicional, o User. Cada um dos Users registados numa sala é uma instância da classe User, que tem dois campos: O user\_id, e o user\_name, em que a key é o id.

**REST API implementada**

A API implementada tem a vantagem de poder ser chamada por qualquer cliente em qualquer tipo de aplicação. Neste projeto, a API retorna os dados em JSON, e estes serão posteriormente chamados através de REST por código JavaScript implementado no código html.

Com base nas classes acima indicadas, foram implementadas

**Tecnologias/bibliotecas implementadas**

As bibliotecas utilizadas foram:

Jinja

App2

**Interface com o utilizador**

A página inicial chama-se *index.html*, e dá ao utilizador a possibilidade de fazer login na aplicação. Aí é redirecionado para uma página de login em que escreve o seu email, e através de uma checkbox pode escolher entrar como administrador ou como estudante.

Funcionalidades disponibilizadas:

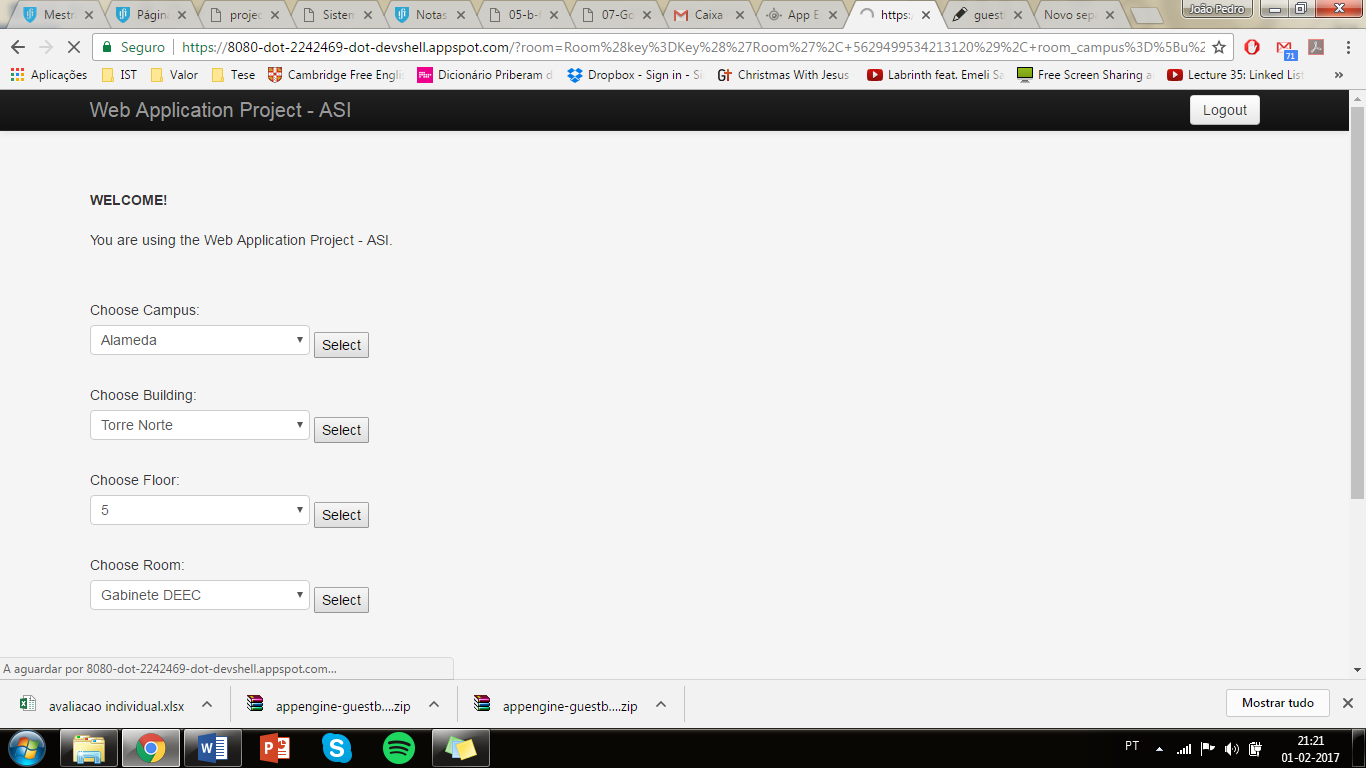
* Se entrar como administrador:

É redirecionado para a página admin.html. Aí, o Administrador pode:

- Consultar a lista de salas que já foram disponibilizadas, a sua contagem total, e a respetiva ocupação de cada uma uma (nº de estudantes com o check in feito nessa sala nesse instante) – vinda de um scrip em JavaScript, que vai buscar os dados à API criada

Disponibilizar uma nova sala. Para isto, o administrador tem que selecionar opções em 4 dropdown lists, respectivamente: o Campus, o Edifício, o Piso, e a Sala desejados. Assim que esta for adicionada, é de imediato adicionada à lista de salas disponíveis mostrada pelo JavaScript vindo da API, com a ocupação igual a zero.

Funcionalidade extra: Caso a nova sala que o administrador escolheu adicionar seja uma já anteriormente disponibilizada, então esta não é adicionada novamente.



* Se não entrar como administrador (assume-se que é um estudante):

É redirecionado para a página student.html:

Caso o estudante já esteja registado nalguma sala, pode:

- fazer check in noutra sala, através de uma dropdown list. Nesta situação, é apagado da base de dados da sala onde estava anteriormente, e adicionado à nova.

- fazer check out dessa sala, sendo apagado da base de dados dos utilizadores dessa sala. Nesta situação em que o estudante está logged in, mas não está checked in em nenhuma sala, tem 3 opções:

- Fazer check in numa sala

- Ver uma lista dos estudantes registados numa certa sala (através dos dados JSON retornados pela API criada)

- Encontrar a sala em que está localizado um certo estudante.

As duas últimas opções redirecionam o estudante para uma nova página html, chamada StudentList, em que são apresentados os resultados do respetivo query.

Em qualquer situação, o estudante e o administrador podem sempre fazer log out sendo reencaminhados para a página inicial, index.html.

