

# Universidade da Beira Interior

## Departamento de Informática



Departamento de  
Informática

### *Aplicação Web para Navegação na UBI Baseada em Tags Posicionadas*

Elaborado por:

**Rodrigo Manuel Teixeira Duarte**

Orientador:

**Professor Doutor Pedro R. M. Inácio**

Coorientador:

**Mestre João B. F. Sequeiros**

23 de junho de 2023



# ***Agradecimentos***

Prestes a concluir a licenciatura em Engenharia Informática, quero aproveitar esta oportunidade para agradecer a todos os que de alguma forma contribuíram para o sucesso do meu percurso académico e pessoal.

Primeiramente, agradeço aos meus pais por todo o apoio e incentivo que me deram ao longo da minha vida académica. Sem o apoio deles, não teria sido possível chegar onde cheguei.

Gostaria de agradecer aos meus amigos que me ajudaram a superar as dificuldades que enfrentei ao longo deste curso, deram força para continuar e que me ajudaram a relaxar.

Agradeço sinceramente os esforços sinceros e o tempo valioso dado pelo meu orientador Professor Doutor Pedro Inácio e pelo Professor Bernardo Sequeiros. As suas orientações e opiniões foram essenciais na conclusão deste projeto.



# Conteúdo

<b>Conteúdo</b>	<b>iii</b>
<b>Lista de Figuras</b>	<b>vii</b>
<b>Lista de Tabelas</b>	<b>ix</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Enquadramento . . . . .	1
1.2 Motivação . . . . .	2
1.3 Objetivos . . . . .	2
1.4 Organização do Documento . . . . .	3
<b>2 Estado da Arte e Tecnologias Utilizadas</b>	<b>5</b>
2.1 Introdução . . . . .	5
2.2 Estado da Arte . . . . .	5
2.2.1 Os Diferentes Tipos de Soluções . . . . .	5
2.2.2 Implementações Comerciais . . . . .	6
2.2.2.1 Pointr . . . . .	7
2.2.2.2 mappedin . . . . .	7
2.3 Tecnologias Utilizadas . . . . .	8
2.3.1 <i>Apache</i> . . . . .	8
2.3.2 <i>Certbot</i> . . . . .	8
2.3.3 <i>MySQL</i> . . . . .	8
2.3.4 <i>PHP</i> . . . . .	9
2.3.5 <i>HTML</i> . . . . .	9
2.3.6 <i>CSS</i> . . . . .	9
2.3.7 <i>JavaScript</i> . . . . .	9
2.3.8 <i>jQuery</i> . . . . .	9
2.3.9 <i>PHP QR Code</i> . . . . .	9
2.3.10 <i>reCAPTCHA</i> . . . . .	10
2.4 Conclusões . . . . .	10
<b>3 Engenharia de Software</b>	<b>11</b>

3.1	Introdução . . . . .	11
3.2	Análise de Requisitos . . . . .	11
3.2.1	Requisitos Funcionais . . . . .	11
3.2.2	Requisitos Não Funcionais . . . . .	12
3.3	Casos de Uso . . . . .	12
3.3.1	Descrição dos Casos de Uso . . . . .	12
3.4	Diagrama de Atividades . . . . .	15
3.5	Modelação da Base de Dados . . . . .	17
3.5.1	Modelo Conceptual . . . . .	17
3.5.2	Modelo Físico . . . . .	18
3.6	Arquitetura do Sistema . . . . .	18
3.7	Conclusões . . . . .	19
<b>4</b>	<b>Implementação</b>	<b>23</b>
4.1	Introdução . . . . .	23
4.2	Servidor . . . . .	23
4.3	Funcionalidades . . . . .	23
4.3.1	Autenticação . . . . .	24
4.3.2	Gestão de Localizações . . . . .	26
4.3.3	Gestão de Conexões . . . . .	28
4.3.4	<i>Scan</i> de Códigos QR . . . . .	31
4.3.5	Pesquisa de Localizações . . . . .	32
4.3.6	Calculo de Rotas . . . . .	34
4.4	Segurança . . . . .	34
4.4.1	<i>BCrypt</i> . . . . .	35
4.4.2	<i>ID Hashing</i> . . . . .	36
4.4.3	Proteção contra <i>SQL Injection</i> . . . . .	36
4.5	Conclusões . . . . .	36
<b>5</b>	<b>Manual de Instalações e Testes</b>	<b>37</b>
5.1	Introdução . . . . .	37
5.2	Manual de Instalação . . . . .	37
5.2.1	Instalação e Configuração do Servidor . . . . .	37
5.2.2	Configuração da Base de Dados . . . . .	40
5.3	Transferência da Aplicação . . . . .	40
5.4	Pacotes Adicionais do <i>PHP</i> . . . . .	41
5.4.1	Transferir o <i>PHP QR Code</i> . . . . .	41
5.4.2	Transferir o <i>reCAPTCHA</i> . . . . .	41
5.5	Testes . . . . .	41
5.6	Inquérito sobre a Aplicação . . . . .	43
5.7	Conclusões . . . . .	44

<b>CONTEÚDO</b>	<b>v</b>
<hr/>	
<b>6 Conclusões e Trabalho Futuro</b>	<b>47</b>
6.1 Conclusões Principais . . . . .	47
6.2 Trabalho Futuro . . . . .	48
<b>A Excerto de código da base de dados</b>	<b>49</b>
<b>Bibliografia</b>	<b>51</b>





## ***Lista de Figuras***

3.1	Diagrama de casos de uso do sistema. . . . .	14
3.2	Diagrama de atividades da autenticação do administrador. . . . .	16
3.3	Diagrama de atividades da página inicial do <i>backoffice</i> . . . . .	17
3.4	Diagrama de atividades da página de administração de utilizadores. . . . .	18
3.5	Diagrama de atividades da página de uma localização. . . . .	19
3.6	Modelo conceptual de dados usado na aplicação. . . . .	20
3.7	Modelo físico dos dados usado na aplicação. . . . .	20
3.8	Diagrama de componentes que demonstra a arquitetura do sistema. . . . .	21
4.1	Captura de ecrã demonstrativa da página de registo do primeiro administrador. . . . .	24
4.2	Captura de ecrã demonstrativa da página de <i>login</i> da aplicação desenvolvida. . . . .	25
4.3	Captura de ecrã demonstrativa da lista de administradores apresentada na página de administração da aplicação desenvolvida. . . . .	26
4.4	Captura de ecrã demonstrativa do formulário de criação de localizações. . . . .	27
4.5	Captura de ecrã demonstrativa da lista de localizações apresentada na página <i>dashboard</i> da aplicação desenvolvida. . . . .	28
4.6	Captura de ecrã demonstrativa do formulário de edição de localizações. . . . .	29
4.7	Captura de ecrã demonstrativa do código QR da localização, gerado pela aplicação. . . . .	29
4.8	Captura de ecrã demonstrativa do formulário de criação de conexões e da lista de conexões da localização. . . . .	30
4.9	Captura de ecrã demonstrativa do formulário de edição de uma conexão. . . . .	31
4.10	Captura de ecrã demonstrativa da página da localização após efetuar <i>scan</i> de um código QR. . . . .	32
4.11	Captura de ecrã demonstrativa da página da localização após efetuar uma pesquisa. . . . .	33
4.12	Captura de ecrã demonstrativa da pesquisa por "biblioteca" na barra de pesquisa da aplicação. . . . .	33

4.13	Captura de ecrã demonstrativa da secção de cálculo de rotas na página da localização. . . . .	34
4.14	Captura de ecrã demonstrativa do resultado do cálculo de rotas. . .	35
5.1	Resultados do inquérito realizado à aplicação <i>web</i> . . . . .	45

## ***Lista de Tabelas***

3.1	Requisitos funcionais da aplicação. . . . .	12
3.2	Requisitos não funcionais da aplicação. . . . .	13
5.1	Testes realizados à aplicação. . . . .	42
5.2	Resultados dos testes realizados à aplicação. . . . .	43



## ***Lista de Excertos de Código***

4.1	Exemplo de uma <i>query</i> protegida contra <i>SQL Injection</i> . . . . .	36
5.1	Instalação dos pacotes necessários para o servidor. . . . .	37
5.2	Verificação do serviço do <i>Apache2</i> . . . . .	38
5.3	Iniciar o serviço do <i>Apache2</i> . . . . .	38
5.4	Configuração da <i>firewall</i> . . . . .	38
5.5	Edição do ficheiro <i>dir.conf</i> . . . . .	38
5.6	Conteúdo do ficheiro <i>dir.conf</i> . . . . .	38
5.7	Reiniciar o serviço do <i>Apache2</i> . . . . .	38
5.8	Conteúdo do ficheiro <i>info.php</i> . . . . .	40
5.9	Instalação da base de dados <i>MySQL</i> . . . . .	40
5.10	Criação do utilizador e da base de dados. . . . .	40
5.11	Transferência da aplicação <i>web</i> . . . . .	40
5.12	Alteração das permissões da pasta <i>/var/www/html</i> . . . . .	41
A.1	Criação das tabelas na base de dados. . . . .	49



# ***Acrónimos***

<b>API</b>	<i>Application Programming Interface</i>
<b>BLE</b>	<i>Bluetooth<sup>®</sup> Low Energy</i>
<b>CSS</b>	<i>Cascading Style Sheets</i>
<b>EI</b>	Engenharia Informática
<b>GPS</b>	<i>Global Positioning System</i>
<b>HTML</b>	<i>Hypertext Markup Language</i>
<b>HTTPS</b>	<i>Hypertext Transfer Protocol Secure</i>
<b>IA</b>	Inteligência Artificial
<b>LAMP</b>	<i>Linux, Apache, MySQL, PHP</i>
<b>MySQL</b>	<i>My Structured Query Language</i>
<b>NFC</b>	<i>Near Field Communication</i>
<b>PHP</b>	<i>Hypertext Preprocessor</i>
<b>QR</b>	<i>Quick Response</i>
<b>RA</b>	Realidade Aumentada
<b>RFID</b>	<i>Radio-Frequency Identification</i>
<b>SQL</b>	<i>Structured Query Language</i>
<b>UBI</b>	Universidade da Beira Interior





## **Capítulo**

# 1

## **Introdução**

Este relatório foi elaborado no âmbito da unidade curricular de Projeto, da Licenciatura em Engenharia Informática da Universidade da Beira Interior (UBI). Este capítulo apresenta o projeto, o enquadramento para o mesmo, a motivação para a sua escolha, os seus objetivos e a respetiva organização do documento.

### **1.1 Enquadramento**

Os utensílios de auxílio à navegação no espaço são de grande utilidade aos humanos e, apesar de toda a tecnologia atual associada à navegação, ainda há muitos desafios a resolver, por exemplo, quando se precisa de um posicionamento mais preciso, como o caso de uso da orientação dentro de edifícios.

Atualmente vivemos na era da informação e, como tal, existem várias ferramentas que nos permitem partilhar essa informação de forma simples e eficaz. Uma dessas ferramentas é a Internet, que nos permite aceder a informação de qualquer lugar, e que nos permite partilhar essa informação com outras pessoas.

Na facilitação da partilha de informação, existem várias tecnologias que permitem a partilha de informação de forma simples e eficaz. As tecnologias *Near Field Communication* (NFC) e *Quick Response* (QR) são exemplos de tecnologias que permitem efetuar essa partilha de informação.

A tecnologia NFC é uma tecnologia de comunicação sem fios que permite a troca de dados entre dispositivos próximos uns dos outros. Esta tecnologia é baseada na troca de dados entre dispositivos mediante campos magnéticos e eletromagnéticos.

A tecnologia QR é uma tecnologia de código de barras bidimensional que permite a leitura de informação via um leitor de código de barras. Este código de barras pode ser lido por um dispositivo móvel, desde que este tenha uma câmara fotográfica e uma aplicação que permita a leitura deste código.

Este projeto tem como foco o uso destas duas tecnologias para a partilha de informação, em particular, a partilha de informação sobre a localização de um determinado ponto de interesse, em conjunto com um servidor *web* que terá como objetivo a distribuição e gestão da informação disponibilizada.

## 1.2 Motivação

A escolha deste projeto deve-se ao interesse em desenvolver aplicações que possam ser utilizadas no dia a dia, e que sejam capazes de melhorar a qualidade de vida dos alunos, visitantes e funcionários da UBI. Sendo esta uma instituição de ensino superior que está sempre em constante evolução acredita-se que a aplicação desenvolvida possa ser utilizada por alunos e funcionários da instituição, especialmente aqueles que se juntaram recentemente à instituição.

A principal motivação para a realização deste projeto é de utilizar tecnologias anteriormente mencionadas para facilitar a navegação entre os edifícios e pontos de interesse da UBI de forma simples e eficaz.

## 1.3 Objetivos

O objetivo principal deste trabalho é o de desenvolver uma aplicação *web*, orientada para dispositivos móveis, que concretize uma solução de navegação simples, mas eficaz para os edifícios da UBI. Esta aplicação deve ser capaz de fornecer ao utilizador uma descrição textual e visual da sua localização, bem como a indicação da direção para o destino pretendido.

Para atingir estes objetivos, foi definido o seguinte conjunto de tarefas:

**Tarefa 1** Contextualização com os objetivos propostos; preparação do ambiente de trabalho, e seleção e familiarização com as tecnologias a utilizar (0,5 meses);

**Tarefa 2** Desenho do modelo de dados; desenvolvimento do primeiro protótipo com funcionalidades de mostrar o posicionamento após leitura de um *tag* NFC (0.5 meses);

**Tarefa 3** Desenvolvimento do *backoffice* para edição das informações na base de dados após autenticação (0,5 meses);

**Tarefa 4** Desenvolvimento de funcionalidades mais avançadas relacionadas com a elaboração de percursos e formas eficientes de guiar o utilizador (1 mês);

**Tarefa 5** Finalização da aplicação. Testes e aprimoramento (0,5 meses);

**Tarefa 6** Escrita do relatório de projeto (0,5 meses).

## 1.4 Organização do Documento

De modo a refletir o trabalho feito, este documento encontra-se estruturado da seguinte forma:

1. O primeiro capítulo – **Introdução** – apresenta o projeto, a motivação para a sua escolha, o enquadramento para o mesmo, os seus objetivos e a respetiva organização do documento.
2. O segundo capítulo – **Estado da Arte e Tecnologias Utilizadas** – descreve os conceitos-chave no âmbito deste projeto, bem como as tecnologias utilizadas durante o desenvolvimento da aplicação *web*.
3. O terceiro capítulo – **Engenharia de Software** – descreve o processo de desenvolvimento da aplicação, desde a sua conceção até à sua implementação.
4. O quarto capítulo – **Implementação** – descreve como as funcionalidades da aplicação foram implementadas e como estas funcionam.
5. O quinto capítulo – **Manual de Instalação e Testes** – apresenta a instalação dos componentes necessários para a execução da aplicação, bem como os testes realizados.
6. O sexto capítulo – **Conclusões e Trabalho Futuro** – apresenta as conclusões obtidas com o desenvolvimento da aplicação, assim como as possíveis melhorias a serem implementadas.



## **Capítulo**

# 2

## ***Estado da Arte e Tecnologias Utilizadas***

### **2.1 Introdução**

Este capítulo apresenta o estado da arte relativo ao tema abordado no trabalho na secção 2.2 , bem como as tecnologias utilizadas para o desenvolvimento do projeto na secção 2.3.

### **2.2 Estado da Arte**

Esta secção encontra-se dividida em duas subsecções, a primeira subsecção 2.2.1 apresenta os diferentes tipos de soluções existentes para a navegação em espaços interiores, e a segunda subsecção 2.2.2 apresenta duas implementações comerciais.

#### **2.2.1 Os Diferentes Tipos de Soluções**

No que diz respeito a localização, o GPS é a solução óbvia, mas quando necessitamos dessa tecnologia no interior de um edifício, devido às frequências de rádio não trespassarem as paredes, não devolve os resultados esperados. Para Huang e Gartner (2010) [1] existem quatro aspetos principais a serem considerados no desenvolvimento de um sistema de localizações: a determinação da posição, a comunicação da informação ao utilizador, a adaptação dependendo do ambiente e outras variáveis como o acesso à Internet.

Uma solução estudada por Huang e Gartner (2010) é o sistema denominado “Chloe@University” onde se utiliza Realidade Aumentada (RA) para apre-

sentar a informação ao utilizador. Para conseguirem um alto nível de precisão, utilizaram várias tecnologias como uma rede de sensores interligados, *WiFi* e *Radio-Frequency Identification* (RFID).

Na implementação de RA, decidiram fazer uma solução para o lado do cliente onde se faz um armazenamento local (*cache*) de informação e uma aplicação que mais tarde irão ser usadas para calcular os destinos para onde o utilizador deseje ir. Existe um lado positivo e outro negativo nesta solução. O lado positivo é que o utilizador não necessita de estar sempre ligado à Internet, e o lado negativo é que o utilizador necessita de fazer os cálculos todos no seu dispositivo, o que requer mais poder de processamento e, como consequência, maior consumo de bateria.

Outra implementação demonstrada foi a “Smart Signs”, onde a visão principal é colocar sinais em localizações pré-definidas e que podem ser interagidos pelo utilizador. Estes sinais podem ser usados para apresentar informação sobre o local onde o utilizador se encontra ou para apresentar informação sobre o local para onde o utilizador se dirige. Não necessita de qualquer equipamento ou acesso à Internet, visto que simplesmente tem que seguir os sinais espalhados pelo edifício.

Um dos problemas evidentes desta implementação é que para cada localização será necessário um sinal diferente, o que fará aumentar os custos muito rapidamente. Pelo lado positivo, o cálculo da rota feito por este sistema tem em consideração o estado do tempo, preferências do utilizador, tempo que demora, restrições de mobilidade, etc., além disso, os sinais são algo que qualquer pessoa entende e não necessita de lembrar onde está e o que falta visto que vai a seguir as direções ao longo do percurso.

No final da análise, Huang e Gartner (2010) concluíram que os sistemas de navegação em espaços interiores tem vários desafios que precisam de ser explorados. Estes desafios encontram-se divididos em cinco categorias: o **posicionamento interno** (em que necessita de ser robusto e preciso), a **consciência do contexto** (adaptando-se às várias variáveis existentes), a **comunicação das rotas** (de forma textual, verbal ou visual e usando pontos de referência), a **combinação do exterior com o interior** (em que a transição deve ser o mais suave possível) e a **integração com outras tecnologias** (maioritariamente aproveitando dispositivos/sensores já existentes para tornar o ambiente mais inteligente).

### 2.2.2 Implementações Comerciais

Como referido na subsecção 2.2, existem várias implementações de sistemas de navegação em espaços interiores. Nesta subsecção, são apresentadas duas

implementações comerciais, a **Pointr** e a **mappedin**, nas secções 2.2.2.1 e 2.2.2.2, respetivamente.

### 2.2.2.1 Pointr

Começando pela **Pointr** [2], esta empresa oferece uma solução de navegação em espaços interiores que permite a navegação sem a necessidade de *hardware* adicional ao utilizador. Esta solução, denominada *Deep Location*<sup>®</sup> é baseada em *software* e utiliza a tecnologia *Bluetooth*<sup>®</sup> *Low Energy* (BLE) para a navegação. A **Pointr** oferece também uma solução de *analytics* que permite a análise de dados de navegação em espaços interiores.

A tecnologia BLE, como o nome indica, foca-se no uso de emissores de *Bluetooth*<sup>®</sup> de pequenas dimensões que comunicam com *tablets* e *smartphones*. Esta empresa não constrói estes emissores, mas estes têm que ser comprados separadamente e instalados em pontos de acesso *WiFi* ou na infraestrutura de iluminação já existente. Para além do uso do BLE, também é possível utilizar pontos de acesso *WiFi* e iluminação inteligente para usufruir do sistema de navegação em espaços interiores da **Pointr**.

Mais recentemente, com o avanço da Inteligência Artificial (IA), a **Pointr** lançou uma nova ferramenta que mapeia automaticamente espaços interiores. Esta ferramenta, denominada *MapScale*<sup>®</sup>, utiliza IA para mapear espaços interiores de forma automática, precisa e rápida. Também consegue calcular rotas eficientemente e possui uma *Application Programming Interface* (API) muito flexível que permite a integração com outras aplicações.

### 2.2.2.2 mappedin

Outra solução analisada foi a da empresa **mappedin** [3], em que estes dividem a sua solução em duas partes: **mapeamento interno** e **posicionamento interno**. Para o mapeamento interno, a **mappedin** utiliza uma ferramenta que permite a criação de mapas interiores de forma rápida e fácil. Esta ferramenta, denominada *Map Editor*, consegue delimitar a geometria do espaço, sincronizar com um gestor de conteúdo para fornecer a informação mais atualizada, planear a alteração de espaço para eventuais eventos e possui uma secção dedicada à análise de dados gerados pelos utilizadores.

Para o posicionamento interno, a **mappedin** segmenta em soluções sem infraestrutura, que utilizam *software* e *hardware* já existente, e soluções com infraestrutura, que necessitam de *hardware* adicional.

As soluções sem infraestrutura aproveitam os *smartphones* dos utilizadores e as redes *WiFi* existentes para calcular em aproximadamente a posição. A tecnologia predileta para esta solução é fornecida pela **Apple** que utiliza os

padrões dos sinais *WiFi* para conseguir precisão similar ao *Global Positioning System* (GPS), aproveitando-se dos padrões de frequências de rádio emitidas por cada ponto de acesso *WiFi*. Isto significa um baixo custo de implementação e manutenção, mas necessita de mapas corretos dos locais internos e de uma sondagem de frequências de rádio dos edifícios [4].

Por outro lado, o das soluções com infraestrutura, a **mappedin** utiliza a tecnologia BLE e pontos de acesso *WiFi* da **Cisco** e **Aruba** para calcular a posição do utilizador. Esta solução é mais precisa que a anterior, mas tem um custo de implementação e manutenção mais elevado. Esta junção de tecnologias denomina-se *sensor fusion* fornecida por várias empresas como a **IndoorAtlas**, **Inpixon**, **Senion** e **Here Technologies**.

## 2.3 Tecnologias Utilizadas

Esta secção apresenta, sucintamente, as tecnologias utilizadas no desenvolvimento do projeto e como estas foram integradas no mesmo. A maioria destas tecnologias são padrão no que diz respeito ao desenvolvimento *web*, no entanto, foram integradas bibliotecas externas como o *PHP QR Code* e *reCAPTCHA* e também uma aplicação para gerir os certificados *SSL* do *website*, denominada *Certbot*.

### 2.3.1 Apache

O *Apache* [5] é um servidor *web* bastante popular responsável pela distribuição de *websites* e conteúdos diversos. A escolha deste servidor deve-se ao facto de ser gratuito, *open source* e de fácil uso e configuração.

### 2.3.2 Certbot

O *Certbot* [6] é um *software* que permite a obtenção de certificados *SSL* gratuitos e de fácil instalação. Este *software* foi utilizado para a obtenção de um certificado *SSL* para o *website* do projeto. O certificado utilizado é um certificado *Let's Encrypt*, que é renovado automaticamente pelo *software*.

### 2.3.3 MySQL

O *MySQL* [7] é um *software* de gestão de bases de dados *relacionais* que permite a criação, gestão e manutenção de bases de dados. Este *software* foi utilizado para a criação de uma base de dados que armazena os dados relativos



às localizações, as suas conexões, fotografias e as contas de administradores do sistema.

#### 2.3.4 *PHP*

O *PHP* [8] é uma linguagem de *scripting open source* que permite a criação de *websites* dinâmicos. Este *software* foi utilizado para a criação do *website* do projeto, bem como para a criação de *scripts* que permitem a comunicação com a base de dados.

#### 2.3.5 *HTML*

O *HTML* [9] é uma linguagem de marcação que permite a criação de *websites* estáticos. Esta linguagem foi utilizada para a criação das várias páginas do *website* do projeto.

#### 2.3.6 *CSS*

O *CSS* [10] é uma linguagem de estilo que permite a estilização de *websites*. Esta linguagem foi utilizada para a criação dos estilos das várias páginas do *website* do projeto para este se tornar mais apelativo.

#### 2.3.7 *JavaScript*

O *JavaScript* [11] é uma linguagem de programação que permite a criação de *scripts* a operar sobre elementos presentes nas páginas. Esta linguagem foi utilizada para a criação de *scripts* que permitem a incorporação de funcionalidades no *website* do projeto, como, por exemplo, o reCAPTCHA 2.3.10.

#### 2.3.8 *jQuery*

O *jQuery* [12] é uma biblioteca de *JavaScript* que permite facilitar como operamos sobre os elementos. Esta biblioteca foi utilizada para a criação de *scripts* que permitem a incorporação de animações e funcionalidades no *website* do projeto.

#### 2.3.9 *PHP QR Code*

O *PHP QR Code* [13] é uma biblioteca de *PHP* que permite a criação de códigos *QR*. Esta biblioteca foi utilizada para a criação de códigos *QR* que permitem a leitura de dados via leitor de códigos *QR* ou telemóvel.

### 2.3.10 *reCAPTCHA*

O *reCAPTCHA* [14] é um *software* que permite a incorporação de um *captcha* no *website* do projeto. Este *software* foi utilizado para a incorporação de um *captcha* na página de registo de novos administradores numa tentativa de prevenir o *spam* de criação de contas.

## 2.4 Conclusões

Neste capítulo foram apresentados os conceitos teóricos necessários para a compreensão do projeto. Foram apresentados os conceitos de localização, navegação e os vários tipos de sistemas de navegação existentes. Foram também apresentadas duas implementações comerciais de sistemas de navegação em espaços interiores. Por fim, foram apresentadas as tecnologias utilizadas no desenvolvimento do projeto e como estas foram integradas no mesmo.

## Capítulo

# 3

## ***Engenharia de Software***

### **3.1 Introdução**

Este capítulo expõe toda a engenharia de *software* do projeto. Começa por apresentar os requisitos funcionais e não funcionais do sistema na secção 3.2, seguido de uma descrição dos casos de uso na secção 3.3 e do diagrama de atividades na secção 3.4. Também, apresenta-se a modelação da base de dados na secção 3.5 e, por fim, a arquitetura do sistema na secção 3.6.

### **3.2 Análise de Requisitos**

Nesta secção será abordada a análise de requisitos do sistema. Serão apresentados os requisitos funcionais (subsecção 3.2.1) e não funcionais do sistema (subsecção 3.2.2), considerados essenciais para o desenvolvimento do projeto.

#### **3.2.1 Requisitos Funcionais**

Os requisitos funcionais definem como o sistema deve funcionar, ou seja, as funcionalidades que o sistema deve ter para responder às necessidades e expectativas do utilizador. Para tal, foram identificados diversos requisitos que, podendo ser comuns entre aplicações *web*, são essenciais para a operacionalidade e qualidade do sistema. Os requisitos funcionais estão apresentados na tabela, 3.1.

ID	Requisito Funcional
1	O sistema deve incorporar um <i>backoffice</i> .
2	O sistema deve permitir o registo de administradores.
3	O sistema deve permitir a autenticação de administradores.
4	O sistema deve permitir o registo, a edição e remoção de localizações.
5	O sistema deve permitir o registo, a edição e remoção de conexões.
6	O sistema deve permitir a adição de imagens às localizações e conexões.
7	O sistema deve permitir a procura de localizações por nome.
8	O sistema deve também permitir a procura de localizações pelo que estas oferecem.
9	O sistema deve permitir o cálculo de uma rota entre duas localizações.
10	O sistema deve gerar códigos QR.

Tabela 3.1: Requisitos funcionais da aplicação.

### 3.2.2 Requisitos Não Funcionais

Os requisitos não funcionais são as limitações impostas ao sistema e que não estão diretamente relacionadas com as funcionalidades do mesmo. Estes requisitos são essenciais para a qualidade do sistema e para garantir que todos os requisitos funcionais são cumpridos com sucesso. Os requisitos não funcionais estão apresentados na tabela, 3.2.

## 3.3 Casos de Uso

Nesta secção serão apresentados os casos de uso do sistema. Estes casos de uso são apresentados via diagramas de casos de uso, que permitem uma melhor visualização das funcionalidades do sistema.

O diagrama de caso de uso mostra os atores envolvidos no sistema, as ações que eles podem realizar e as relações entre essas ações / atores. O diagrama de caso de uso do sistema está representado na figura 3.1.

### 3.3.1 Descrição dos Casos de Uso

Nesta subsecção serão apresentadas as descrições dos casos de uso do sistema apresentados na figura 3.1, estes podem ser descritos da seguinte forma:

ID	Requisito Não Funcional
1	O sistema deve ser desenvolvido em <i>Hypertext Preprocessor</i> (PHP), <i>Hypertext Markup Language</i> (HTML), <i>Cascading Style Sheets</i> (CSS) e <i>My Structured Query Language</i> (MySQL).
2	O sistema deverá ter os dados cifrados para melhorar a segurança dos dados. Principalmente as <i>passwords</i> dos administradores.
3	O sistema deve ser responsivo.
4	O sistema deve ser o mais intuitivo possível.
5	O sistema deve adaptar-se a qualquer tipo de <i>browser</i> e resoluções de dispositivos.
6	O sistema deverá permitir aos utilizadores encontrarem as localizações via uma barra de pesquisa.
7	O sistema deverá estar sempre disponível 24 horas, 7 dias por semana.
8	O sistema deverá gerar os <i>IDs</i> em forma de <i>hash</i> com 64 caracteres.

Tabela 3.2: Requisitos não funcionais da aplicação.

- **Login** — O **administrador** necessitam de se autenticar com um *username* e *password* para aceder ao *backoffice* do sistema. No caso de dados errados o sistema não permite tal acesso;
- **Fazer Registo** — Para aceder ao *backoffice* do sistema, o **administrador** necessita de se registar com um *username* e *password*. No caso de já existir um *username* igual, o sistema não permite o registo. É de salientar que a conta do **administrador** só fica ativa após o **administrador** ser aceite por um dos **administradores** já registados e confirmados;
- **Aceitar Registo** — O **administrador** pode aceitar o registo de um novo **administrador** no sistema.
- **Criar Localização** — O **administrador** pode criar uma localização no sistema. Para tal, o **administrador** necessita de preencher o nome da localização, a descrição da localização e adicionar imagens da localização;
- **Editar Localização** — O **administrador** pode editar uma localização já existente no sistema. Pode alterar todos os componentes ou apenas os necessários. É a partir desta ação que se consegue aceder à criação / edição de conexões entre localizações;

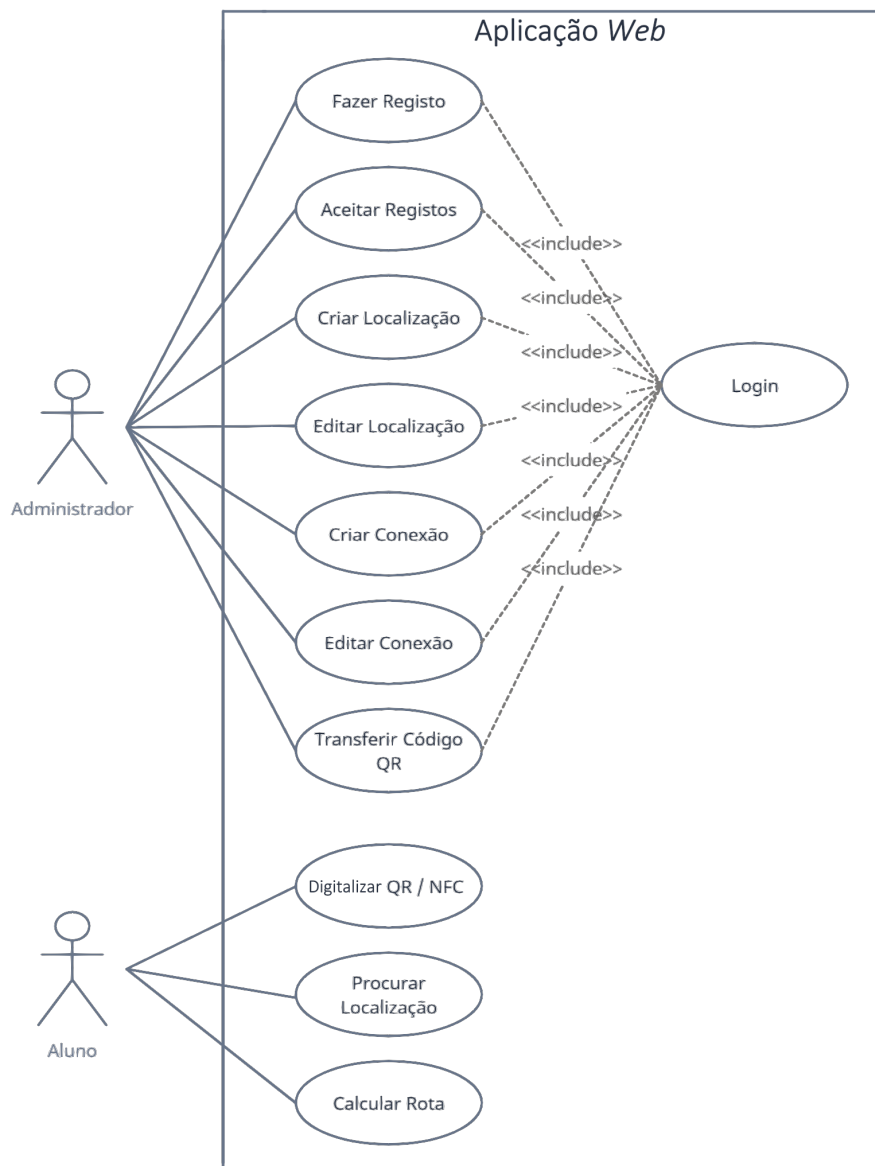


Figura 3.1: Diagrama de casos de uso do sistema.

- **Criar Conexão** — Quando do **administrador** pretende ligar duas localizações, este necessita de criar uma conexão entre as mesmas. Como já referido anteriormente, esta ação é realizada na edição de uma localização, onde existe uma secção para juntar duas localizações e colocar a distância entre elas, bem como, a descrição da conexão;
- **Editar Conexão** — O **administrador** pode editar uma conexão já exis-

tente no sistema. Pode alterar todos os componentes ou apenas os necessários;

- **Transferir Código QR** — Para o **administrador** poder imprimir o código QR de uma localização, este necessita de transferir o código QR da mesma. Esta ação é realizada na edição de uma localização, onde existe um botão para transferir o código QR;
- **Scan QR / NFC** — A forma como o **utilizador** vai aceder à informação da localização é através do *scan* do código QR ou NFC. Esta ação é realizada no dispositivo móvel do utilizador e redirecioná-lo-á para a página da localização;
- **Procurar Localização** — O **utilizador** pode procurar uma localização via uma barra de pesquisa. Esta ação é realizada em qualquer página do sistema. Este sistema utiliza o nome e descrição da localização para realizar a pesquisa;
- **Calcular Rota** — Na página de uma localização, o **utilizador** pode calcular a rota para outra localização. Para tal, o **utilizador** necessita de selecionar a localização para onde pretende ir e o sistema irá calcular a rota mais curta para lá chegar. Também poderá fazer *scan* uma localização, procurar pelo destino e calcular essa rota.

### 3.4 Diagrama de Atividades

Neste capítulo serão apresentados os diagramas de atividades relativos à aplicação *web*. Estes diagramas são importantes para entender os processos que ocorrem em cada uma das especificações e como diferentes componentes do sistema interagem para realizar tarefas específicas. É importante analisá-los com atenção para entender o funcionamento detalhado de cada uma das especificações.

A autenticação é um processo que permite verificar a identidade de um administrador. Este processo é realizado via um *username* e *password*. Também é incluído o processo de registo por este ser necessário para o administrador efetuar a autenticação futuramente. O diagrama de atividades da autenticação está representado na figura 3.2.

Após autenticação, o administrador será redirecionado para a página inicial do *backoffice*. Nesta página, o administrador pode criar, editar e eliminar localizações. O diagrama de atividade da página inicial do *backoffice* está representado na figura 3.3.

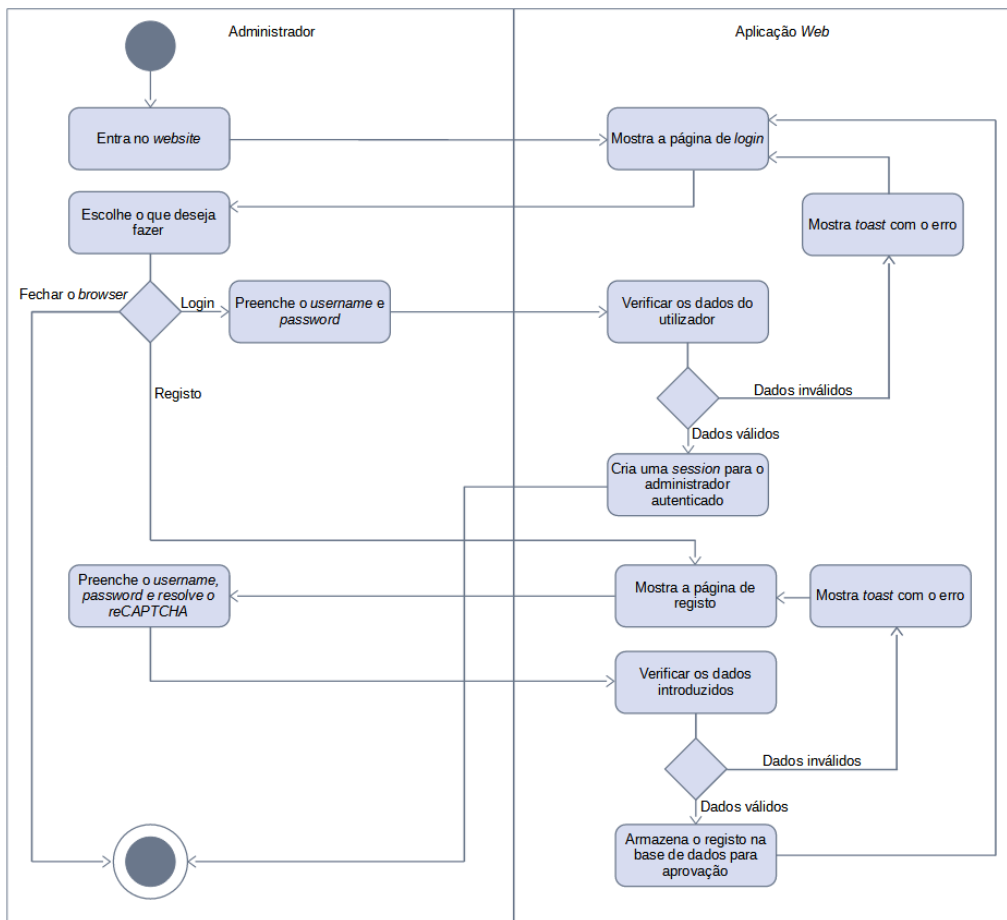


Figura 3.2: Diagrama de atividades da autenticação do administrador.

Também disponível ao administrador está a página de administração de utilizadores. Nesta página, o administrador pode aceitar ou rejeitar o registo de novos administradores, ou expulsar um já registado. O diagrama de atividade da página de administração de utilizadores está representado na figura 3.4.

Por fim, na perspetiva do utilizador, este acede a uma localização via um código QR. Após o *scan* do código QR, o utilizador é redirecionado para a página da localização. Nesta página, o utilizador pode calcular a rota para outra localização. O diagrama de atividades da página da localização está representado na figura 3.5.



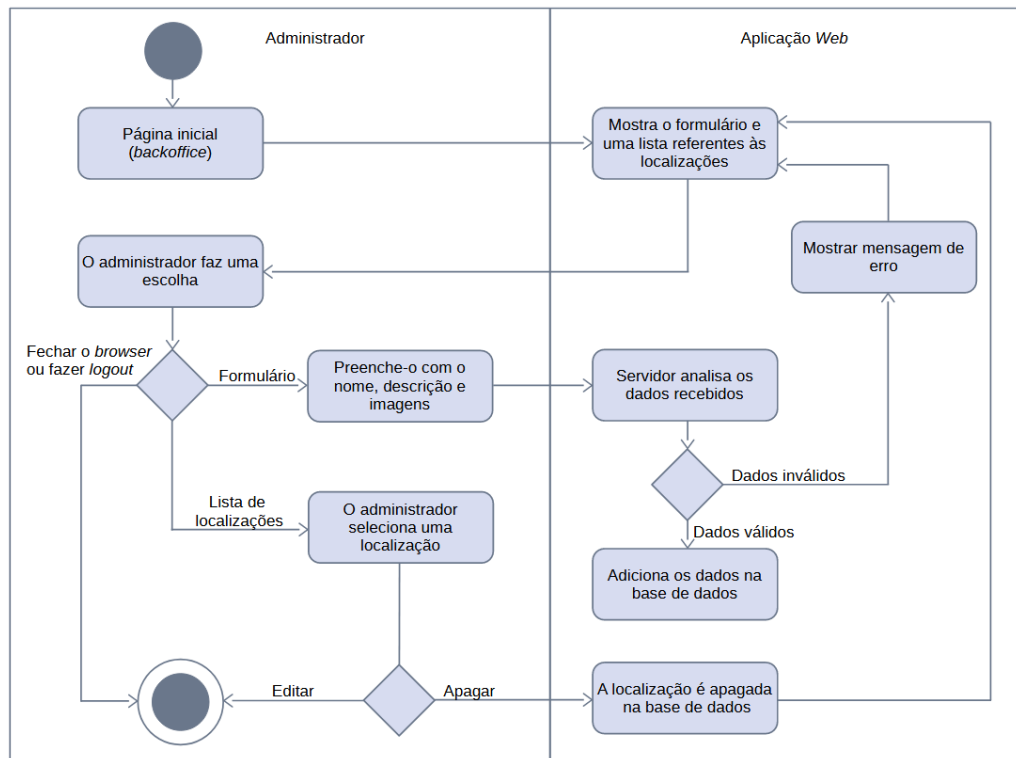


Figura 3.3: Diagrama de atividades da página inicial do *backoffice*.

## 3.5 Modelação da Base de Dados

A modelação de bases de dados é o processo de criar uma representação visual de como os dados serão organizados e armazenados num sistema de gestão de bases de dados. A modelação de bases de dados é uma etapa crítica no desenvolvimento de sistemas, por ajudar a garantir que os dados sejam armazenados eficientemente e possam ser facilmente acedidos e manipulados pelos utilizadores.

Nesta secção serão apresentados dois modelos, o modelo conceptual na secção 3.5.1 e o modelo físico na secção 3.5.2, separados em duas subsecções.

### 3.5.1 Modelo Conceptual

Esta subsecção apresenta o modelo conceptual da base de dados do sistema. É composto por entidades e relações entre essas entidades. Este modelo é apresentado na figura 3.6.

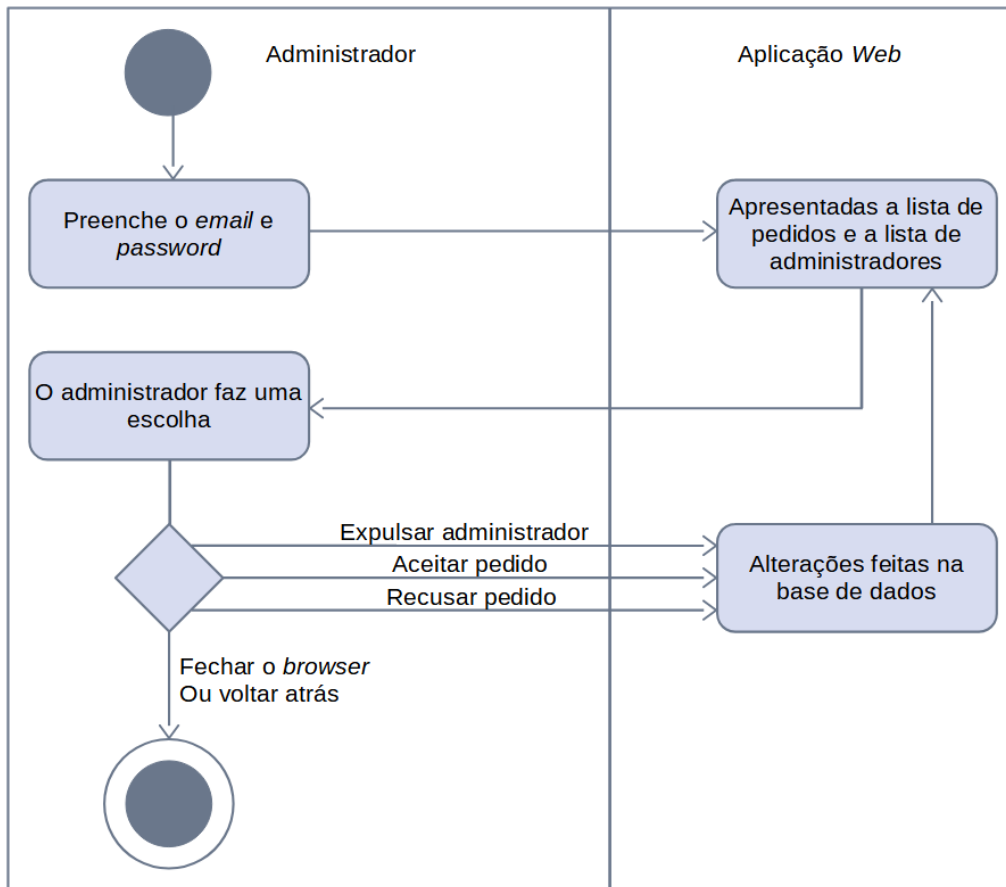


Figura 3.4: Diagrama de atividades da página de administração de utilizadores.

### 3.5.2 Modelo Físico

Nesta subsecção será apresentado o modelo físico da base de dados do sistema onde se encontram as chaves e os seus tipos, assim como as relações e os tipos das variáveis. Este modelo é apresentado na figura 3.7.

## 3.6 Arquitetura do Sistema

Nesta secção será apresentada a arquitetura do sistema, onde se encontram os componentes do sistema e as suas relações. Esta arquitetura é apresentada na figura 3.8.

O sistema é composto por uma aplicação *web* seguindo a estrutura *Linux*, *Apache*, *MySQL*, *PHP* (LAMP) e pelos diferentes *browsers* que se ligam ao ser-

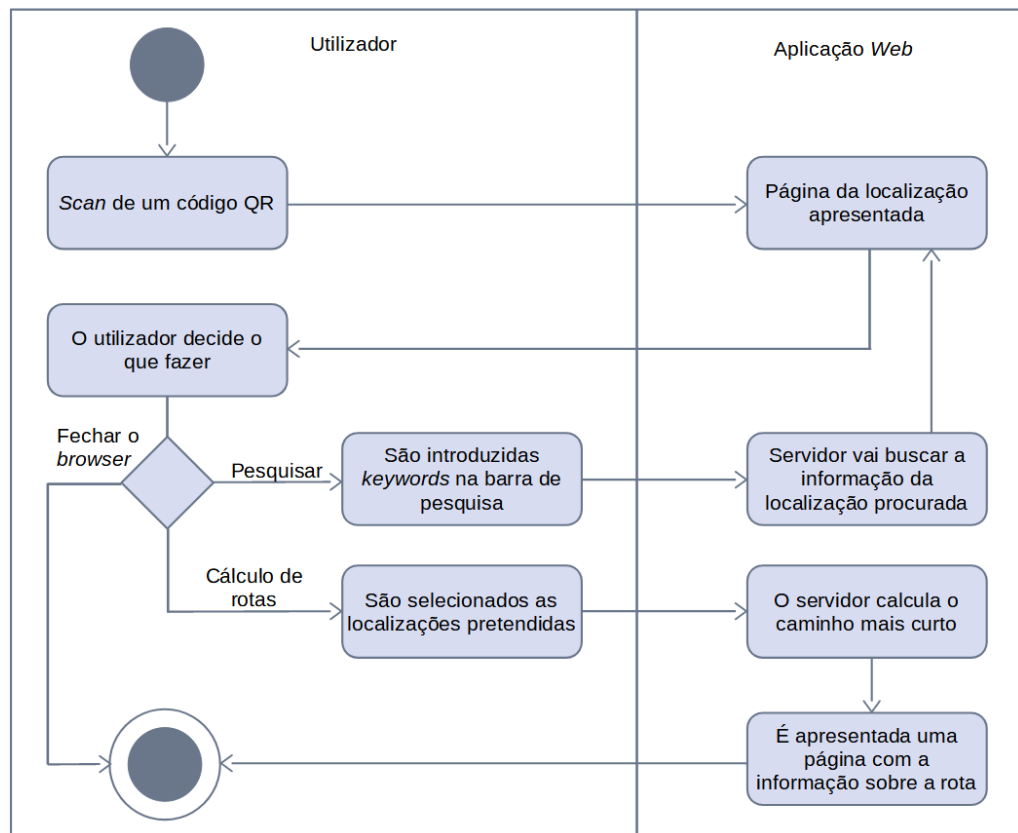


Figura 3.5: Diagrama de atividades da página de uma localização.

vidor *web* por *Hypertext Transfer Protocol Secure* (HTTPS). O servidor *web* comunica com a base de dados para obter os dados necessários para apresentar ao utilizador e processa-os utilizando a linguagem PHP.

### 3.7 Conclusões

Neste capítulo foram descritos os requisitos do sistema, funcionais e não funcionais, assim como, os casos de uso do sistema, a modelação da base de dados e por último a arquitetura do sistema.

Com base no processo de engenharia de *software* foram definidos os pontos necessários para a implementação do sistema, estes asseguram que o sistema é desenvolvido de acordo com os requisitos estipulados e que será desenvolvido de forma eficiente e segura.

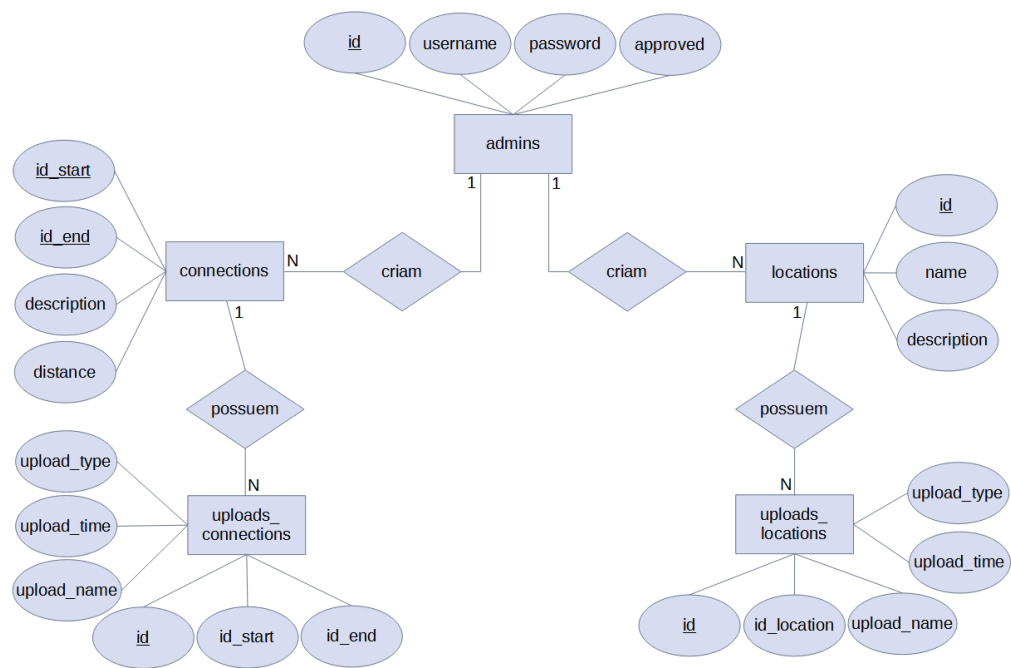


Figura 3.6: Modelo conceitual de dados usado na aplicação.

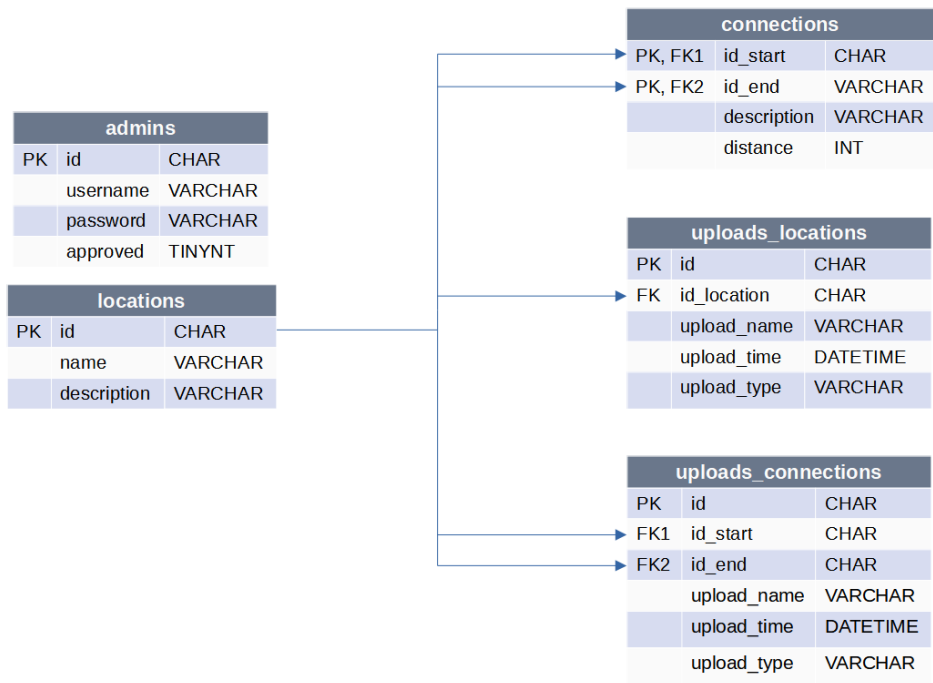


Figura 3.7: Modelo físico dos dados usado na aplicação.

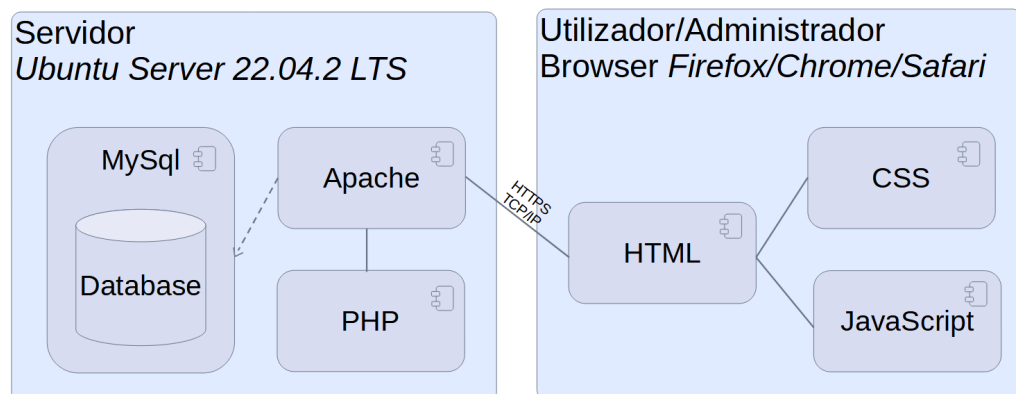


Figura 3.8: Diagrama de componentes que demonstra a arquitetura do sistema.



## Capítulo

# 4

## Implementação

### 4.1 Introdução

Neste capítulo será descrito o processo de implementação da aplicação *web* desenvolvida. Este encontra-se dividido em três secções. A primeira secção, 4.2, descreve o servidor e os vários componentes que este necessita, a segunda secção, 4.3, descreve as funcionalidades da aplicação *web*, e a última secção, 4.4, descreve as medidas de segurança implementadas.

### 4.2 Servidor

O princípio base na criação deste servidor foi aplicar a pilha de software LAMP, a qual é um conjunto de *software* livre para desenvolvimento de aplicações *web*, que inclui o sistema operativo *Linux*, o serviço *Apache*, o sistema de gestão de bases de dados MySQL e a linguagem de programação PHP.

Para além destes, foi utilizada outra ferramenta denominada *Certbot*. Esta ferramenta foi usada na criação e renovação dos certificados *Let's Encrypt*, para permitir uma conexão segura entre o servidor e o cliente através do protocolo HTTPS. A configuração destes componentes está descrita no capítulo 5.2, subsecção 5.2.1.

### 4.3 Funcionalidades

Esta secção descreve as principais funcionalidades da aplicação *web* desenvolvida. As primeiras três secções são relativas aos administradores da aplicação, enquanto as restantes são relativas aos utilizadores.

### 4.3.1 Autenticação

Para aceder ao *backoffice* da aplicação, o administrador terá de se autenticar com uma conta de administração composta por um nome de utilizador e uma palavra-passe. Para além da necessidade de existir uma conta, esta terá de ser aprovada por um administrador já presente na plataforma, antes da primeira autenticação, para evitar registos de utilizadores não autorizados.

Como inicialmente não existe nenhum administrador na plataforma, o primeiro administrador será automaticamente aprovado, logo é recomendado criar este utilizador antes de colocar a aplicação disponível para o público.

Para efetuar um registo, o administrador terá de preencher um formulário (figura 4.1) com os campos nome de utilizador, palavra-passe, confirmação da palavra-passe, e completar um *reCAPTCHA* para evitar *spam* de registos.

#### Registar

Por-favor preencha este formulário para criar uma conta.  
Esta é a primeira vez que está a iniciar o servidor.  
Por isso, a sua conta será aprovada automaticamente.

Username

Password

Repita a Password



I'm not a robot



reCAPTCHA

Privacy - Terms

Registar

Já tem uma conta? [Login](#)

Figura 4.1: Captura de ecrã demonstrativa da página de registo do primeiro administrador.

É de salientar que o nome de utilizador terá de ser único, ou seja, não



poderá existir nenhum utilizador com o mesmo nome e a palavra-passe terá de ter no mínimo 8 caracteres.

Após registado e aceite, o administrador poderá autenticar-se na aplicação através do formulário de *login* presente na página inicial da aplicação. Aqui ele terá de preencher os campos nome de utilizador e palavra-passe. Caso exista algum erro, este será apresentado ao utilizador numa forma de *toast* na parte inferior do ecrã. A figura 4.2 demonstra a página de *login* da aplicação.

## Bem-vindo à Localização@UBI

Por favor, faça login para entrar no backoffice.

**Username**

**Password**

Login

Não tem conta? [Registe-se](#)

Figura 4.2: Captura de ecrã demonstrativa da página de *login* da aplicação desenvolvida.

O processo de autenticação é feito através da obtenção dos dados inseridos (*username* e *password*), validar o *username* e *password* e, por fim, caso os

dados sejam válidos, redirecionar o administrador para a página principal do *backoffice* da aplicação (página *dashboard*).

Já dentro do *backoffice* e no contexto de autenticação, o administrador poderá aceitar ou rejeitar pedidos de registo de novos administradores como também excluir administradores já existentes.

Para tal, será apresentada uma lista de todos os administradores existentes na plataforma, onde cada um terá um botão para aceitar ou rejeitar o seu pedido de registo e outra lista similar com apenas um botão para excluir. Ambas as listas usufruem de uma barra de pesquisa de modo a facilitar a procura de um administrador específico.



Figura 4.3: Captura de ecrã demonstrativa da lista de administradores apresentada na página de administração da aplicação desenvolvida.

### 4.3.2 Gestão de Localizações

No contexto da aplicação, uma localização é um local onde se encontra um código QR que está associado a um ponto de interesse. Esta localização é identificada por um nome, uma descrição e uma ou várias imagens.

As localizações devem ser criadas pelos administradores da aplicação, sendo que cada administrador apenas poderá criar localizações para os pontos de interesse que lhe estão atribuídos.

Para criar uma localização, o administrador terá de preencher um formulário com os campos nome, descrição e imagens. O campo nome terá de ser único, ou seja, não poderão existir duas localizações com o mesmo nome. O campo descrição é um campo de texto livre onde o administrador poderá descrever a localização. O campo imagens é um campo de ficheiros onde o administrador poderá seleccionar uma ou várias imagens para associar à localização. A figura 4.4 demonstra o formulário de criação de localizações.

A imagem mostra a interface de um navegador com o título "Localização@UBI". No topo, há uma barra de navegação com "Dashboard" e "Administração" à esquerda, e "Bom dia, Rodrigo" com um ícone de perfil à direita. Abaixo, o título "Adicionar uma localização" é seguido por uma instrução: "Adicione uma nova localização à base de dados." O formulário principal contém três campos: "Nome" (um campo de texto simples), "Descrição" (um campo de texto de área) e "Imagens" (um botão "Browse..." e o texto "No files selected."). Um botão "Adicionar" está no canto inferior direito do formulário.

Figura 4.4: Captura de ecrã demonstrativa do formulário de criação de localizações.

Após criada uma localização, esta estará listada abaixo do formulário de criação de localizações. Nesta lista estão presentes todas as localizações criadas pelos administradores que, quando clicadas, expandem e apresentam a descrição, um botão para apagar e outro para editar. Esta lista usufrui de uma barra de pesquisa de modo a facilitar a procura de uma localização específica. A figura 4.5 demonstra a lista de localizações.

Ao clicar no botão de apagar, a aplicação irá gerar um aviso de confirmação de modo a evitar que o administrador apague uma localização por engano. Após a confirmação, a localização será apagada.

No caso da opção de editar, o utilizador será redirecionado para outra página onde poderá editar os campos nome e descrição, remover ou adicionar imagens, visualizar ou guardar o código QR da localização, criar uma conexão e visualizar as conexões já existentes, com as opções de remover ou editar tais conexões.

### Localizações

Uma lista com todas as localizações disponíveis.

A interface de utilizador para a gestão de localizações. No topo, há um campo de pesquisa com o texto 'Pesquisar...'. Abaixo dele, uma lista de localizações. A primeira localização é '6ª Fase - Entrada', que está expandida para mostrar uma descrição: 'Contém o bar, o balcão de informações, os anfiteatros 6.01, 6.02, 6.03 e o elevador.' À direita desta descrição estão dois botões: 'Apagar' e 'Editar'. Seguem-se seis outras localizações, cada uma com um ícone de seta para baixo indicando que estão recolhidas: '6ª Fase - Escadas Principais Piso 0', '6ª Fase - Escadas Principais Piso 1', '6ª Fase - Escadas Principais Piso 2', '6ª Fase - Escadas Principais Piso 3', '6ª Fase - Gabinetes Piso 2' e '6ª Fase - Gabinetes Piso 3'.

Figura 4.5: Captura de ecrã demonstrativa da lista de localizações apresentada na página *dashboard* da aplicação desenvolvida.

Caso o administrador deseje alterar o nome, a descrição ou as imagens, este terá de preencher os campos respetivos e clicar no botão de confirmar. Se existir algum conflito a aplicação irá gerar um aviso que será apresentado ao utilizador numa forma de *toast* na parte inferior do ecrã.

Para adicionar uma imagem, o administrador terá de clicar em procurar e seleccionar as imagens que deseja adicionar, após seleccionadas, poderá clicar no botão de adicionar para confirmar a adição das imagens. Para remover uma imagem, o administrador terá de clicar no botão de remover presente na imagem que deseja remover. A figura 4.6 apresenta o formulário de edição de localização.

No caso do código QR, este é gerado automaticamente pela aplicação e apresentado ao administrador. Este código é único e possibilita identificar a localização. Este encontra-se abaixo do formulário de edição de localização e contém o código seguido de um botão para guardar o código na sua máquina. A figura 4.7 apresenta o código QR da localização.

### 4.3.3 Gestão de Conexões

No contexto da aplicação, uma conexão é uma ligação entre duas localizações. Esta conexão é identificada por uma origem, um destino, uma descrição

The screenshot shows a web application interface for 'Localização@UBI'. At the top, there is a dark blue header with the title 'Localização@UBI' in white. Below the header, a navigation bar contains 'Dashboard' and 'Administração' on the left, and a user profile 'Boa tarde, rodrigo.duarte@ubi.pt' with a logout icon on the right. The main content area is titled 'Localização' and contains a form with two input fields: 'Nome (Pré-visualizar)' with the value '6ª Fase - Entrada' and 'Descrição' with the value 'Contém o bar, o balcão de informações, os anfiteatros 6.01, 6.02, 6.03 e o elevador.' A 'Confirmar' button is at the bottom right of the form. Below the form, there is a section titled 'Imagens' with a 'Browse...' button, the text 'No files selected.', and an 'Adicionar' button.

Figura 4.6: Captura de ecrã demonstrativa do formulário de edição de localizações.

#### Código QR



Figura 4.7: Captura de ecrã demonstrativa do código QR da localização, gerado pela aplicação.

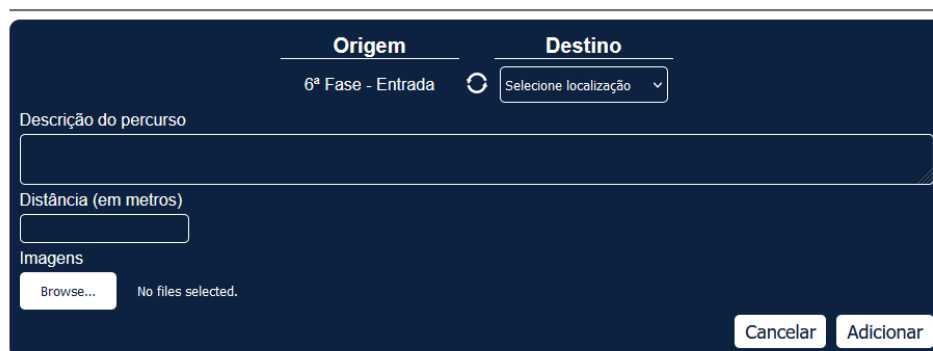
do percurso e uma distância em metros. Opcionalmente poderá ter uma ou várias imagens.

As conexões são apresentadas na página de edição de localizações, onde o

administrador poderá criar conexões ou observar as conexões existentes para uma dada localização. O formulário de criação encontra-se depois do código QR, na página de edição de localizações. Para criar uma localização, o administrador terá de selecionar uma origem e um destino, tendo em consideração que uma das localizações terá de ser a localização atual, caso queira inverter a ordem poderá clicar no botão de inverter. Além disso, será necessário colocar uma descrição com o itinerário a percorrer assim como a distância em metros. Opcionalmente poderá adicionar imagens para complementar a descrição.

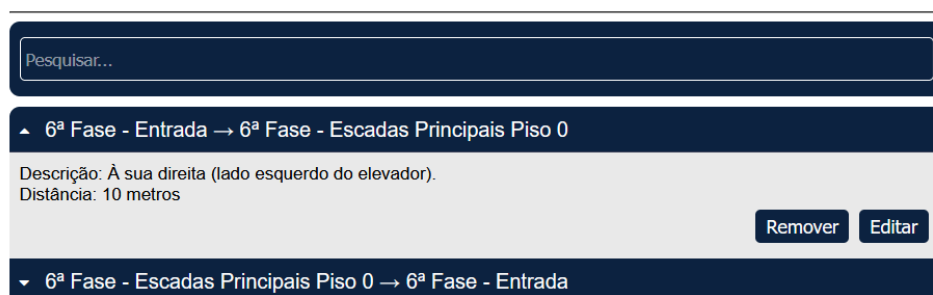
Abaixo do formulário de criação de conexão, é apresentada uma lista com todas as conexões já existentes. Cada conexão apresenta o nome da origem e do destino, a descrição e um botão para remover ou editar. Esta lista também usufrui de uma barra de pesquisa de modo a facilitar a procura de uma conexão específica. A figura 4.8 apresenta o formulário de criação de conexão e a lista de conexões.

#### Adicionar uma conexão



O formulário de criação de conexão possui um fundo escuro. No topo, há duas abas: 'Origem' e 'Destino'. A aba 'Origem' está selecionada e mostra '6ª Fase - Entrada'. Ao lado, há um botão de seta circular e um menu suspenso 'Selecione localização'. Abaixo, há um campo de texto 'Descrição do percurso', um campo 'Distância (em metros)' e uma seção 'Imagens' com um botão 'Browse...' e o texto 'No files selected.'. No canto inferior direito, há dois botões: 'Cancelar' e 'Adicionar'.

#### Conexões existentes



A lista de conexões existentes tem um fundo escuro. No topo, há uma barra de pesquisa com o texto 'Pesquisar...'. Abaixo, há uma lista de conexões. A primeira conexão é expandida, mostrando '6ª Fase - Entrada → 6ª Fase - Escadas Principais Piso 0'. Abaixo disso, há a descrição 'Descrição: À sua direita (lado esquerdo do elevador).', a distância 'Distância: 10 metros' e dois botões: 'Remover' e 'Editar'. A segunda conexão é '6ª Fase - Escadas Principais Piso 0 → 6ª Fase - Entrada'.

Figura 4.8: Captura de ecrã demonstrativa do formulário de criação de conexões e da lista de conexões da localização.

Ao clicar no botão de apagar, a aplicação irá gerar um aviso de confirma-

ção de modo a evitar que o administrador apague uma conexão por engano. Após a confirmação, a conexão será apagada.

Caso o administrador deseje editar, este será redirecionado para outra página onde poderá editar os campos descrição, distância e remover ou adicionar imagens. Na figura 4.9 é apresentado o formulário de edição de conexão.

Figura 4.9: Captura de ecrã demonstrativa do formulário de edição de uma conexão.

#### 4.3.4 Scan de Códigos QR

Na perspetiva do utilizador, o *scan* de códigos QR é uma das funcionalidades mais importantes da aplicação. Esta funcionalidade permite ao utilizador obter informações sobre a localização onde se encontra, assim como obter a rota para uma localização de destino.

Para aceder a página da localização, o utilizador terá de efetuar *scan* do código QR colocado na localização. Depois do *scan* do código, o utilizador será redirecionado para a página da localização, onde encontrará toda a informação sobre a localização, imagens e também algumas opções de cálculo de rotas. A página também dispõe de uma barra de pesquisa que permite ao utilizador procurar por uma localização específica através do nome da mesma, ou pelos componentes incorporados na descrição.

É de salientar que existem dois tipos de identificadores na página, que vão auxiliar ao cálculo de rotas assim como a apresentação da forma como estas são apresentadas. Os dois identificadores são “*scan*” e “*browse*”. O primeiro identificador é utilizado para identificar a localização onde o utilizador se encontra, enquanto que o segundo é utilizado para identificar a localização que

o utilizador procurou usando a barra de pesquisa. Conforme a combinação de identificadores presente ele pode calcular as rotas desde o “*scan*” até ao “*browse*”, desde o “*scan*” até ao que o utilizador escolher numa caixa de seleção ou fazer a sua própria rota escolhendo as duas localizações que deseja.

Nas duas figuras seguintes apresenta-se o caso de quando o utilizador efetua o *scan* de um código QR 4.10 e quando o utilizador pesquisa por uma localização após fazer o *scan* 4.11.

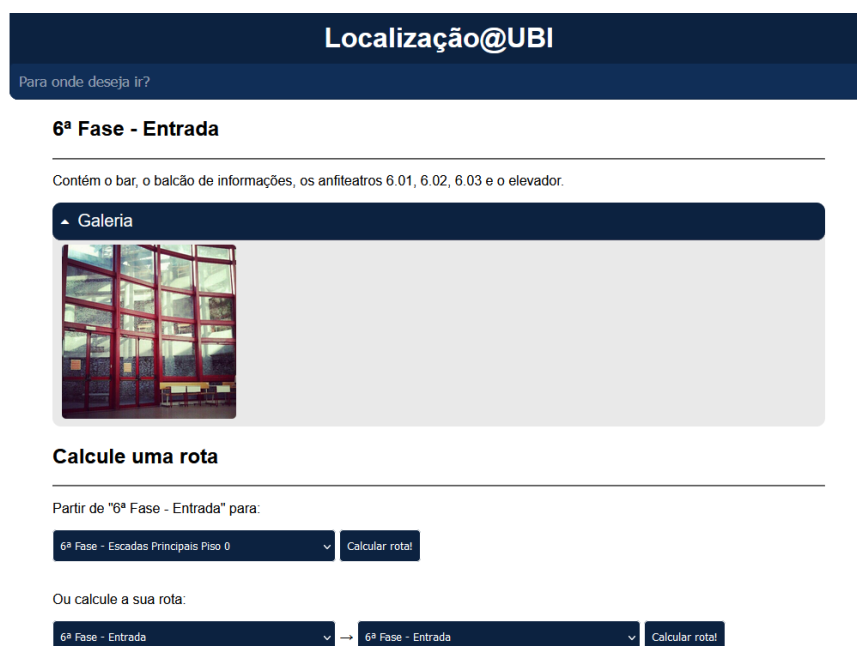


Figura 4.10: Captura de ecrã demonstrativa da página da localização após efetuar *scan* de um código QR.

### 4.3.5 Pesquisa de Localizações

A pesquisa de localizações é uma funcionalidade que permite ao utilizador procurar por uma localização específica. Esta funcionalidade está disponível no cabeçalho de todas as páginas da aplicação apresentadas ao utilizador.

Ao clicar na barra de pesquisa, o utilizador poderá escrever o nome da localização que deseja procurar. A aplicação irá apresentar uma lista de localizações que correspondem ao nome inserido. O utilizador poderá clicar numa das localizações apresentadas e será redirecionado para a página da localiza-





Figura 4.11: Captura de ecrã demonstrativa da página da localização após efetuar uma pesquisa.

ção. Adicionalmente será procurado na descrição da localização por palavras que correspondam às palavras introduzidas pelo utilizador.

A barra de pesquisa funciona com um sistema de pontuação, onde cada palavra introduzida pelo utilizador é comparada com o nome e descrição da localização e caso seja igual será atribuído um ponto à localização. No final, as localizações com mais pontos serão apresentadas em lugares superiores. Na figura 4.12 é apresentado o resultado da pesquisa por "biblioteca".

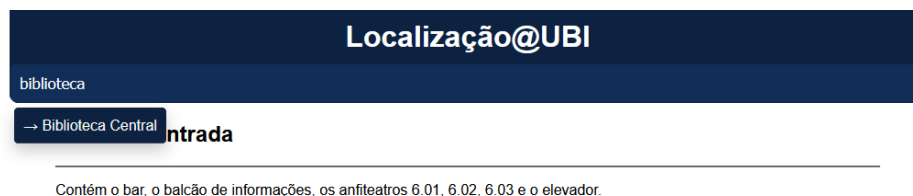


Figura 4.12: Captura de ecrã demonstrativa da pesquisa por "biblioteca" na barra de pesquisa da aplicação.

### 4.3.6 Cálculo de Rotas

O cálculo de rotas é uma funcionalidade que permite ao utilizador calcular a rota entre duas localizações. Esta funcionalidade está disponível na página de qualquer localização. O utilizador terá diferentes formas de calcular as rotas dependendo da combinação de identificadores “*scan*” e “*browse*” presentes na página. Na figura 4.13 é apresentado a secção relativa ao cálculo de rotas após o utilizador fazer o *scan* de um código QR.

#### Calcule uma rota

Partir de “6ª Fase - Entrada” para:

6ª Fase - Escadas Principais Piso 0    **Calcular rota!**

Ou calcule a sua rota:

6ª Fase - Piso 2    →    6ª Fase - Gabinetes Piso 2    **Calcular rota!**

Figura 4.13: Captura de ecrã demonstrativa da secção de cálculo de rotas na página da localização.

Para efeitos demonstrativos, a rota demonstrada será da localização “6ª Fase - Piso 2” até à localização “6ª Fase - Gabinetes Piso 2”, em que, como observado na figura 4.13, o utilizador terá de escolher nas caixas de seleção as localizações de origem e destino. Seguidamente o utilizador terá de clicar no botão *Calcular Rota* para a aplicação calcular a rota. Na figura 4.14 é apresentado o resultado do cálculo de rotas. As imagens apresentadas estão desfocadas devido a questões de privacidade.

Nesta página são apresentadas as instruções para o utilizador chegar ao destino. As instruções são apresentadas por uma descrição em texto do trajeto, seguido da distância aproximada até chegar à próxima instrução, e também é acompanhada de uma ou mais imagens demonstrativas do caminho que irá percorrer.

## 4.4 Segurança

Como vários utilizadores acessam à aplicação, é necessário assegurar a integridade dos dados dos mesmos, e também da informação disponibilizada pela aplicação. Para tal, foram implementadas algumas medidas de segurança. Estas medidas são apresentadas nas subsecções seguintes.

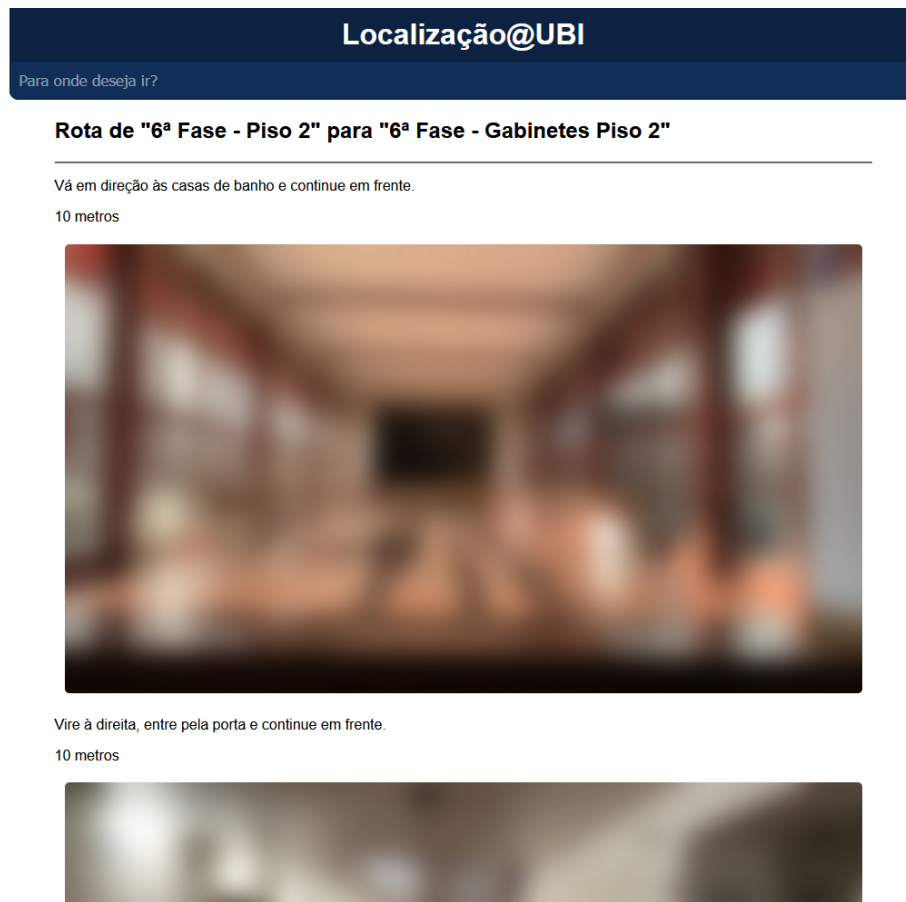


Figura 4.14: Captura de ecrã demonstrativa do resultado do cálculo de rotas.

#### 4.4.1 BCrypt

Para assegurar que as *passwords* dos administradores não são guardadas em texto simples, foi utilizada a função *password\_hash* [15] do *PHP* que utiliza o algoritmo *BCrypt* para as *passwords* serem representadas na forma de *hashes* na base de dados.

O *BCrypt* utiliza, na sua base, o algoritmo *Blowfish* e é utilizado para cifrar *passwords* de forma segura. O *BCrypt* divide-se em duas fases, em que na primeira fase é feito o *setup* da chave de cifra para a segunda fase; este *setup* envolve a *password* do administrador, um *salt* aleatório e o custo de iterações. Na segunda fase, é utilizada a chave da primeira fase para cifrar sessenta e quatro vezes a *string* “*OrpheanBeholderScryDoubt*” [16], que é o *hash* que é devolvida pela função *password\_hash*.

#### 4.4.2 ID Hashing

Para assegurar quaisquer identificadores não são sequenciais, foi utilizada a função de *hash* [17] que utiliza o algoritmo *SHA-256* para gerar um *hash* de 64 caracteres a partir de um número aleatório, usando a função *random\_int* [18] e uma data. Com isto, é possível garantir que os identificadores não são sequenciais e que não é possível prever o próximo identificador, o que evita que um utilizador aceda a todas as localizações com facilidade e extraia essa informação para usos maliciosos.

#### 4.4.3 Proteção contra SQL Injection

Para proteger a aplicação contra *Structured Query Language (SQL) Injection* foram utilizados *prepared statements* [19] para todas as *queries* feitas à base de dados. Com isto, é possível garantir que os dados inseridos pelo utilizador não são interpretados como código SQL e que não é possível aceder a dados que não são supostos ser acedidos. No excerto de código 4.1 é apresentado um exemplo de uma *query* que está protegida contra esse tipo de ataques.

```
// Add location to database
$id_location = generate_id($connection, "locations");
$sql = "INSERT INTO locations (id, name, description) VALUES (?, ?, ?)";
$stmt = mysqli_prepare($connection, $sql);
mysqli_stmt_bind_param($stmt, "sss", $id_location, $name, $description);
mysqli_stmt_execute($stmt);
$result = mysqli_stmt_get_result($stmt);
```

Excerto de Código 4.1: Exemplo de uma *query* protegida contra *SQL Injection*

### 4.5 Conclusões

Neste capítulo foram apresentadas as funcionalidades implementadas na aplicação e como estas funcionam. Além disso, foram apresentadas as medidas de segurança implementadas para proteger a aplicação.

## Capítulo

# 5

## ***Manual de Instalações e Testes***

### **5.1 Introdução**

Este capítulo visa explicar como instalar a aplicação nas secções 5.2, 5.3 e 5.4, os testes realizados ao longo do desenvolvimento da mesma, na secção na secção 5.5 e o inquérito elaborado sobre os diversos aspetos da aplicação, apresentados na secção 5.6.

### **5.2 Manual de Instalação**

Nesta subsecção serão explicados os passos necessários para instalar a aplicação *web* desenvolvida neste projeto.

Este projeto foi desenvolvido para o sistema operativo *Linux*, mais concretamente, a distribuição *Ubuntu Server 22.04.2 LTS*, pelo que os passos apresentados nesta subsecção são para este sistema operativo.

#### **5.2.1 Instalação e Configuração do Servidor**

Durante esta instalação vão ser necessários alguns pacotes que não estão instalados por defeito no sistema operativo. Para tal é recomendado atualizar os repositórios e instalar os pacotes necessários como *Apache2*, *MySQL*, *PHP* e alguns módulos do *PHP*, seguindo os passos apresentados no excerto de código 5.1.

```
sudo apt update
sudo apt upgrade
sudo apt install apache2 mysql-server php ufw
sudo apt libapache2-mod-php php-mysql php-gd
```

---

Excerto de Código 5.1: Instalação dos pacotes necessários para o servidor.

Após instalados, é recomendado reiniciar o servidor para as alterações terem efeito. Quando o servidor estiver operacional será necessário verificar que o serviço do *Apache2* está operacional. Para tal, basta executar o seguinte excerto de código 5.2.

```
sudo systemctl status apache2
```

Excerto de Código 5.2: Verificação do serviço do *Apache2*.

Se o serviço não estiver a correr, terá de executar o seguinte excerto de código 5.3 para iniciar o serviço.

```
sudo systemctl start apache2
```

Excerto de Código 5.3: Iniciar o serviço do *Apache2*.

Seguidamente é necessário configurar a *firewall* para permitir o acesso ao servidor. Para tal, execute o seguinte excerto de código 5.4.

```
sudo ufw allow in "Apache Full"
sudo ufw enable
```

Excerto de Código 5.4: Configuração da *firewall*.

Após configurada a *firewall*, é necessário configurar o *Apache2* para utilizar o PHP como pré-definição. Logo, é necessário editar o ficheiro *dir.conf* que se encontra na pasta */etc/apache2/mods-enabled*. Deverá executar o seguinte excerto de código 5.5 para editar o ficheiro.

```
sudo nano /etc/apache2/mods-enabled/dir.conf
```

Excerto de Código 5.5: Edição do ficheiro *dir.conf*.

Aberto o ficheiro *dir.conf*, é preciso alterar a ordem dos módulos para que o PHP seja o primeiro. Logo, basta alterar o conteúdo do ficheiro para o seguinte excerto de código 5.6.

```
<IfModule mod_dir.c>
    DirectoryIndex index.php index.html index.cgi index.pl index.
        xhtml index.htm
</IfModule>
```

Excerto de Código 5.6: Conteúdo do ficheiro *dir.conf*.

Quando o ficheiro for alterado, é necessário reiniciar o serviço do *Apache2* para as alterações terem efeito. Para tal, basta executar o seguinte excerto de código 5.7.

```
sudo systemctl restart apache2
```

Excerto de Código 5.7: Reiniciar o serviço do *Apache2*.

Agora, é aconselhado testar se o servidor funciona corretamente. Então será criado um ficheiro *info.php* na pasta */var/www/html* com o seguinte conteúdo apresentado no excerto de código 5.8.

```
<?php
    phpinfo () ;
?>
```

Excerto de Código 5.8: Conteúdo do ficheiro *info.php*.

Após criado o ficheiro, basta aceder ao endereço *http://localhost/info.php* ou *http://<ip-servidor>/info.php* e verificar se o *PHP* funciona corretamente.

## 5.2.2 Configuração da Base de Dados

Nesta subsecção será explicado como configurar a base de dados *MySQL* para o servidor. É aconselhado usar *mysql\_secure\_installation*, apresentado no excerto de código 5.9, e seguir os passos apresentados no terminal.

```
sudo mysql_secure_installation
```

Excerto de Código 5.9: Instalação da base de dados *MySQL*.

Depois configurar a base de dados, é necessário criar um utilizador e uma base de dados para a aplicação. Para tal, entre com a conta *root* e execute o seguinte excerto de código 5.10. Para efeitos de demonstração, o utilizador e a base de dados vão ser criados com o nome “*user*” e “*database*” respetivamente.

```
CREATE USER 'user'@'localhost' IDENTIFIED BY 'password';
CREATE DATABASE database;
GRANT ALL PRIVILEGES ON database.* TO 'user'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

Excerto de Código 5.10: Criação do utilizador e da base de dados.

Finalmente, efetue *login* com o utilizador *user* e crie as tabelas para a base de dados. O código *SQL* para criar as tabelas é apresentado no excerto de código A.1, contido no apêndice A.

## 5.3 Transferência da Aplicação

Depois das configurações-base do servidor, é necessário transferir o código para a aplicação *web*. Deve transferir os conteúdos da pasta *WebServer* para a pasta */var/www/html* e alterar as permissões da pasta para que o *Apache2* consiga aceder à pasta. O comando necessário para transferir os conteúdos da pasta *WebServer* para a pasta */var/www/html* é apresentado no excerto de código 5.11.

```
sudo cp -r WebServer/* /var/www/html
```

Excerto de Código 5.11: Transferência da aplicação *web*.



Depois transferir os conteúdos da pasta *WebServer* para a pasta */var/www/html*, é necessário alterar as permissões da pasta para que o *Apache2* consiga aceder à pasta. Para tal efeito, terá de executar o seguinte excerto de código 5.12. Para efeitos de demonstração o utilizador do servidor é denominado “*user*”.

```
sudo chown -R user html/  
sudo chgrp -R www-data html/  
sudo chmod -R 750 html/  
sudo chmod g+s html/  
sudo chmod g+w html/uploads/  
sudo chmod g+w html/uploads/locations  
sudo chmod g+w html/uploads/connections
```

Excerto de Código 5.12: Alteração das permissões da pasta */var/www/html*.

## 5.4 Pacotes Adicionais do *PHP*

Nesta secção será explicado como transferir os pacotes adicionais do *PHP* necessários para o funcionamento da aplicação.

### 5.4.1 Transferir o *PHP QR Code*

Para transferir o *PHP QR Code* necessita de aceder ao website *https://phpqrcode.sourceforge.net/* e dirigir-se aos *downloads*. A seguir clique na pasta *releases* e transfira o ficheiro *phpqrcode-2010100721\_1.1.4.zip*. Quando a transferência estiver concluída, descomprima o ficheiro e copie a pasta *phpqrcode* para a pasta */var/www/html/libs*.

### 5.4.2 Transferir o *reCAPTCHA*

Para transferir o *reCAPTCHA* necessita de aceder ao website *https://github.com/google/recaptcha*, clique em *releases* e transfira a última versão da versão *zip*. Quando a transferência estiver concluída, descomprima o ficheiro e altere o nome da pasta para *recaptcha-master*. A seguir copie a pasta *recaptcha-master* para a pasta */var/www/html/libs*.

## 5.5 Testes

Nesta secção serão demonstrados os testes realizados para verificar o funcionamento da aplicação *web*. Estes foram feitos com o intuito de verificar se a aplicação *web* funciona corretamente e se não existem erros. Na tabela 5.1

são apresentados os testes realizados, bem como os resultados esperados para esse teste.

<b>ID</b>	<b>Descrição do Teste</b>	<b>Resultado Esperado</b>
1	Efetuar um registo sem resolver o <i>reCAPTCHA</i> .	O utilizador não consegue efetuar o registo e aparece uma mensagem de erro.
2	Efetuar o <i>login</i> com um administrador válido.	O administrador é redirecionado para a página <i>dashboard</i> da aplicação <i>web</i> .
3	Criar uma localização sem imagem.	A localização é criada com sucesso.
4	Clicar no botão de editar uma localização.	O administrador é redirecionado para uma página onde pode editar a localização.
5	Clicar no botão de eliminar uma localização.	A localização é eliminada com sucesso.
6	Dentro da página de edição de uma localização, criar uma conexão.	A conexão é criada com sucesso.
7	Dentro da página de edição de uma localização, criar uma conexão sem selecionar uma localização.	A conexão não é criada e aparece uma mensagem de aviso.
8	Dentro da página de edição de uma conexão, escolher um campo aleatoriamente e alterar o seu valor. A seguir clicar no botão de confirmar.	O campo é alterado com sucesso.
9	Na página de administração, aceitar um pedido de registo.	O pedido de registo é aceite com sucesso.
10	Na página de administração, rejeitar um pedido de registo.	O pedido de registo é rejeitado com sucesso.
11	Na página de administração, clicar no botão de expulsar um administrador.	O administrador é eliminado com sucesso.

Tabela 5.1: Testes realizados à aplicação.

Após realizar todos os testes, verificou-se que a aplicação *web* funciona corretamente e que não existem erros. Na tabela 5.2 são apresentados os re-

sultados obtidos para cada teste.

<b>ID</b>	<b>Resultado do Teste</b>
1	O utilizador não consegue efetuar o registo e aparece uma mensagem de erro.
2	O administrador é redirecionado para a página <i>dashboard</i> da aplicação <i>web</i> .
3	A localização é criada com sucesso, aparecendo uma mensagem de sucesso.
4	O administrador é redirecionado para uma página onde pode editar a localização.
5	A localização é eliminada com sucesso.
6	A conexão é criada com sucesso.
7	A conexão não é criada e aparece uma mensagem de aviso.
8	O campo é alterado com sucesso e aparece uma mensagem de sucesso.
9	O pedido de registo é aceite com sucesso e o administrador é adicionado à lista.
10	O pedido de registo é rejeitado com sucesso e o administrador é removido da lista.
11	O administrador é eliminado com sucesso e removido da lista.

Tabela 5.2: Resultados dos testes realizados à aplicação.

Os resultados obtidos na tabela 5.2 para cada teste foram os esperados, o que significa que a aplicação *web* funciona corretamente e está conforme os requisitos esperados na tabela 5.1.

## 5.6 Inquérito sobre a Aplicação

Para além de testes, também foi elaborado um inquérito para recolher a opinião dos utilizadores sobre a aplicação *web*. Este inquérito foi realizado na plataforma *Google Forms* onde foram apresentadas quatro perguntas de resposta de nível de um a cinco, em que um significa péssimo e cinco significa excelente, e duas perguntas de resposta sim ou não para obter algum *feedback* relativo à ideia no qual a aplicação se baseia. As questões descritas foram as seguintes:

**Q1:** Quão útil acha esta aplicação?

**Q2:** Acha esta aplicação fácil de utilizar? Intuitiva?

**Q3:** O que acha acerca da sua interatividade?

**Q4:** Em termos visuais, acha que esta é visualmente apelativa?

**Q5:** Diria que esta aplicação poderá ajudar estudantes, funcionários e visitantes da Universidade da Beira Interior a se deslocarem dentro dos edifícios?

**Q6:** Considera, no futuro, usar esta aplicação para se orientar dentro da Universidade da Beira Interior?

O questionário foi respondido por um total de dezasseis pessoas, após utilização da aplicação. A maioria das pessoas que responderam ao questionário são estudantes da UBI, mais concretamente do curso de Engenharia Informática (EI), e por pessoas não tão versadas em tecnologia, como estudantes de outras áreas.

Os resultados obtidos mostram que a aplicação *web* é útil, fácil de utilizar, intuitiva e interativa. Quanto ao aspeto visual, a aplicação é razoavelmente apelativa, mas as respostas demonstram que pode ser um ponto a melhorar no futuro. Nas perguntas de resposta sim ou não, todos os utilizadores responderam que a aplicação poderá ajudar estudantes, funcionários e visitantes da Universidade da Beira Interior a se deslocarem nos edifícios, e quanto ao uso da aplicação no futuro, a grande maioria dos utilizadores responderam que iriam usar a aplicação. Os resultados obtidos encontram-se descritos na figura 5.1.

## 5.7 Conclusões

Neste capítulo foi explicado como configurar o servidor *web* com grande detalhe. Foram explicados todos os passos necessários para configurar o servidor *web* e transferir a aplicação *web* para o servidor. Também foram demonstrados os testes realizados para verificar o funcionamento da aplicação *web*, bem como as conclusões retiradas destes testes. Por fim, foi apresentado um inquérito realizado sobre a aplicação *web* e os resultados obtidos.

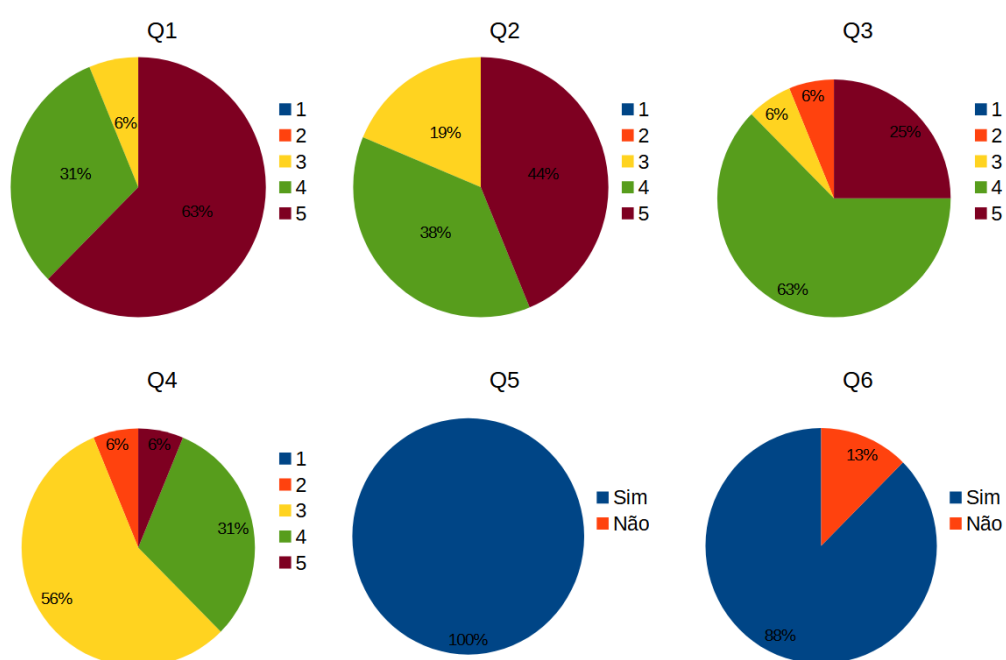


Figura 5.1: Resultados do inquérito realizado à aplicação *web*.



## ***Conclusões e Trabalho Futuro***

### **6.1 Conclusões Principais**

Este projeto consistiu na implementação de uma plataforma *web* que permite a gestão de localizações e conexões de uma região (mais focado na UBI), bem como a geração de códigos QR para as localizações, com o propósito de melhorar a acessibilidade, navegação e usabilidade da universidade para os utilizadores, que serão os alunos, atuais e futuros, funcionários e visitantes. Numa forma mais técnica, este teve em consideração as boas práticas de segurança, a adaptabilidade do sistema, para que este possa ser utilizado em qualquer dispositivo, e também foi pensado na possível expansão do sistema caso seja necessário realizar alterações ou adicionar novas funcionalidades.

Todas as funcionalidades na proposta do projeto foram implementadas com sucesso, tendo em conta os requisitos definidos no início do projeto como também foram adicionadas outras funcionalidades que não estavam previstas inicialmente, que foram consideradas importantes para o sistema após salientadas pelo orientador.

O sistema entregue funciona com êxito, tendo em conta os requisitos definidos e as funcionalidades implementadas. Este sistema foi testado com sucesso, tendo em conta os testes definidos e realizados. Pode ainda ser referido que o sistema apresenta uma solução de baixo custo, de fácil utilização e que pode ser alterada para qualquer tipo de contextos, não sendo apenas relevante num contexto universitário.

## 6.2 Trabalho Futuro

Mesmo tendo em conta que o sistema foi desenvolvido com sucesso, existem ainda algumas funcionalidades que podem ser implementadas no futuro, para melhorar o sistema e a experiência do utilizador e administrador. Algumas dessas funcionalidades são:

- Para além de imagens, ser possível adicionar vídeos às localizações e conexões;
- Acrescentar um sistema de comentários ou avaliações às localizações e conexões;
- Implementar um sistema que calcula rotas entre localizações, tendo em conta a acessibilidade do utilizador;
- Fazer um agrupamento de localizações, por exemplo, por edifícios, para que os administradores e utilizadores possam ter uma melhor noção da localização das mesmas;
- Melhorar o sistema de procuras de localizações, pelo uso de um algoritmo melhor ou pela apresentação de resultados mais relevantes ao utilizador;



## Apêndice

# A

## ***Excerto de código da base de dados***

Neste apêndice é apresentado um excerto de código da base de dados (excerto de código A.1), que contém a criação das tabelas utilizadas pelo sistema.

```
USE database;

CREATE TABLE admins (
    id            char(64)          NOT NULL,
    username      varchar(256)      NOT NULL,
    password      varchar(256)      NOT NULL,
    approved      tinyint           DEFAULT 0,

    CONSTRAINT PK_admins PRIMARY KEY (id)
);

CREATE TABLE locations (
    id            char(64)          NOT NULL,
    name          varchar(512)      NOT NULL,
    description    varchar(4096)    NOT NULL,

    CONSTRAINT PK_locations PRIMARY KEY (id)
);

CREATE TABLE connections (
    id_start      char(64)          NOT NULL,
    id_end        char(64)          NOT NULL,
    description    varchar(4096)    NOT NULL,
    distance      int               NOT NULL,

    CONSTRAINT PK_connections PRIMARY KEY (id_start, id_end),
    CONSTRAINT FK_connections_locations_s FOREIGN KEY (id_start)
```

```
REFERENCES locations (id),
CONSTRAINT FK_connections_locations_e FOREIGN KEY (id_end)
REFERENCES locations (id)
);

CREATE TABLE uploads_locations (
  id          char(64)      NOT NULL,
  id_location char(64)      NOT NULL,
  upload_name varchar(256)  NOT NULL,
  upload_time datetime      NOT NULL,
  upload_type varchar(64)   NOT NULL,

  CONSTRAINT PK_uploads PRIMARY KEY (id),
  CONSTRAINT FK_uploads_locations FOREIGN KEY (id_location)
REFERENCES locations (id)
);

CREATE TABLE uploads_connections (
  id          char(64)      NOT NULL,
  id_start    char(64)      NOT NULL,
  id_end      char(64)      NOT NULL,
  upload_name varchar(256)  NOT NULL,
  upload_time datetime      NOT NULL,
  upload_type varchar(64)   NOT NULL,

  CONSTRAINT PK_uploads PRIMARY KEY (id),
  CONSTRAINT FK_uploads_connections FOREIGN KEY (id_start, id_end)
REFERENCES connections (id_start, id_end)
);
```

Excerto de Código A.1: Criação das tabelas na base de dados.

# ***Bibliografia***

- [1] Haosheng Huang and Georg Gartner. *A survey of mobile indoor navigation systems*. Springer, 2010.
- [2] Pointr. Pointr, 2023. [Online] <https://pointr.tech/>. Último acesso a 6 de Junho de 2023.
- [3] mappedin. mappedin, 2023. [Online] <https://mappedin.com/>. Último acesso a 6 de Junho de 2023.
- [4] Apple. Apple indoor maps and positioning, 2023. [Online] [https://register.apple.com/resources/indoor/program/indoor\\_maps](https://register.apple.com/resources/indoor/program/indoor_maps). Último acesso a 6 de Junho de 2023.
- [5] Apache. Welcome! - the apache http server project, 2023. [Online] <https://httpd.apache.org>. Último acesso a 6 de Junho de 2023.
- [6] Certbot. Certbot, 2023. [Online] <https://certbot.eff.org>. Último acesso a 6 de Junho de 2023.
- [7] MySQL. Mysql, 2023. [Online] <https://www.mysql.com>. Último acesso a 6 de Junho de 2023.
- [8] PHP. Php: Hypertext preprocessor, 2023. [Online] <https://www.php.net>. Último acesso a 6 de Junho de 2023.
- [9] W3schools. Html tutorial, 2023. [Online] <https://www.w3schools.com/html/>. Último acesso a 6 de Junho de 2023.
- [10] W3schools. Css tutorial, 2023. [Online] <https://www.w3schools.com/css/>. Último acesso a 6 de Junho de 2023.
- [11] W3schools. Javascript tutorial, 2023. [Online] <https://www.w3schools.com/js/>. Último acesso a 6 de Junho de 2023.
- [12] jQuery. What is jquery?, 2023. [Online] <https://jquery.com/>. Último acesso a 6 de Junho de 2023.

- [13] PHP QR Code. Php qr code - qr code generator, an lgpl php library, 2023. [Online] <https://phpqrcode.sourceforge.net/>. Último acesso a 6 de Junho de 2023.
- [14] Google. What is recaptcha?, 2023. [Online] <https://www.google.com/recaptcha/about/>. Último acesso a 6 de Junho de 2023.
- [15] PHP. password\_hash, 2023. [Online] <https://www.php.net/manual/en/function.password-hash.php>. Último acesso a 6 de Junho de 2023.
- [16] usenix. Bcrypt algorithm, 2023. [Online] [https://www.usenix.org/legacy/publications/library/proceedings/usenix99/full\\_papers/provos/provos\\_html/node5.html](https://www.usenix.org/legacy/publications/library/proceedings/usenix99/full_papers/provos/provos_html/node5.html). Último acesso a 6 de Junho de 2023.
- [17] PHP. Php: hash - manual, 2023. [Online] <https://www.php.net/manual/en/function.hash.php>. Último acesso a 6 de Junho de 2023.
- [18] PHP. Php: random\_int - manual, 2023. [Online] <https://www.php.net/manual/en/function.random-int.php>. Último acesso a 6 de Junho de 2023.
- [19] PHP. Php: mysqli::prepare - manual, 2023. [Online] <https://www.php.net/manual/en/mysqli.prepare.php>. Último acesso a 6 de Junho de 2023.