

Universidade do Minho

Escola de Engenharia

Departamento de Informática

a83899 - André Morais

a86272 - João Coutinho

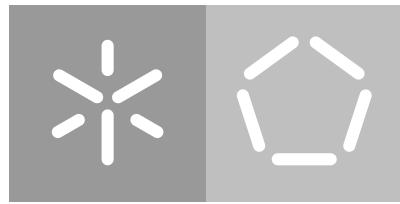
a84577 - José Pedro Silva

a85954 - Luís Ribeiro

a84783 - Pedro Rodrigues

PORTOURGAL

Julho 2021



Universidade do Minho

Escola de Engenharia

Departamento de Informática

a83899 - André Morais
a86272 - João Coutinho
a84577 - José Pedro Silva
a85954 - Luís Ribeiro
a84783 - Pedro Rodrigues

PORTOURGAL

Laboratórios de Informática IV
Mestrado Integrado em Engenharia Informática

Trabalho efetuado sob orientação de
Hugo Peixoto

Julho 2021

RESUMO

O presente relatório descreve as três fases do desenvolvimento da aplicação mobile "PORTOURGAL", elaborada no âmbito da unidade curricular de Laboratórios de Informática IV, cujo o seu objetivo é facilitar a atividade turística em Portugal.

A primeira fase (Fundamentação) consistiu na apresentação do domínio da aplicação, isto é, em que consistiria a aplicação a que nos estariamos a propor desenvolver. Para isso, foram descritos os objetivos, a contextualização, apresentação do caso de estudo, ferramentas a serem utilizadas, assim como a planificação do projeto e os mockups iniciais da aplicação.

A segunda fase (Especificação) corresponde à especificação dos requisitos, onde para além do levantamento e análise de requisitos, foram também criados, com recurso a UML, os modelos de sistema. Estes modelos correspondem ao diagrama de Use Cases, modelo de domínio e diagrama de classes. Após a conclusão do levantamento de requisitos e a construção dos modelos, procedeu-se então à idealização e elaboração da base de dados.

Na terceira e última fase deste projeto (Implementação), são apresentadas as principais funcionalidades implementadas. Sendo esta a última fase do projeto, estamos perante a culminação de todas as fases previamente desenvolvidas, dando origem ao produto final.

CONTEÚDO

1 FUNDAMENTAÇÃO	1
1.1 Título e Enquadramento	1
1.2 Objetivos, Motivação e Contribuição do Projeto	1
2 ESPECIFICAÇÃO	2
2.1 Levantamento de Requisitos	2
2.1.1 Requisitos Funcionais	2
2.1.1.1 Registo na Aplicação	2
2.1.1.2 Autenticação na Aplicação	2
2.1.1.3 Visualizar perfil	3
2.1.1.4 Edição de perfil dos utilizadores	3
2.1.1.5 Terminar Sessão	3
2.1.1.6 Visualizar Distritos e Roteiros	3
2.1.1.7 Visualizar Hotéis, Atrações e Restaurantes	4
2.1.1.8 Visualizar Restaurante	4
2.1.1.9 Visualizar Hotel	4
2.1.1.10 Visualizar Atração	4
2.1.1.11 Classificar	5
2.1.1.12 Marcar como visitado	5
2.1.1.13 Visualizar Roteiro	5
2.1.1.14 Visualizar Leaderboard	6
2.1.2 Requisitos não Funcionais	6
2.2 Modelação de Software	6
2.3 Modelo de Domínio	7
2.4 Diagrama de Use Cases	8
2.5 Especificação Use Cases	10
2.5.1 Criar Conta/Registrar Conta	10
2.5.2 Visualizar Perfil	10
2.5.3 Login	11
2.5.4 Logout	11
2.5.5 Repor Password	11
2.5.6 Classificar Atração	12
2.5.7 Editar Perfil	12
2.5.8 Visualizar Rota	12
2.5.9 Marcar local como visitado	13

2.5.10	Remover Conta	13
2.5.11	Gerir Rotas	13
2.6	Diagrama de Classes	13
3	IMPLEMENTAÇÃO	15
3.1	Funcionalidades	15
3.1.1	Página Inicial	15
3.1.2	Login	16
3.1.3	Registar	17
3.1.4	Perfil de utilizador	18
3.1.5	Editar perfil	19
3.1.6	Upload de uma imagem	20
3.1.7	Navegar pelos distritos/roteiros	21
3.1.8	Conteúdo de um distrito	22
3.1.9	Atração	23
3.1.10	Percorso	24
3.1.11	Sugestões para um percurso	25
3.1.12	Leaderboard	26
3.2	Base de Dados - Abordagem não relacional	27
3.2.1	Justificação da utilização de um sistema NoSQL	27
3.2.2	MongoDB	27
3.2.3	Implementação BD	27
3.2.3.1	Distrito	28
3.2.3.2	Utilizador	29
3.2.3.3	Roteiro	29
3.2.4	Acesso à Base de Dados	30
3.3	RestAPI - Comunicação entre cliente-servidor	31
3.3.1	RestAPI	31
3.3.2	Implementação	31
3.3.3	Endpoits	32
3.3.4	Demonstração	33
3.3.5	API externa	36
3.4	Cliente - Servidor - Base de Dados	37
3.5	Segurança	37
3.5.1	Implementação	37
3.6	Administração do Sistema	38
3.6.1	Demonstração	38
3.7	Identificação dos recursos utilizados	43
4	CONCLUSION	44

LISTA DE FIGURAS

Figura 1	Modelo de Domínio	7
Figura 2	Diagrama de Use Cases inicial	8
Figura 3	Diagrama de Use Cases Final	9
Figura 4	Diagrama de classes.	14
Figura 5	Página inicial.	15
Figura 6	Página de login.	16
Figura 7	Página de registo	17
Figura 8	Página de perfil	18
Figura 9	Página para editar o perfil.	19
Figura 10	Anexar uma imagem para foto de perfil.	20
Figura 11	Página dos distritos e roteiros.	21
Figura 12	Exemplo de um distrito.	22
Figura 13	Exemplo de uma atração.	23
Figura 14	Exemplo de um percurso.	24
Figura 15	Sugestões para um percurso exemplo.	25
Figura 16	Tabela de classificações.	26
Figura 17	GET /distritos.	33
Figura 18	GET /distritos/nome/:nomeASCII.	34
Figura 19	GET /roteiros.	34
Figura 20	GET /roteiros/nome/:nomeASCII.	35
Figura 21	GET /users.	35
Figura 22	GET /users/nome/:nomeASCII.	36
Figura 23	Tráfego do Sistema.	37
Figura 24	Visualização de todos utilizadores.	38
Figura 25	Visualização de um utilizador.	39
Figura 26	Visualização de todos os distritos.	39
Figura 27	Visualização de um distrito, contendo todas as cidades.	40
Figura 28	Visualização de uma cidade, contendo todas as atrações, restaurantes e hotéis.	40
Figura 29	Visualização de uma atração.	41
Figura 30	Criação de uma nova atração.	41
Figura 31	Visualização de todos os roteiros.	42
Figura 32	Visualização de um roteiro.	42

1

FUNDAMENTAÇÃO

1.1 TÍTULO E ENQUADRAMENTO

A razão pela qual decidimos intitular a aplicação de PORTOURGAL, foi devido ao facto de a aplicação ser desenvolvida com o propósito de facilitar o processo turístico em localidades Portuguesas. O título é formado pela junção de duas palavras, sendo estas "tour" e "Portugal" (formando então "PORTOURGAL"), onde a palavra "tour" é elucidativa à parte de viajar e visitar e a palavra "Portugal" é referente ao país a ser visitado.

1.2 OBJETIVOS, MOTIVAÇÃO E CONTRIBUIÇÃO DO PROJETO

Num mundo em que o turismo é uma atividade em constante crescimento, é nas pequenas decisões que ficamos deslumbrados ou desapontados com a localidade que estamos a visitar. O aumento do interesse pela atividade turística serviu como motivação para o desenvolvimento da PORTOURGAL. Após ter sido feita uma extensa pesquisa, apercebemo-nos que não existiam aplicações que aplicavam os objetivos que tínhamos em mente, tendo isso sido também um fator de peso na motivação para o desenvolvimento da aplicação.

Desta forma, a PORTOURGAL tem como objetivo oferecer uma aplicação viável que permite um utilizador navegar entre as distintas atrações turísticas, restaurantes e hóteis de cada cidade do país. Para além de fornecer a funcionalidade acima descrita, a PORTOURGAL oferece também a possibilidade de um utilizador escolher um roteiro, sendo este composto por um conjunto de cidades.

De modo a promover a aplicação e no futuro receber possíveis patrocínios, temos como objetivo, para cada roteiro, ter uma página com sugestões contendo atrações, restaurantes e hóteis para cada cidade pertencente ao roteiro.

De modo a diferenciar a PORTOURGAL de outras aplicações de turismo, existe ainda o objetivo de promover o turismo rural, tendo por isso em mente, um sistema de pontos onde as cidades do interior valem mais pontos, sendo atribuídos prémios mensais aos três utilizadores que mais pontos acumularem.

2

ESPECIFICAÇÃO

2.1 LEVANTAMENTO DE REQUISITOS

O levantamento e análise de requisitos é uma das fases mais importantes para garantir o bom desenvolvimento do software. Serão apresentados os requisitos funcionais e não funcionais, tanto do sistema como do utilizador, podendo este ser cliente ou admin.

2.1.1 *Requisitos Funcionais*

2.1.1.1 *Registo na Aplicação*

1. Definição Requisitos de utilizador

- O utilizador tem que se registar na aplicação para a poder utilizar.

2. Especificação de requisitos de sistema

- No momento do registo, o sistema deve solicitar os dados necessários ao utilizador, nomeadamente o nome, o e-mail, o seu distrito e respetiva cidade e uma password;
- O sistema não permite o registo de utilizadores com o mesmo e-mail;
- O sistema guarda a informação do utilizador.

2.1.1.2 *Autenticação na Aplicação*

1. Definição Requisitos de utilizador

- O utilizador deverá introduzir o seu e-mail e password para iniciar sessão.

2. Especificação de requisitos de sistema

- O Sistema verifica a validade da autenticação, analisando se o e-mail e password estão corretos. Caso contrário, o sistema não permite a autenticação.

2.1.1.3 Visualizar perfil

1. Definição Requisitos de utilizador

- O utilizador deverá puder visualizar o seu perfil depois de se autenticar, podendo dar scroll down para a visualizar as atrações por ele visitadas.

2. Especificação de requisitos de sistema

- O sistema deve apresentar as informações do utilizador (nome, cidade, distrito, os seus pontos e foto de perfil) e também guardar as atrações que este marcou como visitado.

2.1.1.4 Edição de perfil dos utilizadores

1. Definição Requisitos de utilizador

- O utilizador pode editar as suas informações pessoais, como o seu e-mail, password, distrito, cidade, nome e foto de perfil.

2. Especificação de requisitos de sistema

- O sistema deve armazenar as alterações.

2.1.1.5 Terminar Sessão

1. Definição Requisitos de utilizador

- O utilizador pode terminar a sessão através do seu perfil.

2. Especificação de requisitos de sistema

- O sistema deve permitir que o utilizador termine a suas sessão e volta ao menu inicial.

2.1.1.6 Visualizar Distritos e Roteiros

1. Definição Requisitos de utilizador

- O utilizador deve conseguir visualizar todos os distritos e roteiros da aplicação, através de um swipe left.

2. Especificação de requisitos de sistema

- O sistema apresenta os distritos e os roteiros existentes num carousel view, onde as imagens são clicáveis;
- O sistema deverá permitir que o utilizador selecione os distritos ou roteiros que pretende visualizar.

2.1.1.7 Visualizar Hotéis, Atrações e Restaurantes

1. Definição Requisitos de utilizador

- O utilizador deve conseguir visualizar todos os hotéis, atrações e restaurantes da aplicação, através de um swipe left.

2. Especificação de requisitos de sistema

- O sistema apresenta os distritos e os roteiros existentes numa carousel view, onde as imagens são clicáveis;
- O sistema deverá permitir que o utilizador selecione os distritos ou roteiros que pretende visualizar.

2.1.1.8 Visualizar Restaurante

1. Definição Requisitos de utilizador

- O utilizador consegue visualizar a informação deste restaurante com a sua morada, fotografia e respetiva classificação.

2. Especificação de requisitos de sistema

- O sistema deve possuir na base de dados a informação do restaurante e apresentá-la no ecrã do utilizador.

2.1.1.9 Visualizar Hotel

1. Definição Requisitos de utilizador

- O utilizador consegue visualizar a informação deste hotel com a sua morada, fotografia e respetiva classificação.

2. Especificação de requisitos de sistema

- O sistema deve possuir na base de dados a informação do hotel e apresentá-la no ecrã do utilizador.

2.1.1.10 Visualizar Atração

1. Definição Requisitos de utilizador

- O utilizador consegue visualizar a informação desta atração com uma pequena descrição, uma fotografia e respetiva localidade/morada;
- O utilizador também pode classificar e marcar como visitado.

2. Especificação de requisitos de sistema

- O sistema deve possuir na base de dados a informação da atração e apresentá-la no ecrã do utilizador.

2.1.1.11 *Classificar*

1. Definição Requisitos de utilizador

- O utilizador pode classificar uma atracão, referindo qual a classificação que este quer atribuir a mesma.

2. Especificação de requisitos de sistema

- O sistema atualiza o número de estrela para pintado, consoante a classificação do utilizador;
- O sistema atualiza a média com a nova classificação dada pelo o utilizador.

2.1.1.12 *Marcar como visitado*

1. Definição Requisitos de utilizador

- O utilizador pode marcar uma atracão como visitado.

2. Especificação de requisitos de sistema

- O sistema atualiza o switch para on se esta tivesse em off;
- O sistema adiciona a foto da atração ao perfil se o botão estiver em on;
- O sistema atualiza o switch para off se esta tivesse em on;
- O sistema remove a foto da atração do perfil se o botão estiver em off.

2.1.1.13 *Visualizar Roteiro*

1. Definição Requisitos de utilizador

- O utilizador deve conseguir visualizar a imagem da rota, tal como os pontos de passagem e uma breve descrição;
- O utilizador deve conseguir visualizar os hotéis, atrações e restaurantes dos pontos de passagem.

2. Especificação de requisitos de sistema

- O sistema deve possuir na base de dados a informação do roteiro e apresentá-la no ecrã do utilizador;
- O sistema apresenta os hotéis, atrações e restaurantes, presentes nos pontos de passagem, num carousel view , onde estas imagens são clicáveis;

- O sistema deverá permitir que o utilizador selecione os hotéis, atrações e restaurantes que pretende visualizar.

2.1.1.14 Visualizar Leaderboard

1. Definição Requisitos de utilizador
 - O utilizador consegue visualizar a leaderboard.
2. Especificação de requisitos de sistema
 - O sistema apresenta as primeiras 3 pessoas com mais pontos, assim como o seu perfil e os seu pontos.

2.1.2 Requisitos não Funcionais

- A aplicação deverá estar disponível durante os 7 dias da semana, 24 horas por dia;
- A aplicação deverá ser de fácil uso, com um layout simples;
- O sistema deverá ser executado num aparelho com conexão à rede;
- O sistema deverá ser desenvolvido para a plataforma Android;
- O sistema de deverá proteger os dados dos diferentes utilizadores.

2.2 MODELAÇÃO DE SOFTWARE

Por forma a implementar de modo conciso e organizado todo o sistema a que nos propusemos, passamos pela modelação que consideramos necessária. Assim, todos os diagramas que a seguir se apresentam foram desenvolvidos de acordo com a linguagem UML, com o auxílio da ferramenta Visual Paradigm.

Deste modo, nos subcapítulos seguintes apresentaremos apenas alguns Modelos que nos foram pedidos implementar, sendo estes, o Modelo de Domínio, Diagrama de Use Cases, juntamente com a sua especificação, e por último, o Modelo de Classes da aplicação.

2.3 MODELO DE DOMÍNIO

O Modelo de Domínio é construído para ilustrar os conceitos e relações entre estes, apresentando assim um modelo conceptual do problema em questão e nele podem ser visualizadas as entidades que participam na aplicação, assim como os papéis e atributos que desempenham. Nesse sentido, consideramos que este se trata de um modelo fulcral pois serve como um ponto de partida para toda a modelação.

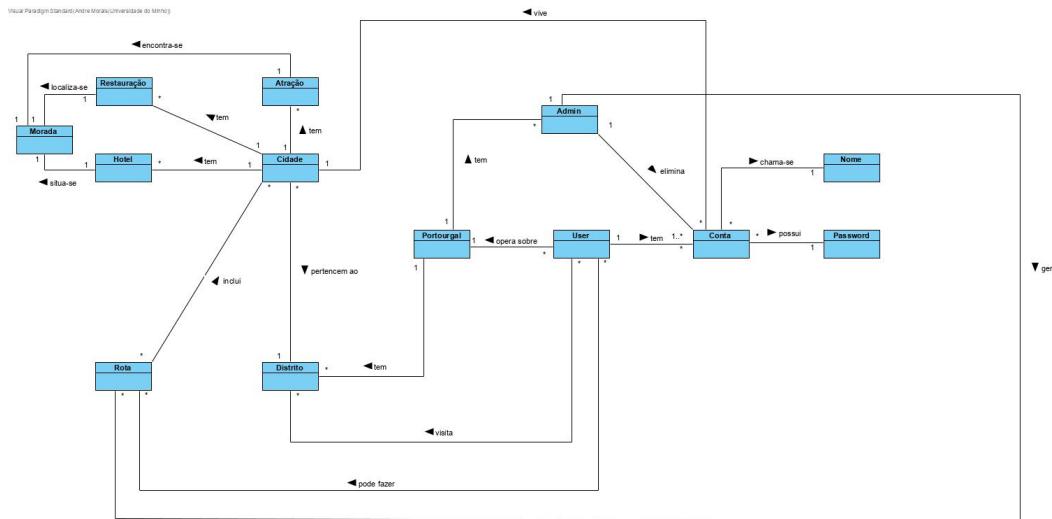


Figura 1: Modelo de Domínio

Consequentemente esboçamos a nossa versão, onde consideramos que as principais entidades do nosso sistema seriam: PORTOURGAL, User, Admin, Conta, Password, Nome, Distrito, Rota, Cidade, Atração, Morada, Restauração, Hotel.

Começando pelo PORTOURGAL a peça fundamental deste modelo, pois representa o “nome” da aplicação desenvolvida, que contêm Admins responsáveis pela aplicação, e para além disso é representada pelos Distritos. Os Users representam todos os utilizadores que vão operar/usar a aplicação, e estes terão uma Conta associada a cada um, com um Nome, Password e uma Cidade. Os Users operam sobre a PORTOURGAL de modo a visitar Cidades ou percorrer Rotas. As Rotas são identificadas por um conjunto de Cidades. Cada Cidade é identificada por Atrações, Restauração e Hotéis.

2.4 DIAGRAMA DE USE CASES

Os Use Cases são uma forma sistemática de capturar requisitos funcionais, fornecendo assim uma notação diagramática que permite modelar o contexto geral do sistema. Isto é, este diagrama representa os atores do Sistema (quem utiliza o Sistema), os use cases (o que se pode fazer no Sistema) e as associações entre os atores e os use cases. As entidades que consideramos serem as principais no sistema, isto é, as entidades que efetivamente vão interagir com ele são as seguintes: User e Admin.

O primeiro ator que identificamos foi o Administrador, que é responsável pela gestão das contas, sendo esta gestão apenas representada por um use case: Eliminar Conta. Este também é responsável por Gerir Rotas. Para isto tem que se autenticar.

O ator que se segue é o Utilizador, e este tem acesso a tudo o que a aplicação fornece, e precisa de fazer login para se autenticar.

Inicialmente consideramos o seguinte diagrama Use Cases:

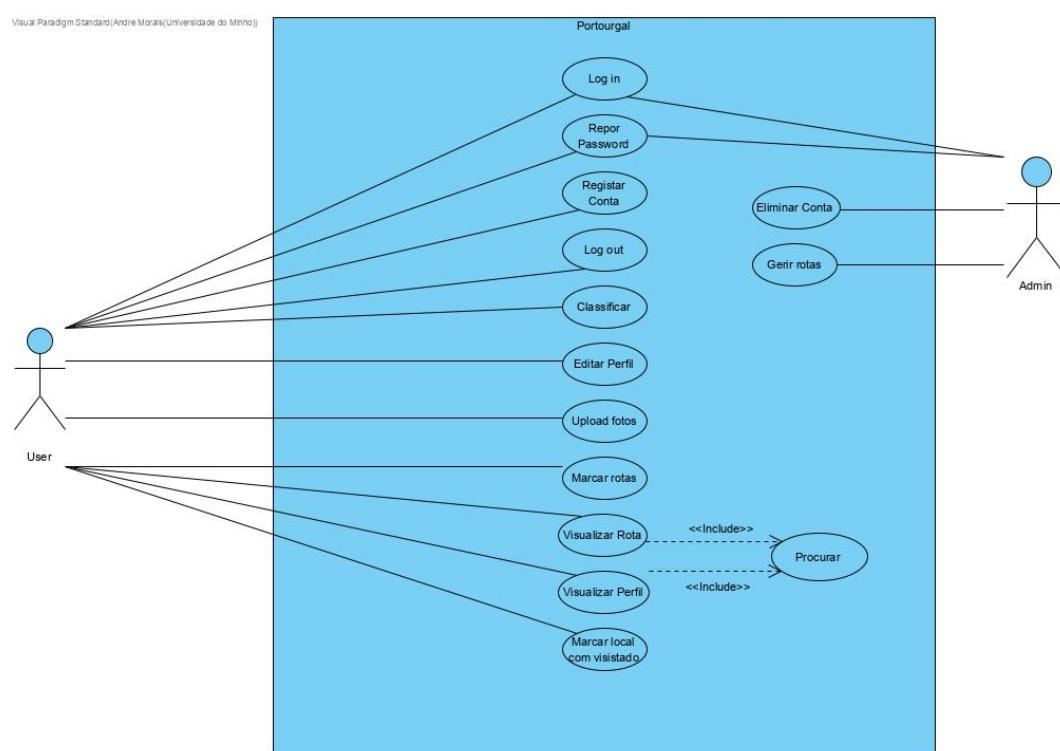


Figura 2: Diagrama de Use Cases inicial

Depois de uma nova e mais profunda análise aos requisitos funcionais do sistema e aos Mockups desenvolvidos, decidimos que alguns Use Cases não iriam ser representados, portanto fomos removendo esses Use Cases do nosso diagrama. Removemos os use cases relativos à Search Bar, pois não seria necessário dado a nossa implementação do Front End da aplicação, isto é, a nível de apresentação da aplicação não iria ser necessário esta funcionalidade. O use case Upload de fotos poderia quebrar regras de segurança da aplicação, e teríamos que implementar um sistema de verificação de fotos, portanto decidimos, após uma conversa com o docente da U.C, retirar este use case.

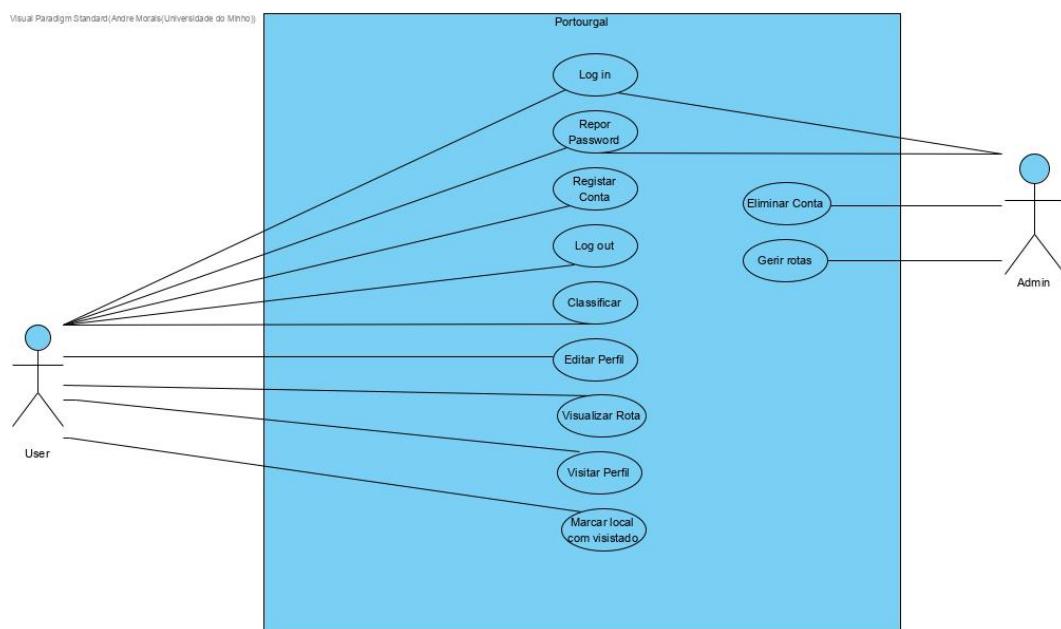


Figura 3: Diagrama de Use Cases Final

2.5 ESPECIFICAÇÃO USE CASES

Todos os Use Case necessitam dos seus determinados fluxos, estando estes apresentados abaixo.

2.5.1 Criar Conta/Registrar Conta

Descrição: Consiste no ato de criar conta, em que o utilizador insere os seus dados para criar a conta associada a ele.

Pré-condição: A aplicação está a funcionar.

Pós-condição: O Sistema regista o novo User.

Fluxo Normal:

1. Utilizador insere as suas credencias.
2. Sistema verifica se os campos estão vazios.
3. Sistema verifica se já existe email.
4. Sistema atualiza e regista novo utilizador.

Fluxo Alternativo: [Campos vazios] (passo 2)

- 2.1 Sistema informa que os campos estão vazios.
- 2.2 Utilizador insere dados nos campos vazios.
- 2.3 Volta passo 2.

Fluxo Alternativo: [Email já existe] (passo 3)

- 3.1 Sistema informa que já existe o email inserido.
- 3.2 Utilizador insere novo email.
- 3.3 Volta passo 3.

2.5.2 Visualizar Perfil

Descrição: Utilizador visita perfil de outros utilizadores

Pré-condição: Utilizador está autenticado.

Pós-condição: Utilizador visita perfil de outro utilizador.

Fluxo Normal:

1. Utilizador carrega na página ScoreBoard.
2. Sistema carrega ScoreBoard.
3. Utilizador carrega na foto do Utilizador que pretende visitar.
4. Sistema carrega perfil do utilizador a visitar.

2.5.3 Login

Descrição: Consiste em se autenticar na aplicação PORTOURGAL.

Pré-condição: A aplicação está a funcionar.

Pós-condição: Utilizador está autenticado.

Fluxo Normal:

1. Sistema pede credenciais ao utilizador.
2. Utilizador insere email e palavra passe.
3. Sistema verifica credenciais inseridas.
4. Utilizador autentica-se.

Fluxo Alternativo: [Credenciais erradas] (passo 3)

3.1 Sistema informa que as credenciais estão erradas.

3.2 Volta para o passo 1.

2.5.4 Logout

Descrição: Consiste em se desconectar da aplicação PORTOURGAL.

Pré-condição: Utilizador/Admin está autenticado.

Pós-condição: Utilizador/Admin deixa de estar autenticado.

FLuxo Normal:

1. Utilizador escolhe fazer logout.
2. Sistema informa que o utilizador está agora desconectado.

2.5.5 Repor Password

Descrição: Consiste no ato de repor a palavra passe do User.

Pré-condição: O utilizador está autenticado.

Pós-condição: O utilizador tem nova password..

FLuxo Normal:

1. Utilizador escolhe editar o perfil.
2. Utilizador insere nova palavra passe.
3. Sistema atualiza e associa nova palavra passe.

2.5.6 *Classificar Atração*

Descrição: Consiste no ato do User classificar uma Atração.

Pré-condição: O utilizador está autenticado.

Pós-condição: Sistema adquire nova classificação de uma atração.

Fluxo Normal:

1. Utilizador procura Atração que pretende classificar.
2. Utilizador escolhe classificação a atribuir.
3. Sistema atualiza com nova classificação inserida.

2.5.7 *Edita Perfil*

Descrição: Consiste no ato de editar credenciais de um User.

Pré-condição: O utilizador está autenticado.

Pós-condição: O utilizador tem lhe associado novas credenciais.

Fluxo Normal:

1. Utilizador escolhe editar o perfil.
2. Utilizador insere novas credenciais que pretende mudar.
3. Sistema verifica se User mudou email.
4. Sistema verifica se já existe email.
5. Sistema atualiza com as novas credenciais.

Fluxo Alternativo: [Email já existe] (passo 4)

- 4.1 Sistema informa que já existe o email inserido.
- 4.2 Volta passo 2.

2.5.8 *Visualizar Rota*

Descrição: Consiste no ato do User visualizar uma Rota do Sistema.

Pré-condição: O utilizador está autenticado.

Pós-condição: Sistema mostra rota procurada.

Fluxo Normal:

1. Utilizador procura Rota que pretende visualizar.
2. Sistema mostra Rota.

2.5.9 Marcar local como visitado

Descrição: Consiste no ato do User marcar como visitado uma Atração.

Pré-condição: O utilizador está autenticado.

Pós-condição: Sistema adquire nova visita de um local/atração.

Fluxo Normal:

1. Utilizador procura Atração que pretende marcar como visitada.
2. Utilizador marca a atração como visitada.
3. Sistema atualiza.

2.5.10 Remover Conta

Descrição: Administrador remove conta existente de um user.

Pré-condição: Administrador está autenticado.

Pós-condição: Sistema remove conta do Utilizador.

Fluxo Normal:

1. Administrador insere email do Utilizador.
2. Sistema verifica se existe o email.
3. Sistema atualiza e remove o Utiliador.

Fluxo Alternativo: [Email já existe] (passo 2)

- 2.1 Sistema informa que conta já não existe.

2.5.11 Gerir Rotas

Descrição: Administrador gere rotas do Sistema.

Pré-condição: Administrador está autenticado.

Pós-condição: Sistema atualiza com novas rotas.

Fluxo Normal:

1. Administrador adiciona ou remove rota.
2. Sistema atualiza.

2.6 DIAGRAMA DE CLASSES

Decidimos implementar um Diagrama de Classes pois são diagramas estruturais que permitem ilustrar as classes e o relacionamento entre as mesmas. Assim, estes permitem descrever a estrutura de um sistema, modelando ainda os atributos e as operações entre objetos.

Nesse sentido, a obtenção deste diagrama resultou de uma análise cuidada do modelo de domínio, identificando assim, quais as classes suscetíveis de serem criadas.

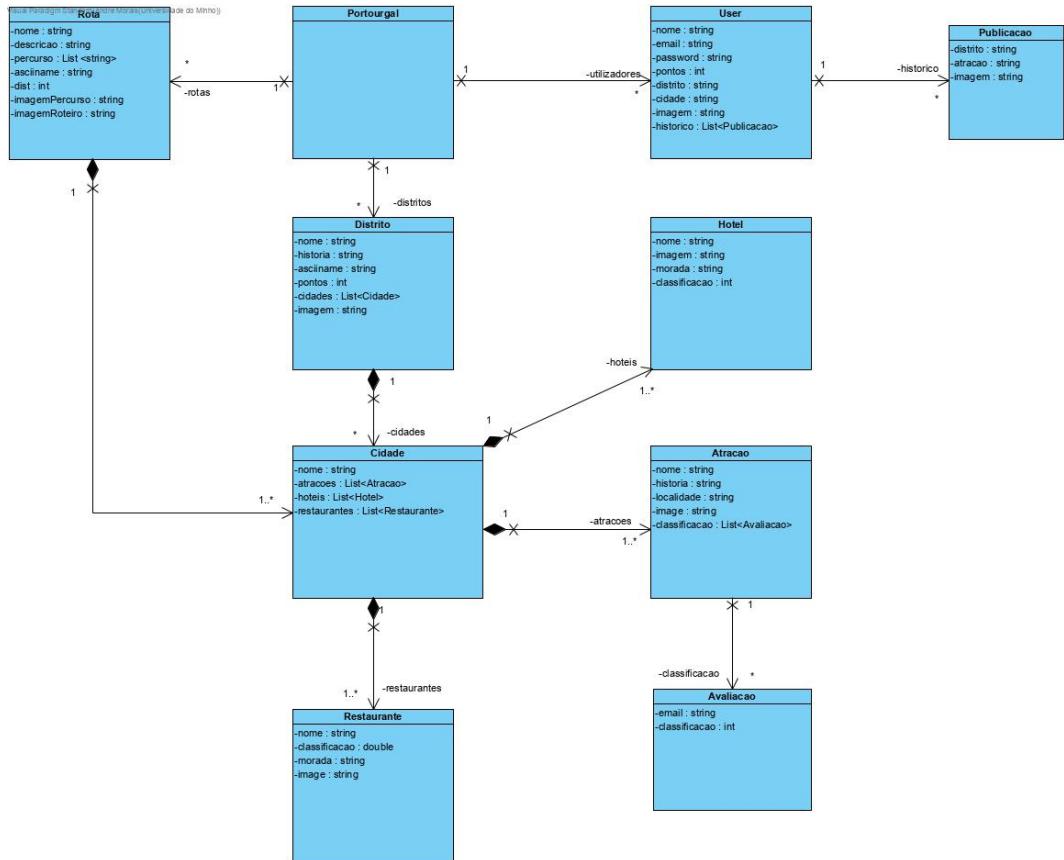


Figura 4: Diagrama de classes.

3

IMPLEMENTAÇÃO

3.1 FUNCIONALIDADES

3.1.1 Página Inicial



Figura 5: Página inicial.

Esta é a página inicial da aplicação, sendo esta apresentada sempre que o utilizador queira utilizar a aplicação.

3.1.2 Login

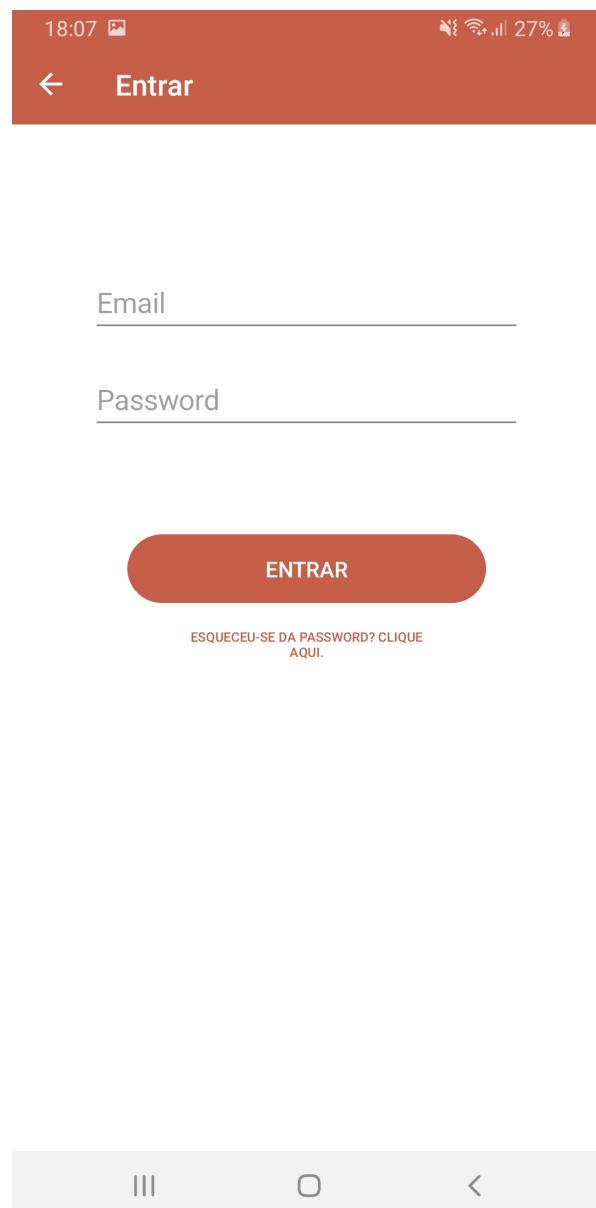


Figura 6: Página de login.

3.1.3 Registrar

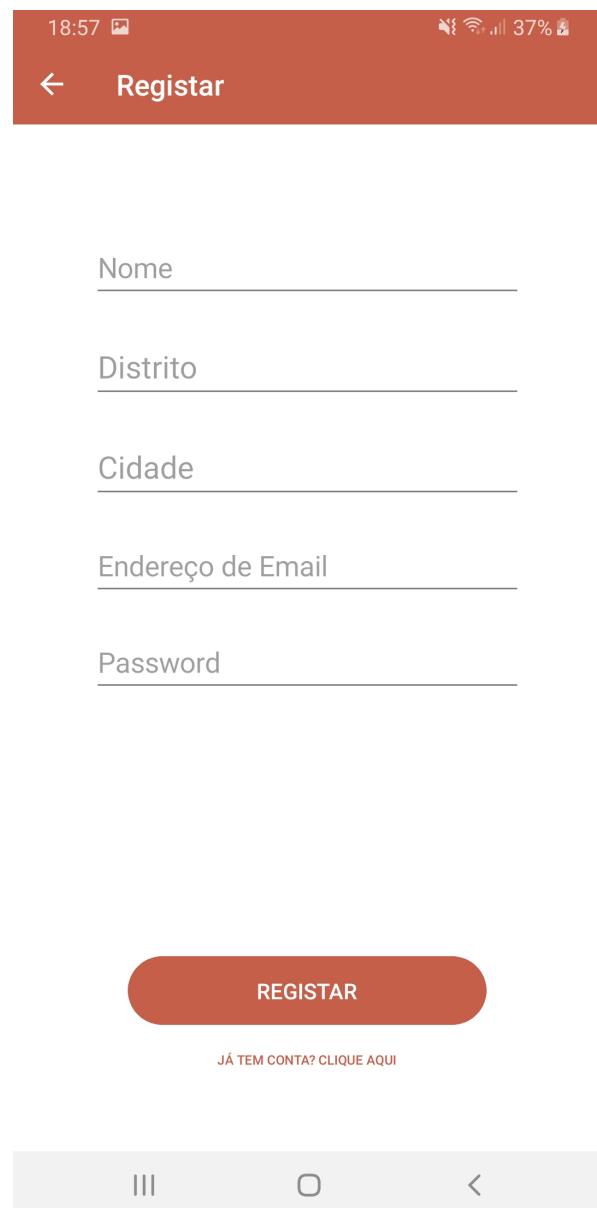


Figura 7: Página de registo

3.1.4 Perfil de usuário

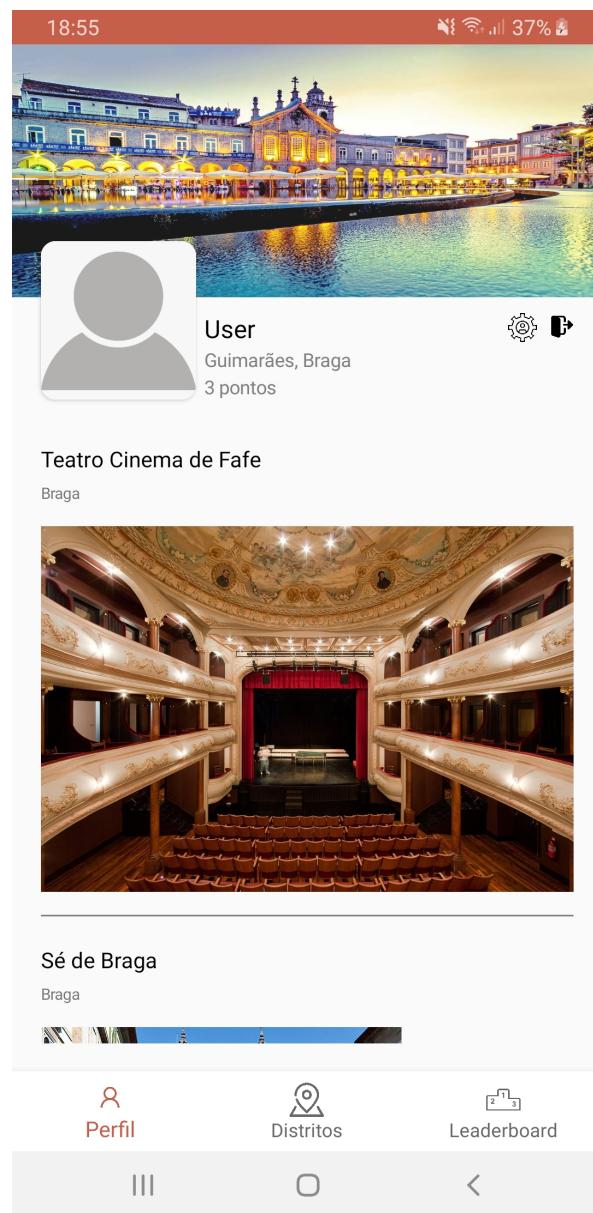


Figura 8: Página de perfil

3.1.5 Editar perfil

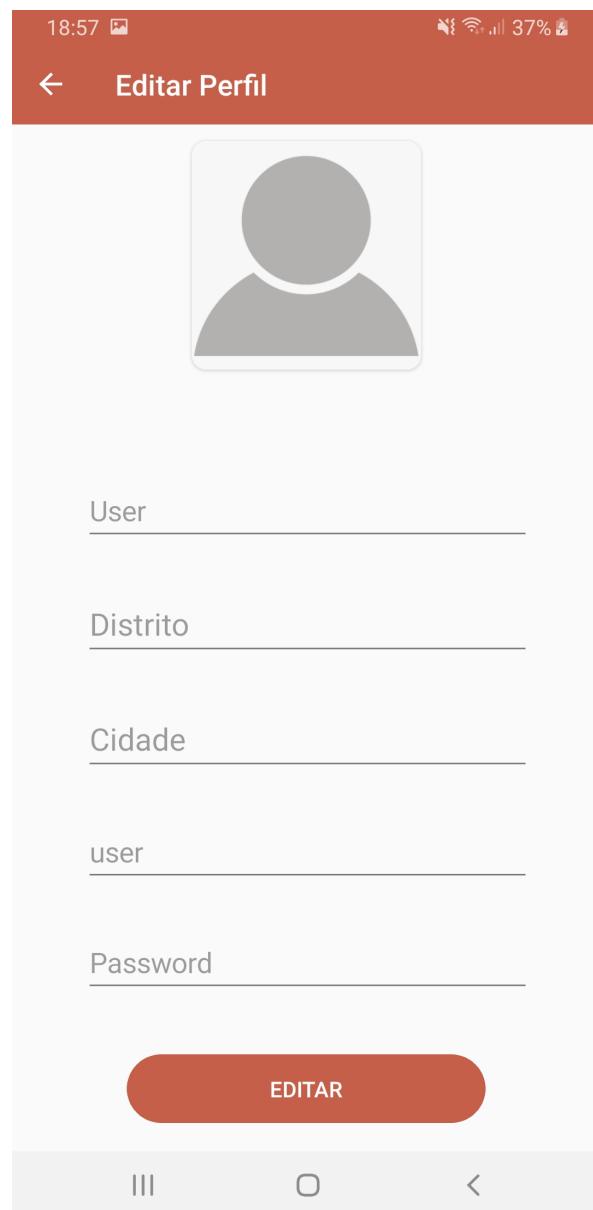


Figura 9: Página para editar o perfil.

3.1.6 Upload de uma imagem

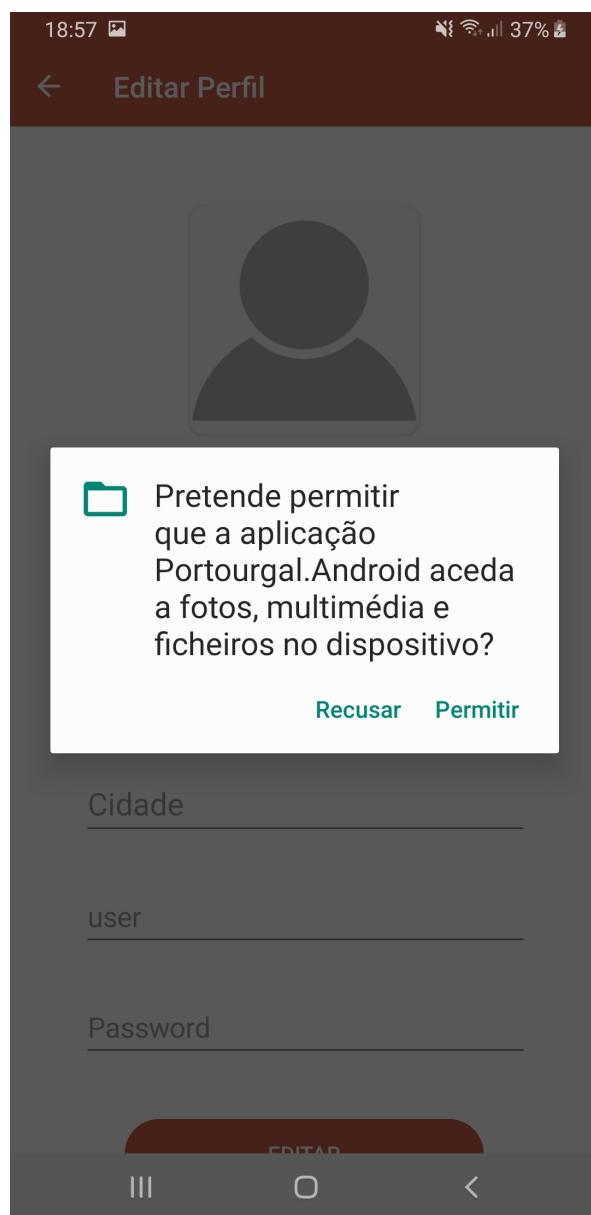


Figura 10: Anexar uma imagem para foto de perfil.

3.1.7 Navegar pelos distritos/roteiros

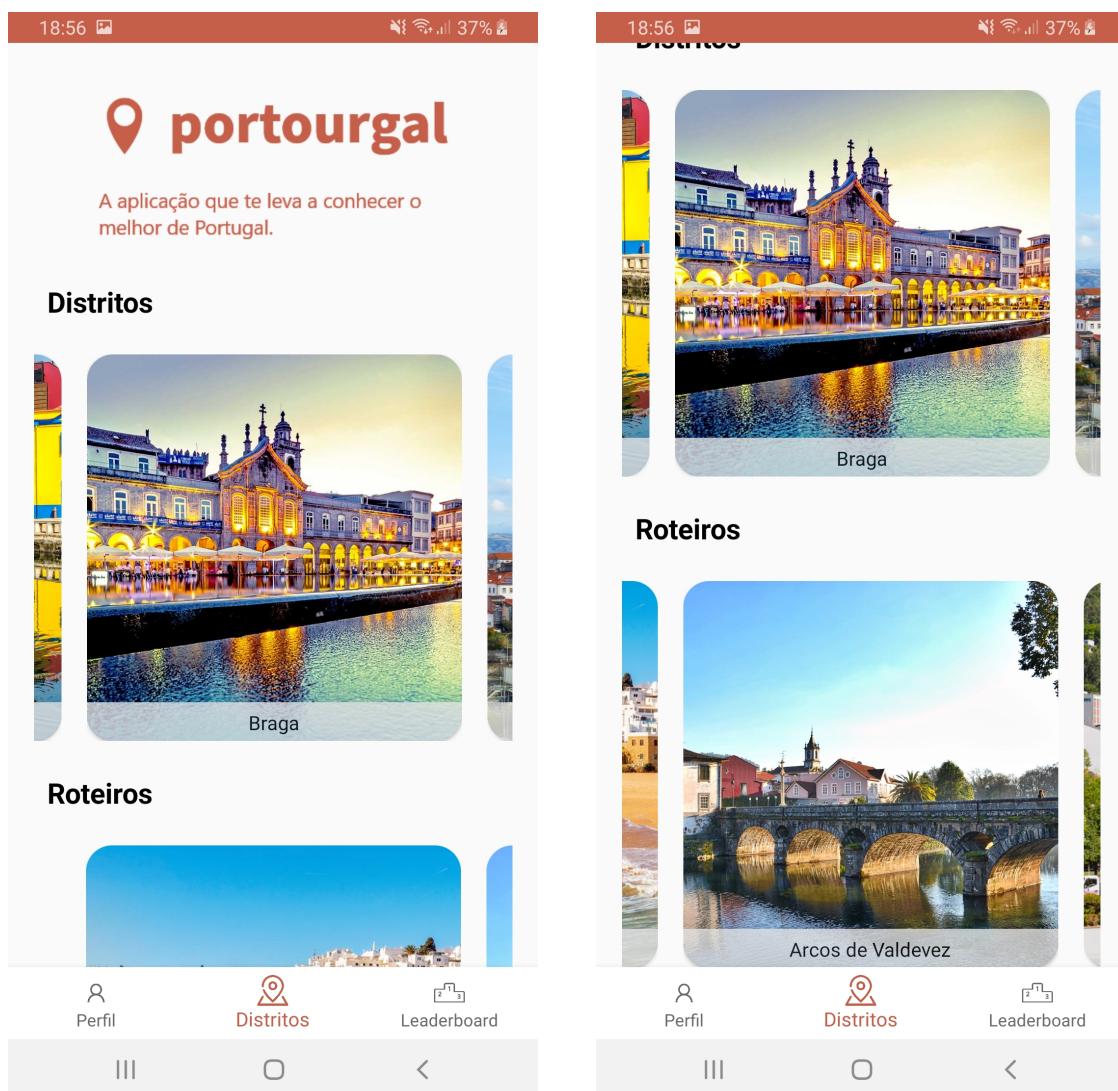


Figura 11: Página dos distritos e roteiros.

3.1.8 Conteúdo de um distrito

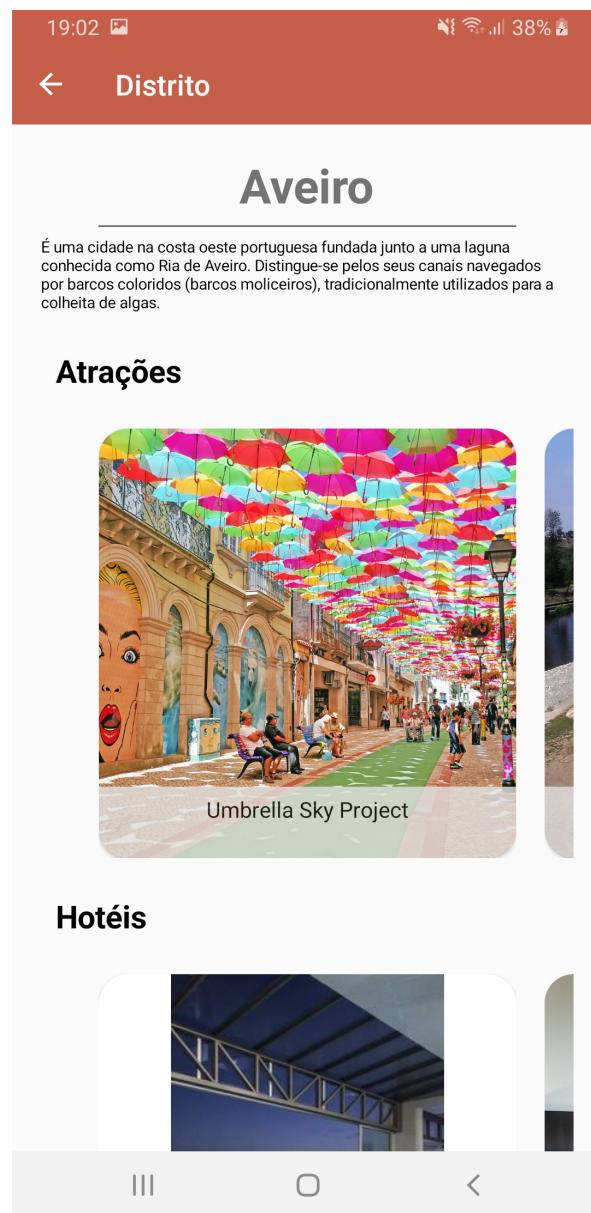


Figura 12: Exemplo de um distrito.

3.1.9 Atração

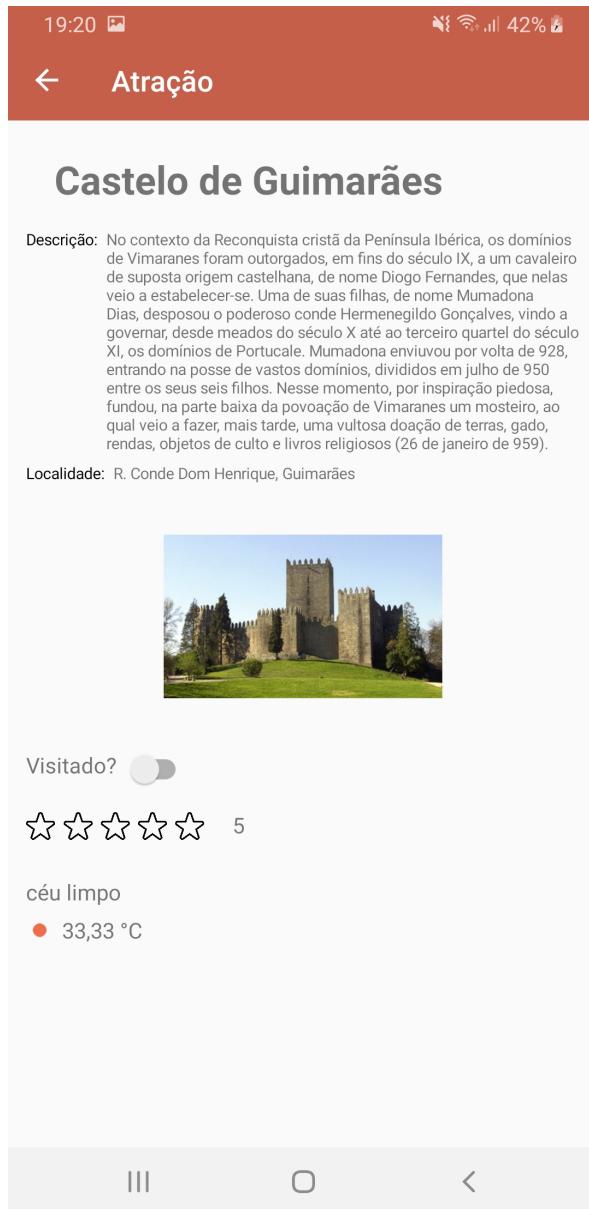


Figura 13: Exemplo de uma atração.

3.1.10 Percurso

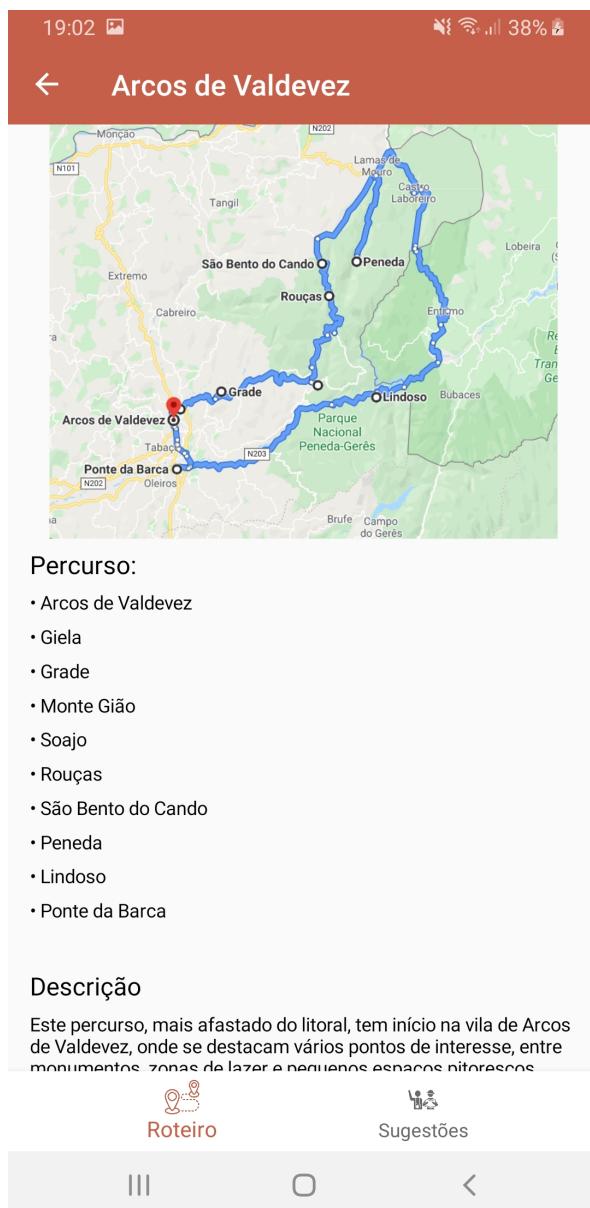


Figura 14: Exemplo de um percurso.

3.1.11 Sugestões para um percurso



Figura 15: Sugestões para um percurso exemplo.

3.1.12 Leaderboard

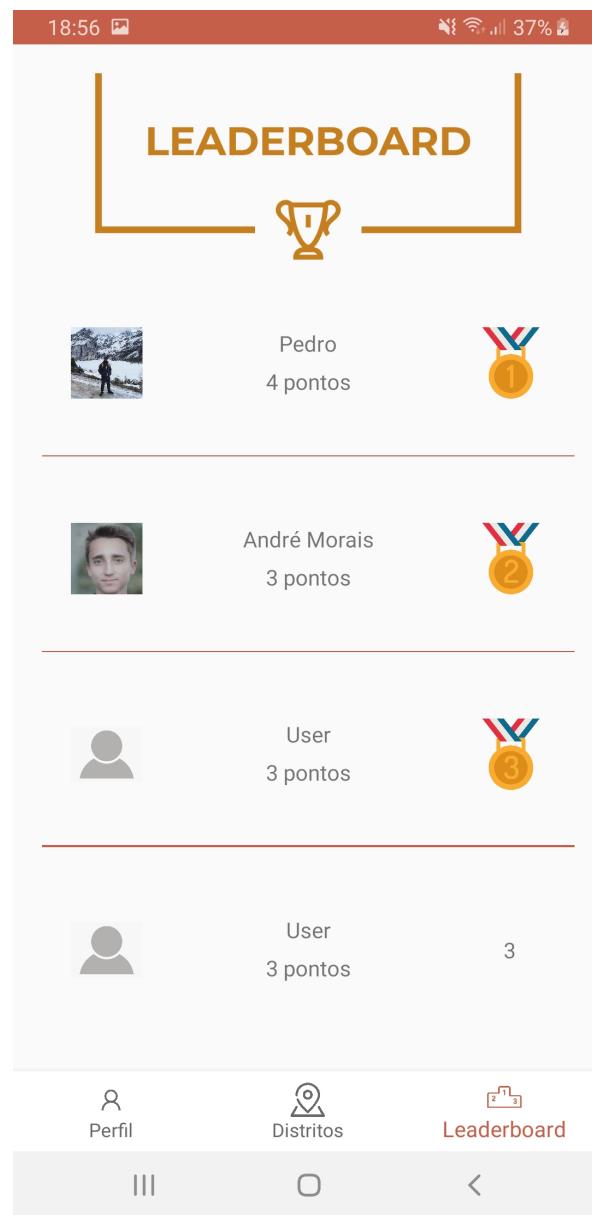


Figura 16: Tabela de classificações.

3.2 BASE DE DADOS - ABORDAGEM NÃO RELACIONAL

3.2.1 Justificação da utilização de um sistema NoSQL

Os sistemas de bases de dados relacionais assumem uma posição de predominância no mercado. Tal deve-se principalmente à eficácia do armazenamento dos dados estruturados. É importante referir que este tipo de sistemas preservam a consistência, integridade, isolamento dos dados e durabilidade.

Contudo, surgiram problemas que conduziram ao aparecimento de sistemas de bases de dados não relacionais (NoSQL) que foram idealizados atendendo às lacunas que as bases de dados tradicionais demonstraram, com alta performance e capacidade de expansão. Estas, em vez de armazenarem os dados em tabelas, utilizam estruturas dependendo do tipo da base de dados, podendo estas serem, grafos, colunas, chave-valor e documentos.

3.2.2 MongoDB

MongoDB é uma base de dados NoSQL baseada em documentos. As suas características permitem que as aplicações modelem informações de modo mais natural, pois os dados podem ser agrupados em hierarquias complexas e continuarem a ser indexáveis e de fácil acesso.

Para além disto:

- Permitem também uma maior escabilidade face às relacionais
- Sendo os requisitos da base de dados na sua maioria de natureza simples, o MongoDB terá uma eficiência maior apresentando os resultados em tempo inferior aos do MySQL.
- Sendo a base de dados multi-user, ou seja, utilizada por vários utilizadores ao mesmo tempo, o MongoDB apresenta também melhor performance no que toca ao acesso concorrente à base em questão.

3.2.3 Implementação BD

Depois de um estudo às diferentes abordagens para a utilização de uma base de dados, ficou decidido que a utilização de um modelo não relacional, mais concretamente MongoDB, era o mais indicado.

Foi então necessário definir coleções para fundamentar a base de dados, nomeadamente:

- Distritos
- Roteiros

- Users

De modo a que seja possível o bom funcionamento da aplicação, devem ser respeitados alguns campos para definir cada elemento presente nas coleções da base de dados.

3.2.3.1 *Distrito*

- string Nome → Identifica o distrito pelo seu nome próprio, é necessário para apresentar o distrito na aplicação;
- string ASCIIName → Corresponde ao nome do distrito sem quaisquer caracter especial, tais como espaços, acentos e "ç". Importante para ser possível fazer chamadas à API devido aos pedidos HTTP;
- int Pontos → Número de pontos que o utilizador obtém sempre que visitar uma atração deste distrito;
- string História → Uma breve descrição sobre o distrito em si;
- Lista de Cidades → Lista com todas as cidades deste distrito. Estas cidades são objetos que guardam informações sobre as mesmas:
 - string Nome → Identifica a cidade pelo seu nome próprio;
 - Lista de Atrações → Indica todas as atrações da cidade:
 - * string Nome → Identifica a atração pelo seu nome próprio;
 - * string Localidade → Indica a local onde a atração está presente;
 - * string História → Breve descrição do local;
 - * string Imagem → URL representativo da imagem;
 - * Lista Classificações → Lista com todas as classificações feitas a esta atração:
 - string Email → Email do utilizador que realizou a classificação;
 - string Classificação → Número representativo da classificação feita pelo utilizador.
 - Lista de Restaurantes → Indica todos os restaurantes da cidade:
 - * string Nome → Identifica o restaurante pelo seu nome próprio;
 - * string Morada → Tal como o nome indica, representa a morada do mesmo;
 - * double Classificação → Classificação do restaurante;
 - * string Imagem → URL representativo da imagem do restaurante.
 - Lista de Hotéis → Indica todas as hotéis da cidade:

- * string Nome → Identifica o hotel pelo seu nome próprio;
- * string Morada → Tal como o nome indica, representa a morada do mesmo;
- * int Classificação → Classificação do restaurante;
- * string Imagem → URL representativo da imagem do restaurante.
- string Imagem → URL da imagem para ser apresentada mais tarde

3.2.3.2 Utilizador

- string Nome → Nome próprio do utilizador, apresentado no seu perfil;
- string Email → Email do utilizador, deve ser único;
- string Distrito → Distrito onde o utilizador vive;
- string Cidade → Cidade onde o utilizador vive;
- string Password → Password do utilizador, utilizada na sua autenticação. Deve ser segura e intransmissível;
- string Imagem → Imagem de perfil do utilizador, representada em base 64;
- int Pontos → Número de pontos conquistados pelo utilizador ao visitar diferentes atrações;
- Lista de Publicações → Representa o histórico do utilizador, que representa as atrações que o mesmo visitou:
 - string Distrito → Indica o distrito onde fica a atração que o utilizador visitou;
 - string Atração → Indica a atração visitada pelo utilizador;
 - string Imagem → URL representativo da imagem da atração.

3.2.3.3 Roteiro

- Lista de strings com Percurso → Representa o percurso do roteiro, com a lista dos locais por onde ele passa;
- string Descrição → Breve descrição do roteiro;
- string Imagem Roteiro → URL com imagem apelativa do roteiro;
- string Imagem Percurso → URL com a imagem representativa do percurso do roteiro;
- string Nome → Nome identificativo do roteiro;

- string ASCII → Nome do roteiro sem caracteres especiais, tais como espaços, acentos e "ç", de forma a ser possível realizar chamadas à API com o nome do roteiro. Deve ser único;
- int Dist → Indica a distância do percurso.

3.2.4 Acesso à Base de Dados

A aplicação desenvolvida pelo grupo deve estar funcional 24 horas durante 7 dias por semana, para isto, é necessário haver disponibilidade no acesso à base de dados durante todo este período de tempo. Por isso, houve necessidade de dar *host* da base de dados na internet, para que fosse possível o acesso constante e remoto a toda a informação necessária para o funcionamento da aplicação.

A plataforma utilizada para manter a base de dados disponível todo o tempo foi **MongoDB Atlas**.

3.3 RESTAPI - COMUNICAÇÃO ENTRE CLIENTE-SERVIDOR

3.3.1 RestAPI

REST é um acrônimo para REpresentational State Transfer e deve seguir 6 princípios fundamentais:

1. **Client-server** - "*By separating the user interface concerns from the data storage concerns, we improve the portability of the user interface across multiple platforms and improve scalability by simplifying the server components;*"
2. **Stateless** - "*Each request from client to server must contain all of the information necessary to understand the request, and cannot take advantage of any stored context on the server. Session state is therefore kept entirely on the client;*"
3. **Cacheable** - "*Cache constraints require that the data within a response to a request be implicitly or explicitly labeled as cacheable or non-cacheable. If a response is cacheable, then a client cache is given the right to reuse that response data for later, equivalent requests;*"
4. **Uniform interface** - "*By applying the software engineering principle of generality to the component interface, the overall system architecture is simplified and the visibility of interactions is improved. In order to obtain a uniform interface, multiple architectural constraints are needed to guide the behavior of components. REST is defined by four interface constraints: identification of resources; manipulation of resources through representations; self-descriptive messages; and, hypermedia as the engine of application state;*"
5. **Layered system** - "*The layered system style allows an architecture to be composed of hierarchical layers by constraining component behavior such that each component cannot "see" beyond the immediate layer with which they are interacting;*"
6. **Code on demand (optional)** - "*REST allows client functionality to be extended by downloading and executing code in the form of applets or scripts. This simplifies clients by reducing the number of features required to be pre-implemented;*"

3.3.2 Implementação

Devido às necessidades da aplicação, foi necessário desenvolver uma RestAPI para controlar toda a informação que circula de um lado para o outro (cliente - servidor).

Foi então desenvolvida uma RestAPI, com recurso a tecnologias Microsoft (ASP.NET, .NET, C#)

Para isto foram implementados 3 controladores diferentes: **Distritos, Utilizadores e Roteiros.**

3.3.3 Endpoints

A nossa RestAPI oferece os seguintes endpoints:

- /api
 - /distritos
 - * /nome
 - **GET** /:nomeASCII → Devolve o distrito com nomeASCII (representa o nome sem quaisquer caracter especial);
 - **PUT** /:nomeASCII → Atualiza o distrito com nomeASCII. Necessita dos dados do novo distrito;
 - **DELETE** /:nomeASCII → Elimina o distrito com nomeASCII.
 - * **GET** / → Devolve todos os distritos presentes na base de dados;
 - * **GET** /:id → Devolve o distrito com id enviado como parametro;
 - * **POST** / → Insere um distrito na base de dados;
 - * **PUT** /:id → Atualiza um distrito consoante o seu id;
 - * **DELETE** / → Elimina um distrito da base de dados. Necessita dos dados do distrito a eliminar;
 - * **DELETE** /:id → Elimina um distrito da base de dados consoante o seu id.
 - /users
 - * /email
 - **GET** /:email → Devolve o utilizador com email enviado como parametro;
 - **PUT** /:email → Atualiza o utilizador consoante o seu email;
 - **DELETE** /:email → Elimina o utilizador consoante o seu email.
 - * **GET** / → Devolve todos os utilizadores presentes na base de dados;
 - * **GET** /:id → Devolve o utilizador consoante o seu id;
 - * **POST** / → Insere um utilizador na base de dados;
 - * **PUT** /:id → Atualiza um utilizador consoante o seu id;
 - * **DELETE** / → Elimina um utilizador da base de dados;
 - * **DELETE** /:id → Elimina o utilizador da base de dados consoante o seu id.
 - /roteiros
 - * /nome

- **GET** /:nomeASCII → Devolve o roteiro consoante o seu nomeASCII;
 - **PUT** /:nomeASCII → Atualiza o roteiro consoante o seu nomeASCII;
 - **DELETE** /:nomeASCII → Elimina o roteiro consoante o seu nomeASCII.

* **GET** / → Devolve todos os roteiros presentes na base de dados;

* **GET** /:id → Devolve um roteiro consoante o id do mesmo;

* **POST** / → Insere um roteiro na base de dados;

* **PUT** /:id → Atualiza um roteiro consoante o seu id;

* **DELETE** / → Elimina um roteiro da base de dados;

* **DELETE** /:id → Elimina um roteiro consoante o seu id.

3.3.4 Demonstração

GET https://portourgalapi2020.azurewebsites.net/api/distritos/

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies (1) Headers (8) Test Results

Pretty Raw Preview Visualize JSON ↻

```
1 [ { 2   "Id": "5eff68e2bc0f9c2b408d635a", 3   "Nome": "Aveiro", 4   "ASCIIName": "Aveiro", 5   "Pontos": 2, 6   "Historia": "É uma cidade na costa oeste portuguesa fundada junto a uma laguna conhecida como Ria de Aveiro. Distingue-se pelos seus canais navegados por barcos coloridos (barcos moliceiros), tradicionalmente utilizados para a colheita de algas.", 7   "Cidades": [ 8     { 9       "Nome": "Águeda", 10      "Atracoes": [ 11        { 12          "Nome": "Umbrella Sky Project", 13          "Localidade": "R. Luís de Camões 111, 3750-156 Águeda", 14          "Historia": "", 15          "Imagen": "https://www.impactplan.pt/multimedia/1/FOTOS/306/9301513285308A_1920x1070.jpg", 16          "Classificacao": [ 17            { 18              "Email": "kadapiroagrande@dona.helena", 19              "Classificacao": 5 20            } 21          ], 22        }, 23        { 24          "Nome": "Parque Fluvial de Bolfiã", 25          "Localidade": "3750-064 Águeda", 26        } 27      ] 28    } 29  ] }
```

Figura 17: GET /distritos.

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
{
  "Id": "5eff68e2bc0f9c2b408d635a",
  "Nome": "Aveiro",
  "ASCIIName": "Avéiro",
  "Pontos": 2,
  "Historia": "É uma cidade na costa oeste portuguesa fundada junto a uma laguna conhecida como Ria de Aveiro. Distingue-se pelos seus canais navegados por barcos coloridos (barcos moliceiros), tradicionalmente utilizados para a colheita de algas.",
  "Cidades": [
    {
      "Nome": "Águeda",
      "Atrações": [
        {
          "Nome": "Umbrella Sky Project",
          "Localidade": "R. Luís de Camões 111, 3750-156 Águeda",
          "Historia": "",
          "Imagen": "https://www.impactplan.pt/multimedia/1/FOTOS/306/9301513285308A_1920x1070.jpg",
          "Classificacao": [
            {
              "Email": "kadapirogrande@dona.helena",
              "Classificacao": 5
            }
          ]
        },
        {
          "Nome": "Parque Fluvial de Belfiar",
          "Localidade": "3750-064 Águeda",
          ...
        }
      ]
    }
  ]
}

```

Figura 18: GET /distritos/nome/:nomeASCII.

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
1,
{
  "Id": "5f01e67c55239a2d7c0cc46b",
  "Percorso": [
    "Albufeira",
    "Paderne",
    "Alte",
    "Torre",
    "Rocha da Pena",
    "Salir",
    "Fonte de Benémola",
    "Querença",
    "Loulé",
    "São Lourenço",
    "Vilamoura",
    "Albufeira"
  ],
  "Descricao": "A zona de Albufeira é uma das mais turísticas do Algarve - Todos os anos, as suas praias e as dos arredores são invadidas por hordas de turistas nacionais e estrangeiros. No entanto, o percurso que lhe proponhos afasta-se dessa abordagem mais tradicional, com o intuito de lhe mostrar alguns recantos naturais, ainda escondidos e alguns locais de interesse histórico que, acreditamos, vale a pena conhecer.",
  "ImagenRoteiro": "https://d1oelfd4hfth9.cloudfront.net/wp-content/uploads/albufeira2.jpg",
  "ImagenPercorso": "https://i.ibb.co/Wggc1jf/image.png",
  "Nome": "Albufeira",
  "ASCII": "Albufeira",
  "Dist": 92
}

```

Figura 19: GET /roteiros.

GET https://portourgalapi2020.azurewebsites.net/api/roteiros/nome/VianaCastelo

Send Save

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies (1) Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {  
2     "Id": "5f01e68255239a2d7c0cc472",  
3     "Percurso": [  
4         "Viana do Castelo",  
5         "Montedon",  
6         "Afife",  
7         "Vila Praia de Âncora",  
8         "Moledo",  
9         "Caminha",  
10        "Vilar de Mouros",  
11        "Arga de Cima",  
12        "Orbacem",  
13        "Perre"  
14    ],  
15    "Descrição": "O traçado deste percurso começa e acaba em Viana do Castelo, sem dúvida uma das mais belas cidades portuguesas. Após um passeio exploratório pela cidade, aconselha-se uma subida ao Monte de Santa Luzia, que lhe dará uma perspectiva panorâmica de toda a região. Depois, seguindo a linha costeira a norte de Viana, terá ao seu dispor algumas praias agradáveis, onde, dependendo da ocasião escolhida e da disposição do momento poderá estender a toalha ou, simplesmente, fazer um pequeno (mas revitalizante) passeio à beira mar. Mais para o interior, encontrará algumas aldeias e vilas integradas numa paisagem bastante atrativa, que causa uma sensação de bem estar e convide, inevitavelmente ao descanso... Boas curvas!",  
16    "ImagenRoteiro": "https://ncultura.pt/wp-content/uploads/2020/02/vianay.jpg",  
17    "ImagenPercurso": "https://i.ibb.co/kHcZDxt/unknown.png",  
18    "Nome": "Viana do Castelo",  
19    "ASCII": "VianaCastelo",  
20    "Dist": 145  
21}
```

Figura 20: GET /roteiros/nome/:nomeASCII.

```
GET https://portourgala2020.azurewebsites.net/api/users/
```

No Environment

Send Save Bulk Edit

KEY	VALUE	DESCRIPTION
Key	Value	Description

Status: 200 OK Time: 3.25 s Size: 1.11 MB Save Response

Body Cookies (1) Headers (8) Test Results

Pretty Raw Preview Visualize JSON

1 "Id": "5f009868bc0f9c2b408d6371",
2 "Name": "Pedro",
3 "Distrito": "Braga",
4 "Cidade": "Barcelos",
5 "Email": "pedro@mail.com",
6 "Password": "XXXXXXXXXX",
7 "Imagen": "/9j/4OBXRpxzgAATUOAkAAQAAAABAAAPgESAQAAAABAAAAAAEBAQAAAABAAJADQ0EAAAQAAAABAAJADQ0AAAAAIIAQAQAAAABAAAAAAAD/4AAQSkZJrgABAQAAAQABAAD/
8 2xCAEAICAgICAgICAgIDAwDIAwDIBAEYBQAYBQYFQCFQgBzHcQgOwCwJ3Qw4QDwQDwMIEBMRHMyxgfhYoBaICAgICAgICAgDwBgDwBANDA=MeBgQEBgkBfgUJhCkAHBwcJcA4LcQkLDhN
9 DAQ0cExRExRgxXGB8FKv/CABE01001QWp1gAeCE0EFQH/AA3AABBAmpAQEAQAAAABAAQAAAABAAQHBAUICfCQeACgFgAAgBQAEQAAAABAAQAAAAGhQAOQgBf/g2gMAwEuaDAEAAAQzm3P0nD15/+
SeebhffHzyib8dttgFbd8JLlTQPKewHnkrkjRMH21NF0/nXQYhCtC1ekbj+Tjkj0sRUBCf81/nlFFwFHf17r/4/tHubKqQnzaqD53h1o21/L1t1btvz2bhpS/5t9k3LwJ
+grPn0j750mArFqpu18gyv10SMHw7ut0r9Tk01zzPx80KNN1795tcmbrtwq8+fhghdh7rh7MP0d21ryodv+jfd/fe/lmn7pkU44eXan03ZnyP66CgfdZ
+1SnTyfjH23ta1qlomymNCFxrxINP3519JS/HMX4+IlySL50bXn9yH5XUD1dOnefu0r+b+24Vm0h5W+1khkqha1/c/rnHmFnAh0rJabm5ByFV5+xFhJlgzXjbPurK
+0wPgbpmbZdmvXdsX5v0Gv6UDXUOXL1at0r9Tk01zzPx80KNN1795tcmbrtwq8+fhghdh7rh7MP0d21ryodv+jfd/fe/lmn7pkU44eXan03ZnyP66CgfdZ
+EjJns5u9NSV2V+avtVzu0d32x5a0nlnPfJ7t1c5s0ad0fW0hN/
0a500dcZ3jz5Z3H0p15tGaU77t0a401f77ye1x0K0ESt0ys+Dgt4ls9a10Fp2XauH145tl3y5b0z8r11NuJ25t26SPNOzCV4n3pmjPgGbSuqyVzQbwNw8QHURGIOd/5ksDxEx8jaNj00yc
+umolouawuY03Ty1Gg1KypG1gRa51Xd280w0of7msk5cygb4qSkw1jPBpB92ey02yevwpzlXs91/cr3ab5ThKvFbVn52zrhffJHf2X1z1/X/
Q8d2077XrGyK1u9AT9RNhZ5Dj0ms2b50n2g5QkHvHlH2j47F7p3EDYdvgvFQEcEdmheVau1UcezePYJ3VKFDY41V4TwH91isgbyfcFaqQhKlzblC4Xpd6lAc1/2VBnEdPhw8Va1/
Xhr5HE1YR1dc12Vg/vPE02/a9kn330n26NTgnBgv0WVFPIDAsB3gL626051Dw017d01L1TAJYdLEVSsosqAxERKhoE8sb1piyvE7B10d0ur01sVey11udZhnsU/
+hxpeF7f11085C1UgrUrUyvhypbuaFpcZMcG+T3jPkcvkTzskhPoPYK1j3
+VoZfUp2162518KwfgRrbtsb13L1t79519914yBCllP27Ph07H9Gvc73539j18t1kz1kRf5J3xt11mzsukar4yewImjuoTKX4qvF8arTaa2PlHo5y5ErKKKGBeSu15MjubLew51v01Tg35XndPc4In2Pwt0
Dlu1k3Nxtr128v9le32EG06C6h1yrr477t01108051nECFWeov1jptdutpe3RMwv+add3v2IVUvcs57Hv24Van tGllumhsPtVhuo04m5TC6U2gYXlW1n0ve0b1Duck8Ds/
DS1CTVjce10HnK1iwhKwTHIychw0Cm2y2wL0mCMNQc7tupF61Kv8An1n7w5HFW2t1z7t1RahCe1z5011MSKRI0KcEESRvz1m/Q1M6Covar1b0PFhBF81Cpd67w9/
+bzVg1Hzzkckfd7tupF61Kv8An1n7w5HFW2t1z7t1RahCe1z5011MSKRI0KcEESRvz1m/Q1M6Covar1b0PFhBF81Cpd67w9/
+12H21qAbp1vz2u97a2u4Fy4qjLq3G06Z3qfCg/X1LnH6FeCyg1Cgla1hvN0uOp3sYarQfZnjwzD117HvhB9u2Bbnov+e23l6y6n1Y1m6
+OnJTsJ09A1uTd6eMA1s2mm0Njwz3Jc/H504952T71m5zwegfpXwLHmkybDf9w2Llh13s3/4/yFakso120Q2w1s/

Figura 21: GET /users.

```

1 {
2   "Id": "5f020c6571b3991378eafb14",
3   "Nome": "User",
4   "Distrito": "Braga",
5   "Cidade": "Guimarães",
6   "Email": "user",
7   "Password": "???m?c??i?□??ui???5Hm?□?□?[??",
8   "Imagen": "",
9   "Pontos": 9,
10  "Historico": [
11    {
12      "Distrito": "Aveiro",
13      "Atracao": "Peloninho de Serém",
14      "Imagen": "https://www.cm-agueda.pt/thumbs/cmagueda/uploads/writer_file/image/1556/plr_SerC_m_b_1_1024_2500.jpg"
15    },
16    {
17      "Distrito": "Aveiro",
18      "Atracao": "Parque Fluvial de Bolfiar",
19      "Imagen": "https://1.bp.blogspot.com/-drBKX12w-mg/VeYLNgXVvmI/AAAAAAAAPBE/yct4L1qDUk/s1600/WP_000660.jpg"
20    },
21    {
22      "Distrito": "Aveiro",
23      "Atracao": "Umbrella Sky Project",
24      "Imagen": "https://www.impactplan.pt/multimedia/1/FOTOS/306/9301513285308A_1920x1070.jpg"
25    },
26    {
27      "Distrito": "Braga",
28    }
29  ]
30}

```

Figura 22: GET /users/nome/:nomeASCII.

3.3.5 API externa

Tal como previsto na idealização do projeto, foi utilizada uma API externa que fornece dados sobre a meteorologia em tempo real. Esta API foi utilizada para apresentar ao cliente o estado do tempo, bem como a temperatura, nas atrações de um determinado distrito.

A API escolhida foi **Open Weather Map API**, que mantém a sua documentação em <http://openweathermap.org/current>.

Na aplicação desenvolvida fora utilizada duas chamadas à API externa:

1. **weather?q=[Distrito],PT&lang=pt&units=metric&appid=[API Key]** - Retorna todo o tipo de informações acerca da meteorologia no local. Os campos utilizados foram a descrição, a temperatura, e o código do icon;
2. **/img/wn/[Código Icon].png** - Devolve a imagem representativa do código de icon devolvido pela chamada anterior.

3.4 CLIENTE - SERVIDOR - BASE DE DADOS

Depois de implementada a base de dados e o modo de comunicação entre cliente e servidor, podemos resumir o tráfego presente no nosso sistema com a Figura 23.

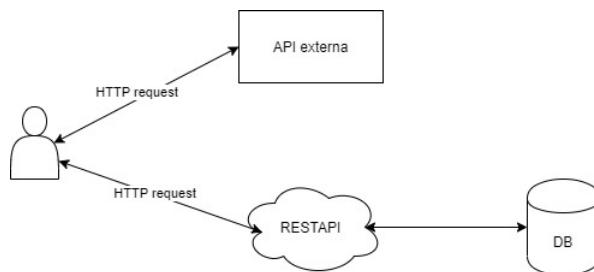


Figura 23: Tráfego do Sistema.

3.5 SEGURANÇA

A segurança e privacidade são dois conceitos fundamentais no mundo atual. Refletindo sobre isto, percebemos que era necessário implementar algo para esconder a password dos utilizadores.

Uma password nunca deve ser guardada nem enviada em *plain text*, então decidimos implementar um mecanismo, capaz de manter a integridade e funcionamento da aplicação, mas que aumentasse a segurança e privacidade de um utilizador nesse sentido.

3.5.1 Implementação

Para isso foi utilizada uma hash, que, antes de enviar um HTTP POST request para criar o utilizador, é aplicada essa hash à password do cliente.

Posto isto, o campo Password no cliente, leva já a password escondida por uma hash, aumento assim a segurança no enviado da password para a API. No final, é guardada a password escondida na base de dados.

Quando o utilizador pretende autenticar-se, a aplicação envia um HTTP GET request para a API a pedir as informações do utilizador com o email indicado pelo cliente, aplica a função de hash à password enviada pelo cliente e compara com o campo Password guardado na base de dados, caso sejam iguais, o cliente fica autenticado.

Este mecanismo aumenta a segurança porque, tal como indicado anteriormente, não há qualquer envio de dados em que o campo Password esteja em *plain text*. É importante também reforçar que estas funções de hash não contêm inversa, logo são irreversíveis, sendo impossível obter a password a partir da hash da mesma.

3.6 ADMINISTRAÇÃO DO SISTEMA

Refletindo sobre o possível crescimento da aplicação, foi criado um simples website, de forma a ser possível administrar e controlar todo o sistema.

Neste website, é possível eliminar e criar distritos, cidades, atrações, restaurantes, hotéis e roteiros. E ainda é possível eliminar utilizadores.

Esta implementação, apesar de muito simples, pode vir a ser muito útil e importante, num futuro onde a aplicação cresça. Caso isto aconteça, é necessário aumentar o nível de segurança do próprio website utilizado para administrar a aplicação.

3.6.1 Demonstração

The screenshot shows a simple web interface. At the top, there is a navigation bar with links: 'PortourgalAdmin', 'Utilizadores', 'Distritos', and 'Roteiros'. The main content area has a title 'Utilizadores'. Below the title, there is a list of four items, each preceded by a small blue bullet point:

- pedro@gmail.com
- andre_morais98@hotmail.com
- user
- amotetaipas@gmail.com

Figura 24: Visualização de todos utilizadores.

PortourgalAdmin Utilizadores Distritos Roteiros

Utilizador

Nome
Pedro

Email
pedro@gmail.com



Distrito
Braga

Cidade
Barcelos

Pontos
7

Histórico

- Câmara Municipal de Alfândega da Fé, Bragança
- Castelo de Estremoz, Évora
- Jardim do Paço, Castelo Branco
- Igreja do Bom Jesus da Cruz, Braga
- Umbrella Sky Project, Aveiro

[Eliminar Utilizador](#)

Figura 25: Visualização de um utilizador.

PortourgalAdmin Utilizadores Distritos Roteiros

Distritos

- Aveiro
- Braga
- Bragança
- Castelo Branco
- Coimbra
- Évora
- Faro
- Guarda
- Leiria
- Lisboa
- Portalegre
- Porto
- Santarém
- Setúbal
- Viana Castelo
- Vila Real
- Viseu
- Beja

[Criar novo distrito](#)

Figura 26: Visualização de todos os distritos.

PortugalAdmin Utilizadores Distritos Roteiros

Distrito

Nome
Braga

ASCII
Braga

Pontos
1

História
É uma cidade no extremo norte de Portugal, a nordeste do Porto. É conhecida pela herança e eventos religiosos. Braga possui uma história bimilenar que se iniciou na Roma Antiga, quando foi fundada em 16 a.C. como Bracara Augusta em homenagem ao imperador romano Augusto (r. 27 a.C.–14 d.C.).



Cidades

- Amares
- Braga
- Cabeceiras
- Celorico
- Espinho
- Fafe
- Guimaraes
- Povoa de Lanhoso
- Terras de Bouro
- Vieira do Minho
- Vila Verde
- Vizela
- Barcelos

[Criar nova Cidade](#)

[Eliminar Distrito](#)

Figura 27: Visualização de um distrito, contendo todas as cidades.

PortugalAdmin Utilizadores Distritos Roteiros

Distrito

Nome
Barcelos

Atrações

- Museu de Olaria
- Igreja do Bom Jesus da Cruz

[Criar nova atração](#)

Restaurantes

- Casa dos Arcos
- Restaurante Vera Cruz
- Turismo Restaurante Lounge

[Criar novo restaurante](#)

Hotéis

- Art'otel Barcelos
- Hotel Bagoeira

[Criar novo hotel](#)

[Eliminar Cidade](#)

Figura 28: Visualização de uma cidade, contendo todas as atrações, restaurantes e hotéis.

PortugalAdmin Utilizadores Distritos Roteiros

Atração

Nome
Igreja do Bom Jesus da Cruz

Localidade
Barcelos

História
A sua origem está ligada ao chamado Milagre da Cruz, prodígio religioso ocorrido em 20 de dezembro de 1504, ligado ao surgimento de uma cruz negra no carnvalal do Campo da Feira. Com o intuito de proteger essa cruz, foi erguido um pequeno templo de planta quadrada, coberto por uma abóbada e aberto por quatro arcos. Em data incerta, os arcos foram fechados e ao seu redor erguida uma arcada coberta com telhado.



Classificação

- 2, andre_moraes98@hotmail.com

[Eliminar Atração](#)

Figura 29: Visualização de uma atração.

PortugalAdmin Utilizadores Distritos Roteiros

Nova Atração

Nome

Localidade

História

Imagen

[Enviar](#)

Figura 30: Criação de uma nova atração.

PortugalAdmin Utilizadores Distritos Roteiros

Roteiros

- Albufeira
- Arcos de Valdevez
- Braga
- Chaves
- Lisboa
- Porto
- Santarém
- Viana do Castelo
- Vieira do Minho

[Criar novo roteiro](#)

Figura 31: Visualização de todos os roteiros.

PortugalAdmin Utilizadores Distritos Roteiros

Roteiro

Nome
Braga

ASCII
Braga

Descrição
Este percurso atravessa uma das zonas mais marcantes da província do Minho. Começando em Braga, a cidade das tradições seculares, segue rumo aos primórdios da nacionalidade até Guimarães, passando pela Cítânia de Bríteiros e pelas Taipas. A seguir, detém-se em Vizela e um pouco mais em Santo Tirso, até chegar a São Miguel de Seide, onde recordamos Camilo. Antes de voltar a Braga, ainda há ocasião de passar por terras de Tibães, onde, ao que parece, os monges beneditinos se fixaram há cerca de mil anos.

Distância:
102 km

Percorso:

- Braga
- Bríteiros
- Caldas das Taipas
- Guimarães
- Vizela
- Santo Tirso
- São Miguel de Seide
- Rio de Tibães
- Braga




[Eliminar Roteiro](#)

Figura 32: Visualização de um roteiro.

3.7 IDENTIFICAÇÃO DOS RECURSOS UTILIZADOS

Ao nível do desenvolvimento da aplicação foi necessário dispor de recursos humanos, de hardware e de software. Os recursos humanos englobam os trabalhadores envolvidos no projeto e as horas dedicadas neste mesmo. Quanto aos recursos de hardware indispensáveis para o desenvolvimento da “PORTOURGAL” foram as máquinas pessoais dos elementos do grupo. Para além destes aspetos, foram utilizadas as seguintes ferramentas no decorrer do desenvolvimento do projeto:

- **Visual Paradigm** – plataforma usada para a modelação do projeto;
- **Latex** - plataforma usada para a escrita do relatório;
- **MongoDB Atlas** - plataforma usada para a persistência de dados;
- **Microsoft Visual Studio** - ambiente de desenvolvimento das aplicações .NET;
- **Microsoft .NET C** - plataforma usada para a codificação da aplicação;
- **Adobex XD** - ferramenta de design para a construção dos mockups;
- **Postman** - plataforma usada para facilitar as chamadas à API;
- **Weather API** - API para obter o clima;
- **Microsoft Azure** - plataforma usada para dar host à API;
- **Google Booking** - usado para o povoamento da Base de Dados.

4

CONCLUSION

Ao longo da realização deste trabalho prático, o grupo foi-se deparando com as diversas fases na produção de aplicações, esforçando-se para ultrapassar todas as dificuldades que delas advêm.

Para a primeira fase – a **Fundamentação** – foram reunidos os pilares fundamentais do projeto: Contextualizou-se o problema, definiram-se os objetivos a cumprir, estudou-se a viabilidade, complexidade e utilidade do sistema a desenvolver, e de seguida listamos os recursos que serão necessários para as subsequentes fases. Esta fase é fulcral pois catalisam desde já a criação de um caminho para a compreensão e modelação com sucesso do sistema de Software.

O plano de desenvolvimento, que foi apresentado na forma de um Diagram de Gantt, foi uma ajuda preciosa pois permitiu organizar e quantificar o número de horas de trabalho necessárias para o desenvolvimento da peça de software proposta.

Na segunda fase – a **Especificação** – procedeu-se à modelação do projeto assente nos princípios ajustados na fase anterior. Primeiramente foram identificados os requisitos não funcionais e funcionais. De seguida, elaboraram-se alguns diagramas UML considerados relevantes para a compreensão do sistema a ser desenvolvido, tais como:

- Modelo de Domínio
- Diagrama Use Cases
- Diagrama de Classes

O Modelo de Domínio é claro e conciso. Esta propriedade, assim como a identificação correta dos diferentes papéis dos relacionamentos entre as entidades, vão de encontro aos objetivos que traçamos no começo da concessão do projeto. Relativamente ao Diagrama de Use Cases, permitiu capturar requisitos funcionais, fornecendo assim uma notação diagramática que permite modelar o contexto geral do sistema. O Diagrama de Classes permite ilustrar as classes e o relacionamento entre as mesmas, daí ser tão importante implementar.

Para lá da simplificação do problema, adicionalmente, estes modelos, contribuíram para obtermos uma ideia final clara e objetiva de todo o sistema de software implementado.

Estruturou-se a base de dados sólida e esboçaram-se mockups, idealizando a interface e camadas de apresentação do sistema.

Para a terceira e última fase deste projeto – a **Implementação** –, debruçámo-nos sobre a sobre a produção do software. Tendo por base todo o trabalho desenvolvido referente à modelação, criou-se uma aplicação Web que considerámos cumprir as diretrizes mais relevantes estabelecidas pelo cliente.

O grupo deparou-se com uma linguagem de programação que não tinha sido trabalhada até ao momento, ferramentas de produção de software desconhecidas e um paradigma de desenvolvimento novo. Todos estes fatores contribuíram para que o planeamento não fosse cumprido como tinha sido idealizado na primeira fase de projeto, levando a que este fosse ocasionalmente alterado. Apesar do grupo perceber que deveria ter seguido à risca o plano idealizado no início do projeto (o que não foi possível por não termos experiência com este tipo de aplicações e linguagem), também soube ter um lado mais prático e pro-ativo, por forma a ter um produto final (inacabado) para entregar ao cliente.

Assim sendo, – e apesar dos obstáculos encontrados - o grupo pensa que o seu trabalho foi positivo. O produto final é funcional, satisfaz todos os Use Cases feitos e, acima de tudo, é flexível e utilizável o que sugere que trará bastante entusiasmo aos futuros utilizadores da aplicação.