



UNIVERSIDADE DE ÉVORA

Engenharia Informática
Sistemas Operativos II

Trabalho Prático - 2^aFase

Ocupação de espaços comerciais

Professor: José Saias

Sarah Simon Luz 38116
Ana Ferro 39872

7 Julho, Ano Letivo 2019/2020

Índice

1	Introdução	2
2	Tecnologias Usadas	3
2.1	Spring Boot	3
2.2	Dependências	4
2.2.1	Spring Data Jpa e Postgres	4
2.2.2	Spring Security	5
2.2.3	Thymeleaf	5
2.2.4	Spring Web	5
3	Front-end	6
3.1	Menu	6
3.2	Sign up	7
3.3	Login	7
3.4	Listar Ocupações	8
3.5	Registar espaços	9
3.6	Consultar registos efetuados	10
3.7	Remover registos	11
3.8	Error	11
4	Como correr o projeto	12
5	Bibliografia	13

1 Introdução

Atualmente, as necessárias regras de distanciamento levam a uma gestão da concentração de pessoas em supermercados e outros espaços com grande afluência de cidadãos. Quando a concentração de pessoas por metro quadrado é superior a um determinado valor, é mesmo suspensa a entrada de mais clientes até que outros tenham saído. Foi pedido uma implementação de uma aplicação web com a solução:

1. Consultar o grau de ocupação de um espaço. Deve incluir a data do registo e um de quatro níveis:
 - i.vazio ou com mínima lotação
 - ii.com pessoas, mas espaço suficiente
 - iii.muito cheio
 - iv.muito cheio e com fila de espera

Se houver mais de um registo na última hora para esse espaço, deve indicar todos os níveis registados nesse período, com a respetiva frequência. Exemplo: muito cheio: 3 / com pessoas, mas espaço suficiente: 1

2. Registar o grau de ocupação de um espaço (indicando o espaço, nível da ocupação, data e hora do registo,e o login do utilizador)
3. Consultar registos efetuados (cada utilizador vê apenas os seus registos)
4. Remover um registo (próprio).

Os dados têm de ser guardados de forma persistente É pedi

É também pedido a implementação de uma componente adicional:

1. Uma interface amigável
2. Acrescentar coordenadas sobre cada espaço comercial.
3. Adicionar latitude e longitude (no formato Decimal Degrees) a cada espaço.
4. Na operação 1, acrescentar agora uma variante para mostrar os espaços no mapa.
5. Juntar uma 5^a operação, que a partir das coordenadas do utilizador (inseridas num form ou via mapa) encontra o espaço comercial mais próximo. Quaisquer parâmetros de configuração devem estar fora do código, sendo lidos de um ficheiro de propriedades.

2 Tecnologias Usadas

2.1 Spring Boot

O Spring Boot é uma ferramenta que visa facilitar o processo de configuração e publicação de aplicações que utilizam o ecossistema Spring, tornando possível termos uma aplicação rodando em produção rapidamente com o esforço mínimo de configuração e implementação.

O Spring Boot permite selecionarmos os módulos/dependências que desejamos utilizar, disponibilizados na página Spring Initializr que irá criar um projeto Maven ou Gradle pré-configurado com todos os componentes solicitados especificados .

O Spring Boot é adicionado à aplicação Gradel através de um plug-in adicionado no ficheiro build.gradle

```
plugins {  
    id 'org.springframework.boot' version '2.3.1.RELEASE'  
}
```

2.2 Dependências

Spring Boot também facilita a implementação de novas dependências com o plug-in: io.spring.dependency-management. As dependências usadas neste trabalho são as seguintes:

```
dependencies {
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
    implementation 'org.springframework.boot:spring-boot-starter-security'
    implementation 'org.springframework.boot:spring-boot-starter-thymeleaf'
    implementation 'org.springframework.boot:spring-boot-starter-web'
    runtimeOnly 'org.postgresql:postgresql'
    testImplementation('org.springframework.boot:spring-boot-starter-test') {
        exclude group: 'org.junit.vintage', module: 'junit-vintage-engine'
    }
    testImplementation 'org.springframework.security:spring-security-test'
}
```

2.2.1 Spring Data Jpa e Postgres

Java Persistence API (ou simplesmente JPA) é uma API padrão da linguagem Java que descreve uma interface comum para frameworks de persistência de dados. PostgreSQL é um sistema gerenciador de banco de dados objeto relacional (SGBD), desenvolvido como projeto de código aberto.

A nossa aplicação Spring Boot usa o Spring Data JPA que faz operações com o JPA repository na base de dados PostgreSQL.

Para conectar e localizar a instância PostgreSQL o Spring-Data usa `spring.datasource.properties`.

```
implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
runtimeOnly 'org.postgresql:postgresql'
```

2.2.2 Spring Security

Spring Security é uma estrutura de autenticação e controle de acesso poderosa e altamente personalizável.

Na nossa aplicação Web é usada conjuntamente com o JPA+Postgres para guardar novos utilizadores da aplicação e verificar as credenciais de utilizadores já registados.

```
implementation 'org.springframework.boot:spring-boot-starter-security'
```

2.2.3 Thymeleaf

Thymeleaf é um gerador de modelos Java para processar e criar HTML, XML, JavaScript, CSS. Em aplicações web, o *Thymeleaf* visa substituir completamente o JSP (JavaServer Pages).

No nosso trabalho, é o *Thymeleaf* que vai fazer a "ponte" entre o *back-end* e o *front-end* permitindo-nos aceder e introduzir dados no base de dados. Está encarregue das *views*.

```
implementation 'org.springframework.boot:spring-boot-starter-thymeleaf'
```

2.2.4 Spring Web

O Spring Web permite construir aplicações web usando Spring MVC.

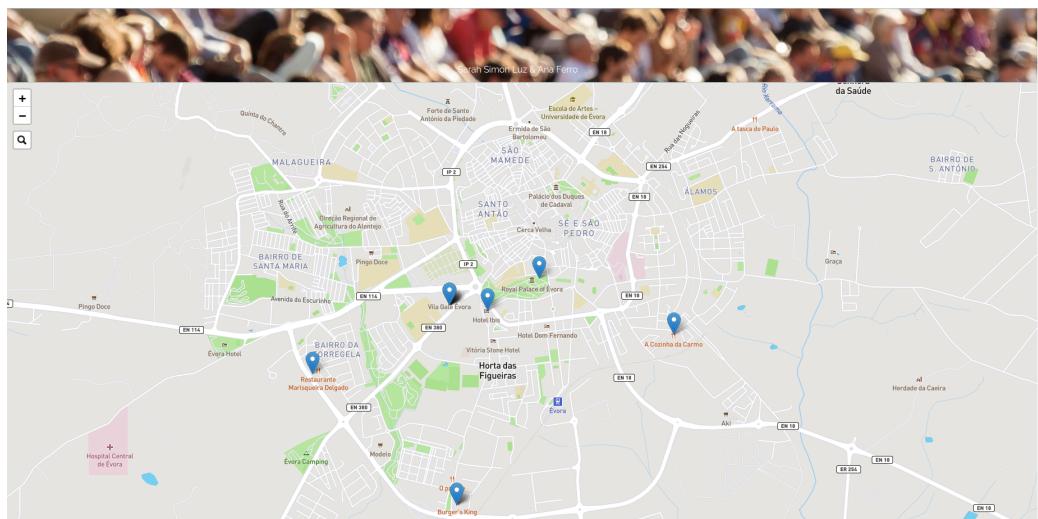
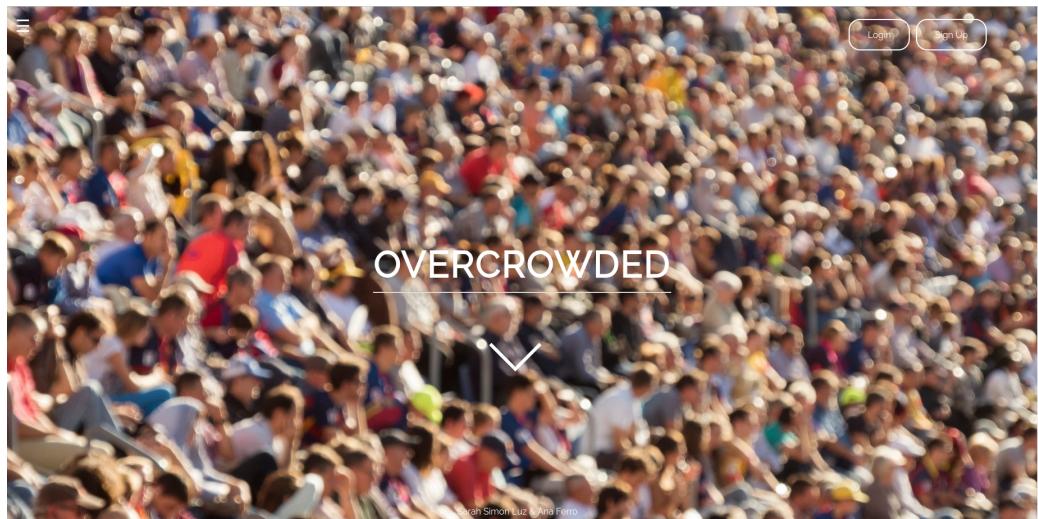
```
implementation 'org.springframework.boot:spring-boot-starter-web'
```

3 Front-end

3.1 Menu

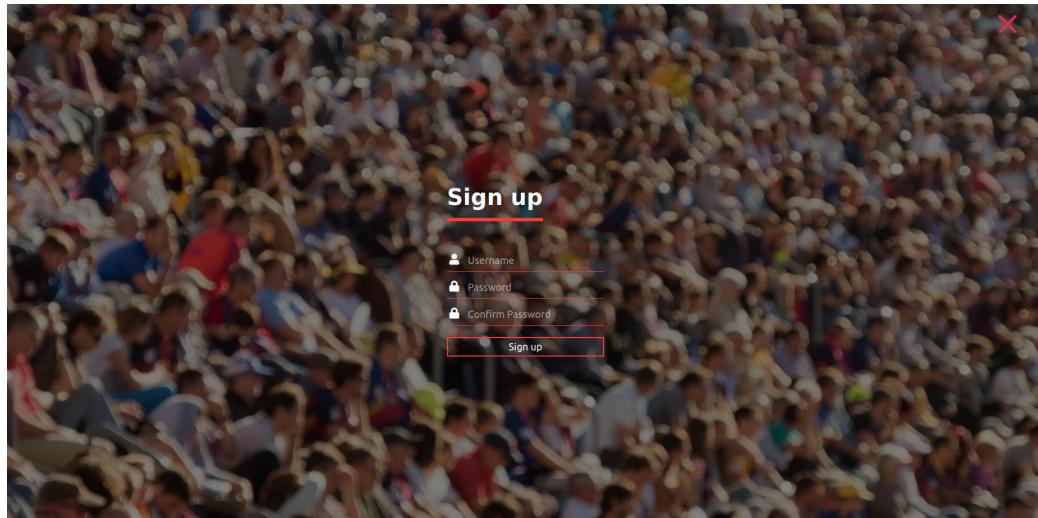
Esta é a página inicial, qualquer pessoa pode visitar esta página, apartir daqui podemos fazer login e sign up (botões lado direito) ou consultar o mapa (carregando na seta abaixo no título) com espaços já registados.

Para procurar basta pesquisar no *search bar* do mapa, que o vai direcionar para o local pesquisado. Caso não se encontre um marcador nessa local significa que o espaço ainda não foi registrado.



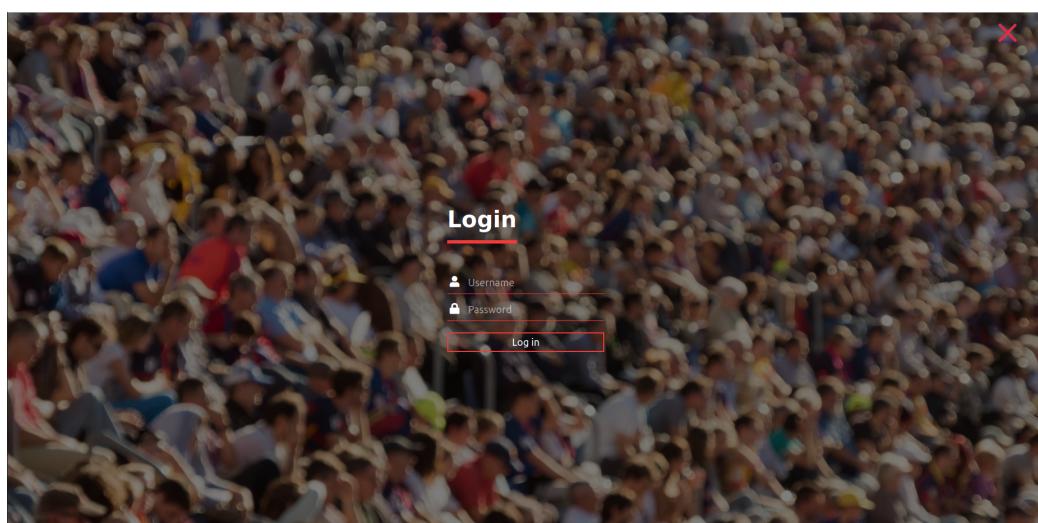
3.2 Sign up

Nesta página o *user* pode criar uma conta preenchendo os campos do *username*, *password* e verificação de *password*. Caso seja bem sucedido é reencaminhado para página *Listar Ocupações*



3.3 Login

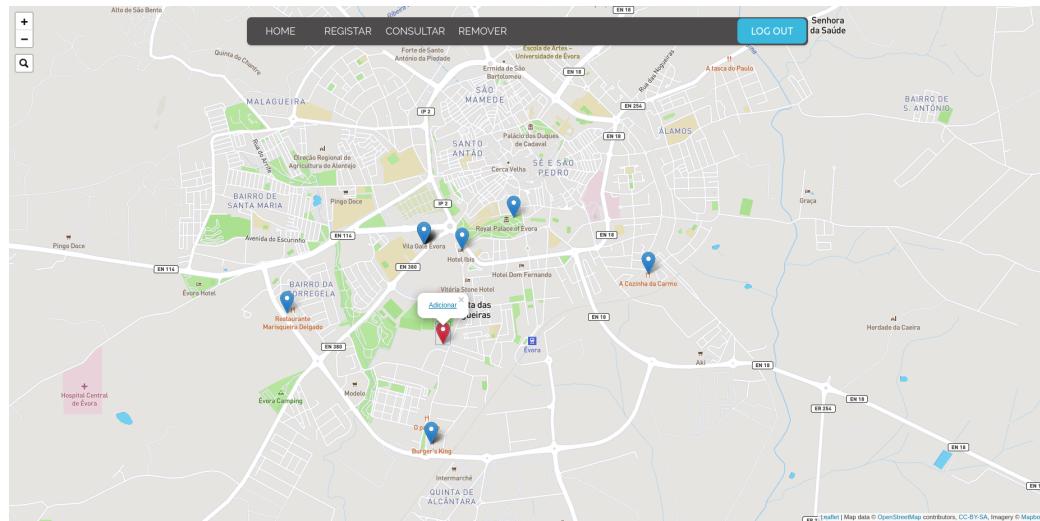
Nesta página o *user* pode entrar na sua conta (préviamente criada) introduzindo o *username* e *password* corretos. Caso esta operação seja bem sucedida, o *user* é reencaminhado para a página *Listar Ocupações*, caso contrário, aparece uma mensagem de erro e o *user* pode voltar a tentar entrar.



3.4 Listar Ocupações

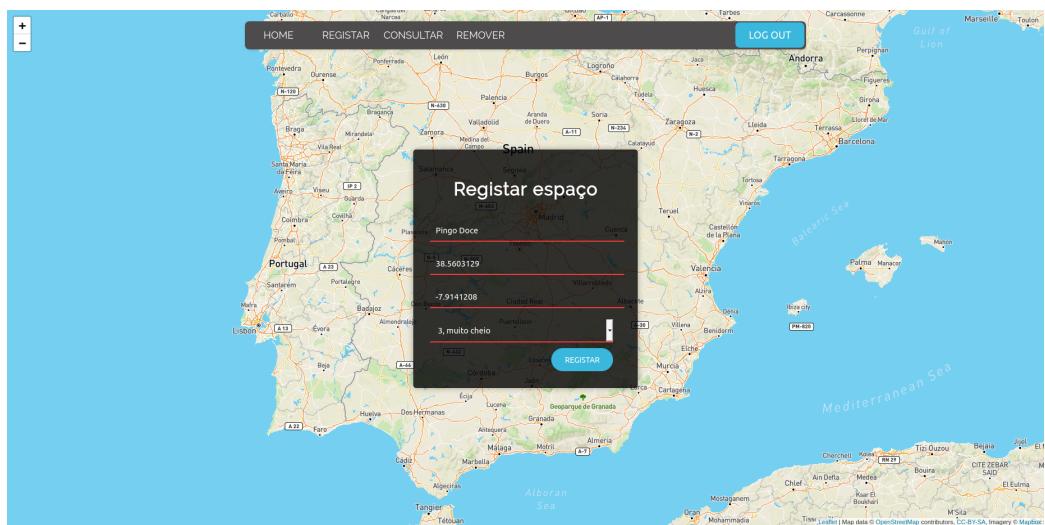
Desta página em diante apenas *users* com conta inicializada podem aceder. Aqui é disponibilizado um mapa com todos os espaços registados e respetivos graus de ocupação (*popup* que aparece quando se carrega num marcador).

Para pesquisar recorre-se ao *search bar* do mapa que vai colocar um marcador no local pesquisado , aí dará a opção de adicionar esse espaço aos registos reencaminhando para a página *Registar espaços*.



3.5 Registar espaços

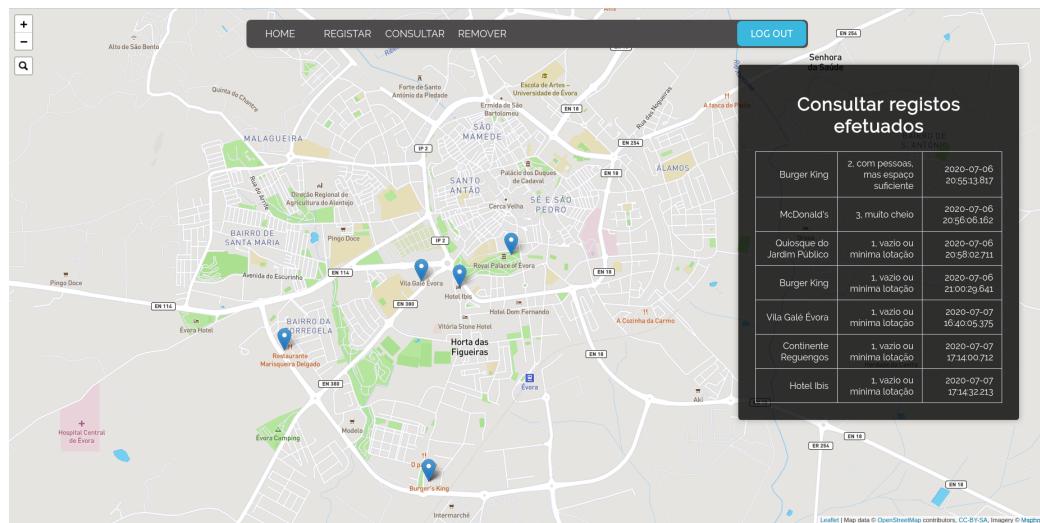
Aqui registamos o grau de ocupação de um espaço, indicando o nome do espaço, a latitude e longitude (parâmetros preenchidos automaticamente e provenientes da página *Listar Ocupações* ou da página *Consultar registos efetuados*) e o nível de ocupação. A cada registo está também associado uma data e hora do registo e um *user* que são contabilizados automaticamente sem ser necessário introduzir esses dados no formulário.



3.6 Consultar registos efetuados

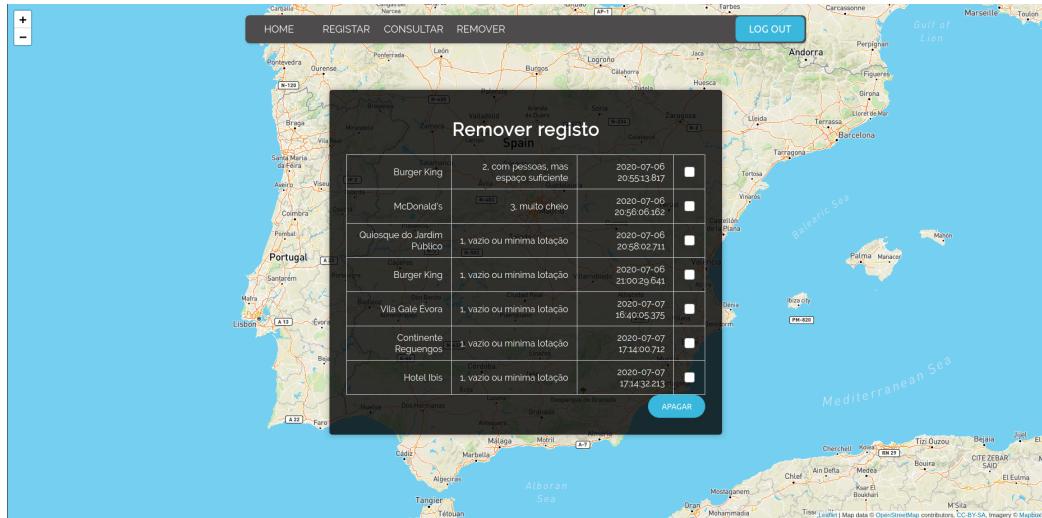
Nesta página, podemos consultar todos os registos efetuados pelo *user* atual. Estes dados são disponibilizados em formato de uma tabela com valores sobre o nome do espaço, o grau de ocupação e a data do registo. Estes espaços apresentam-se também registados no mapa.

É ainda possível, tal como na página *Listar Ocupações*, pesquisar no mapa e adicionar diretamente registos.



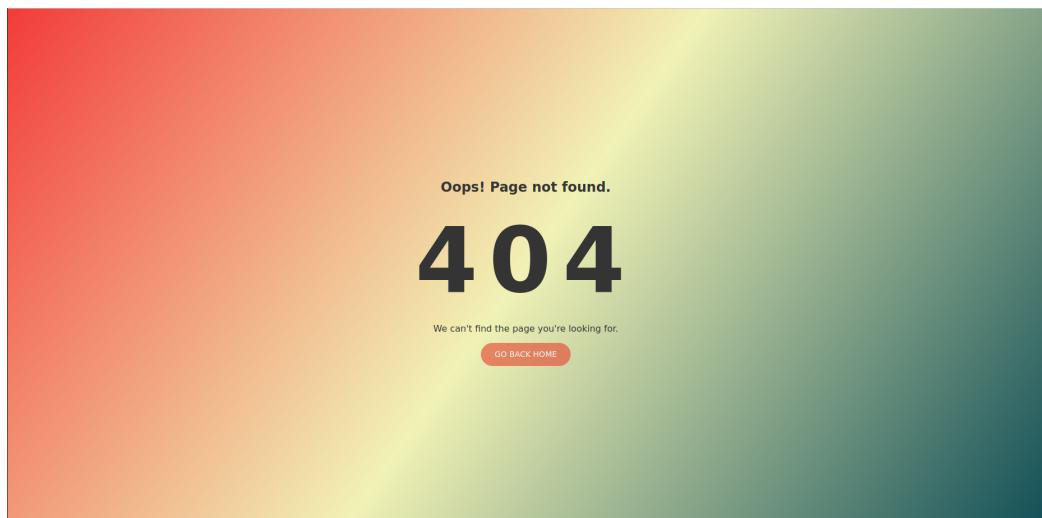
3.7 Remover registos

Nesta página, é possível apagar os registos efetuados pelo *user*. A informação é, uma vez mais, disponibilizada em formato de tabela e é dada a opção de selecionar os registos que queremos apagar.



3.8 Error

Esta página aparece quando tentamos aceder a uma página não existente.



4 Como correr o projeto

Estando na página do projeto, basta correr no terminal:

1. *sudo gradle compileJava*
2. *sudo gradle bootrun*

5 Bibliografia

1. Registration and Login with Spring Boot, Spring Security, Spring Data JPA, Hibernate, MySQL, JSP, Bootstrap and Docker Compose
https://hellokoding.com/registration-and-login-example-with-spring-security-spring-boot-spring-data-jpa-hsql-jsp/
7 julho, 20:41
2. Building an Application with Spring Boot
https://spring.io/guides/gs/spring-boot/
7 julho, 21:02