



**Universidade do Minho**  
Escola de Engenharia  
Mestrado Integrado em Engenharia Informática

## Unidade Curricular de Laboratórios de Informática IV

Ano Letivo de 2020/2021

### NearYou

Luís Faria(A93209) Alexandre Martins(A93315) Gonçalo Santos(A93279)  
Gonçalo Ferreira(A93218) Luís Silva (A93293)

29 de novembro de 2021

# LI4

Data de Receção	
Responsável	
Avaliação	
Observações	

## NearYou

Luís Faria(A93209) Alexandre Martins(A93315) Gonçalo Santos(A93279) Gonçalo  
Ferreira(A93218) Luís Silva (A93293)

29 de novembro de 2021

# Resumo

O presente relatório foi elaborado no âmbito da fase de planeamento e especificação de requisitos de uma aplicação, que se apresenta como um guia para pontos de interesse. Aqui, será apresentada uma listagem de pontos de interesse próximos de uma localização (quer do utilizador, quer uma localização sugerida pelo utilizador), acompanhados por reviews introduzidas pelos utilizadores.

Na **Introdução** do relatório serão abordados a Contextualização, a Motivação e Objetivos, a Justificação, a Utilidade e Viabilidade, o Estabelecimento da Identidade do Projeto, a Identificação dos Recursos Necessários, a Maquete do Sistema, a Definição de um conjunto de Medidas de Sucesso e o Plano de Desenvolvimento(Diagrama de Gantt).

A segunda secção do relatório corresponde ao **Levantamento e Análise de Requisitos**, onde são apresentados os métodos e estratégias utilizados para a recolha de requisitos, seguidos da explicitação dos requisitos recolhidos e discutidos pelo grupo. Por fim é feita uma breve análise aos requisitos criados.

Segue-se a **Especificação e Modelação de Software** onde são apresentados os diagramas UML que serão usados para melhor descrever o comportamento da aplicação. Estes dividem-se em diagramas estruturais(que definem a estrutura que a aplicação deve possuir) e diagramas comportamentais (que definem o comportamento da aplicação perante os diferentes cenários). Todos os diagramas possuem uma descrição detalhada acerca do significado e lógica por detrás do seu desenho.

Na quarta parte do relatório é descrita a **Conceção do Sistema de Dados**, onde será apresentado o diagrama de entidade relacionamento, que demonstra a estrutura do sistema de dados. São, depois, descritas todas as entidades presentes na Base de Dados e explicadas algumas das decisões tomadas pelo grupo, quanto ao sistema de dados.

Na secção **Esboços das Interfaces**, são apresentados os *mockups* da aplicação idealizada pelo grupo, bem como a especificação dos elementos necessários em cada cenário.

Por fim, na **Conclusão** o grupo resumirá e analisará criticamente o trabalho realizado.

**Área de Aplicação:** Engenharia de Software, Desenho e Arquitetura de Bases de Dados

**Palavras-Chave:** Planeamento de Software, UML, Pandemia, Cultura, Restauração, Turismo, Guias Turísticos;

# Índice

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Contextualização . . . . .	1
1.2	Motivação e Objectivos . . . . .	2
1.3	Justificação . . . . .	3
1.4	Utilidade e Viabilidade do sistema . . . . .	3
1.5	Estabelecimento da identidade do projeto . . . . .	4
1.6	Identificação dos recursos necessários . . . . .	4
1.7	Maquete do sistema . . . . .	5
1.8	Definição de um conjunto de medidas de sucesso . . . . .	6
1.9	Plano de desenvolvimento (diagrama GANTT) . . . . .	6
<b>2</b>	<b>Levantamento e Análise de Requisitos</b>	<b>8</b>
2.1	Apresentação da estratégia e método . . . . .	8
2.2	Descrição geral dos requisitos (funcionais e não funcionais) levantados . . . . .	9
2.2.1	Requisitos Funcionais . . . . .	9
2.2.2	Requisitos Não Funcionais . . . . .	11
2.3	Validação dos requisitos estabelecidos . . . . .	12
<b>3</b>	<b>Especificação e Modelação do Software</b>	<b>13</b>
3.1	Apresentação geral da especificação . . . . .	13
3.2	Modelos estruturais . . . . .	13
3.2.1	Diagrama de Domínio . . . . .	13
3.2.2	Diagrama de Classes . . . . .	14
3.3	Modelos comportamentais . . . . .	26
3.3.1	Diagrama de <i>Use Cases</i> . . . . .	26
3.3.2	<i>Use Cases</i> . . . . .	27
3.3.3	Use Cases do Utilizador . . . . .	28
3.3.4	Use Cases do Moderador . . . . .	32
3.3.5	Use Cases do Administrador . . . . .	33
3.3.6	Diagrama de Atividades . . . . .	33
<b>4</b>	<b>Conceção do Sistema de Dados</b>	<b>37</b>
4.1	Apresentação geral da estrutura (esquema) do sistema de dados . . . . .	37
4.2	Descrição detalhada dos vários elementos de dados e seus relacionamentos . . . . .	38
4.2.1	Utilizador . . . . .	38
4.2.2	<i>Review</i> . . . . .	38
4.2.3	<i>Gosto e Report</i> . . . . .	39

4.2.4	Ponto de Interesse . . . . .	39
4.2.5	Ponto de Interesse guardado . . . . .	39
<b>5</b>	<b>Esboço dos Interfaces do Sistema</b>	<b>40</b>
5.1	Estrutura geral das interfaces do sistema . . . . .	40
5.2	Caracterização das interfaces . . . . .	49
<b>6</b>	<b>Conclusões e Trabalho Futuro</b>	<b>55</b>
	<b>Lista de Siglas e Acrónimos</b>	<b>57</b>
	<b>Anexos</b>	<b>58</b>
	Use Cases de Utilizador . . . . .	58
	Use Cases de Moderador . . . . .	64
	Use Cases de Administrador . . . . .	65

# Lista de Figuras

1.1	Artigo do Jornal Ípsilon . . . . .	2
1.2	Logótipo NearYou . . . . .	4
1.3	Maquete de Utilização do Sistema . . . . .	6
1.4	Diagrama de Gantt do planeamento do projeto . . . . .	7
3.1	Modelo de domínio . . . . .	14
3.2	Diagrama de Classes . . . . .	15
3.3	Diagrama de <i>Use Cases</i> . . . . .	27
3.4	Atividade : Adicionar uma review . . . . .	34
3.5	Atividade : Avaliação de Lista de Reports . . . . .	36
4.1	Diagrama de Entidade Relacionamento . . . . .	37
5.1	Interface de Login . . . . .	40
5.2	Interface de Signin . . . . .	41
5.3	Interface de Recuperação de Password . . . . .	41
5.4	Interface Principal para o Utilizador Comum . . . . .	42
5.5	Interface Principal para os Moderadores . . . . .	43
5.6	Interface Principal para os Administradores . . . . .	43
5.7	Interface do Ponto de Interesse . . . . .	44
5.8	Interface de Criação de Review . . . . .	44
5.9	Interface Perfil . . . . .	45
5.10	Interface de Alteração da Conta . . . . .	46
5.11	Interface de Gerência de Reportes . . . . .	46
5.12	Interface de Análise de Perfil . . . . .	47
5.13	Interface de Criação de Moderador . . . . .	48
5.14	Interface de Gerência de Moderador . . . . .	48
6.1	Criar Conta . . . . .	58
6.2	Iniciar Sessão . . . . .	58
6.3	Terminar Sessão . . . . .	59
6.4	Recuperar palavra-passe . . . . .	59
6.5	Ver Perfil . . . . .	59
6.6	Alterar Dados da Conta . . . . .	60
6.7	Apagar Conta . . . . .	60
6.8	Procurar Utilizador . . . . .	61
6.9	Ver Ponto de Interesse . . . . .	61

6.10	Pedir Caminho . . . . .	61
6.11	Guardar/Remover Ponto de Interesse . . . . .	62
6.12	Procurar Pontos de Interesse . . . . .	62
6.13	Adicionar Review . . . . .	63
6.14	Editar Review . . . . .	63
6.15	Reportar Review . . . . .	63
6.16	Alterar o "Gosto"na Review . . . . .	64
6.17	Ver Reports . . . . .	64
6.18	Criar Conta de Moderador . . . . .	65
6.19	Eliminar conta de moderador . . . . .	65

# 1 Introdução

Serve o presente relatório, para documentar o desenvolvimento de uma aplicação que se apresenta como um guia para locais de interesse, desenvolvido no âmbito da Unidade Curricular de Laboratórios de Informática IV. Este projeto terá uma duração de cerca de 3 meses e será dividido em 3 fases principais, sendo este relatório referente à primeira fase.

Na primeira fase do projeto, o grupo de trabalho terá de idealizar de forma clara e objectiva um plano para o desenvolvimento da aplicação, de forma a tornar mais fácil e eficiente todo o processo de produção de código, que será atribuído a outro grupo de trabalho, na 2ª fase do projeto.

Neste relatório, o grupo de trabalho responsável pelo planeamento e especificação será designado de "grupo de planeamento", e o grupo que tratará do desenvolvimento e implementação das especificações será designado como "equipa de implementação".

Com tudo isto mencionado o grupo de planeamento decidiu então propor uma aplicação, que fosse capaz de reconhecer a localização do seu utilizador e, a partir desta, obter os pontos de interesse das áreas da cultura, restauração, lazer, entre outras, mais próximos, sendo capaz de lhes atribuir uma pequena descrição, uma classificação, uma etiqueta do tipo de ponto de interesse e o endereço do ponto. Será também possível escrever *reviews* acerca de um dado ponto, bem como visualizar as *reviews* de outros utilizadores.

## 1.1 Contextualização

O mundo em que vivemos ficou "parado" durante cerca de dois anos devido à pandemia causada pelo vírus Covid-19, pandemia esta, que deixou estragos em diversas áreas da sociedade atual.

Em Portugal, milhares de pessoas associadas às áreas da cultura, restauração e lazer, são talvez as mais afectadas, dadas as medidas restritivas, de distanciamento social, de redução de horários e a prevalência do teletrabalho. Muitos dos proprietários destas pequenas e médias empresas estão muito próximos da falência e vão "sobrevivendo" agora com a recuperação da economia.

Retirado de um artigo do jornal "Ípsilon", um jornal criado pelo mesmo publicador do conceituado jornal "Público", o excerto da figura abaixo, demonstra a gravidade do problema, afirmando que a "consultora Ernst and Young estima em 199 mil milhões de euros as perdas



acumuladas em 2020 pela cultura”. Estes números são apenas referentes a cultura, mas para áreas como a restauração, os danos da pandemia são incalculáveis.



Figura 1.1: Artigo do Jornal Ípsilon

Outro problema muito preocupante é o impacto da pandemia na saúde mental dos portugueses. Com o confinamento e teletrabalho obrigatórios, milhares foram afectados por uma deterioração na sua saúde mental: sem a interação social do nosso dia-a-dia, sem poder sair de casa livremente e, passar do por vezes tóxico, ambiente de trabalho para o nosso espaço de conforto, são fatores que contribuem negativamente para a saúde de qualquer indivíduo.

Por estas e outras razões, é importante apoiar estas áreas tão afectadas pela pandemia.

## 1.2 Motivação e Objectivos

Atendendo ao impacto da pandemia, sentido principalmente nas áreas da cultura e restauração, como já referido anteriormente, existe uma grande necessidade de revitalizar e reerguer estes sectores de extrema importância económica para o nosso país.

Assim, a aplicação que entendemos desenvolver, apresenta-se como uma ajuda a estes sectores: acreditamos que o nosso guia poderá trazer um impacto positivo na economia nacional, encorajando os consumidores a explorarem locais nas suas cidades que talvez não conheçam, de modo a impulsionar o comércio e tornar a visita de turistas o mais agradável e enriquecedora possível.

Para além do factor económico, o nosso grupo, acredita existir um grande problema social associado: durante cerca de 2 anos foram milhares as pessoas que se viram obrigadas, pelas circunstâncias, a restringir todas as suas atividades ao conforto do seu lar; o grupo concluiu

que este factor levou à degradação da vida social e saúde mental, de vários indivíduos, que agora se vêem com maior dificuldade em sair de casa. Pretendemos que o nosso guia sirva de impulsionador à retoma de uma vida social normal, e que incentive os nossos utilizadores a saírem de casa e a explorarem o mundo que os rodeia.

## 1.3 Justificação

A aplicação será um guia para locais de interesse, que visa enriquecer o conhecimento dos seus utilizadores, acerca dos pontos de interesse que os rodeiam, pontos estes que poderão ir desde um excelente restaurante de esquina, até uma das mais belas vistas sobre a sua cidade.

Acreditamos que são poucas as aplicações a explorar este conceito e principalmente após esta pandemia, o grupo imagina existir um grande mercado de utilizadores que estariam interessados em experimentar uma aplicação deste género, devido ao grande período de confinamento que os impediu de "explorar" as suas cidades.

Esta situação, conjugada com a necessidade de recuperação financeira, levou o grupo a concluir que a aplicação a ser desenvolvida poderá vir a ter um grande impacto na nossa sociedade.

## 1.4 Utilidade e Viabilidade do sistema

O guia que pretendemos desenvolver será capaz de indicar pontos de interesse próximos do utilizador e próximos da localização fornecida por este, o que trará imensos benefícios para os turistas que querem planear, descobrir ou explorar novos sítios nas suas visitas. O guia terá também *reviews* dos pontos de interesse, que serão bastante úteis para os empresários que pretendam melhorar a experiência dos seus clientes.

Quanto à viabilidade económica, o grupo de planeamento decidiu abstrair-se um pouco desta componente, dada a vertente académica do projeto. Não obstante, o grupo reuniu algumas ideias, que poderiam tornar a aplicação mais apelativa em termos económicos:

- A aplicação poderia conter anúncios, e uma versão paga que permitisse remover estes anúncios e oferecesse mais funcionalidades
- A aplicação poderia apresentar artigos certificados, fruto de parcerias com empresas de jornalismo, que assim poderiam aumentar o tráfego no seu próprio site
- A aplicação poderia diferenciar e sugerir certos pontos de interesse, mediante acordos com as empresas que estivessem dispostas ao investimento

## 1.5 Estabelecimento da identidade do projeto

De forma a que o projeto não seja tratado apenas como um exercício académico, e estabelecendo o valor de uma aplicação para uso público, a equipa decidiu atribuir, com o intuito de destacar o propósito da aplicação na área de recomendações de pontos de interesse, o nome "NearYou".

Para melhor identificação e destaque, decidimos criar o logótipo para a nossa aplicação, "NearYou":



Figura 1.2: Logótipo NearYou

O coordenador do grupo de planeamento será o Gonçalo Ferreira, assumindo a função de mediador de comunicação entre o grupo e a equipa docente.

Como referido na justificação do nome do projeto, este terá como finalidade o desenvolvimento de uma aplicação capaz de indicar e guiar o utilizador para os pontos de interesse na região onde se localiza.

## 1.6 Identificação dos recursos necessários

É possível dividir os recursos necessários ao desenvolvimento do projeto em três categorias diferentes.

A primeira componente corresponde a ferramentas associadas ao planeamento e desenvolvimento do projeto. A linguagem de modelação UML será uma das componentes que se insere nesta categoria, sendo utilizada no desenvolvimento de vários diagramas ilustrativos do

processo de desenvolvimento e do comportamento do programa. Estes diagramas serão desenvolvidos com o auxílio do software Visual Paradigm. Para a escrita do presente relatório será utilizado o sistema de preparação de documentos  $\text{\LaTeX}$ , utilizando o editor online Overleaf, que permitirá a colaboração na escrita deste documento, entre os diversos membros do grupo. Para os esboços das interfaces, o grupo de planeamento utilizou a plataforma Proto.io para as conceber.

A segunda categoria é referente às ferramentas que são essenciais ao funcionamento do projeto, com isto referimos-nos às tecnologias, necessárias para estabelecer um fluxo de informação dentro da aplicação. A esta, correspondem, em primeiro lugar, as API que servirão de suporte a toda a aplicação. Estas terão de ser capazes de determinar a localização do utilizador e listar todos os pontos de interesse que o rodeiam, ou próximos de uma dada localização. Outra componente essencial ao projeto, será a base de dados. Esta permitirá guardar grandes quantidades de dados de forma organizada, segura e com consultas eficientes. Neste projeto, a base de dados será essencial para guardar todas as informações relativas a utilizadores e às reviews. As ferramentas de software para esta categoria ficarão à escolha da equipa de implementação.

A terceira e final categoria será referente aos recursos humanos necessários para garantirem a longevidade da aplicação. Pela análise do grupo de planeamento, serão necessárias: uma equipa de moderadores (para moderarem o conteúdo postado pelos utilizadores), um administrador (que irá gerir a equipa de moderadores) e por fim uma equipa de suporte à aplicação (responsável pela adição de novas funcionalidades e manutenção/atualização da aplicação).

## 1.7 Maquete do sistema

Neste capítulo, é apresentada uma sugestão de utilização para a nossa aplicação.

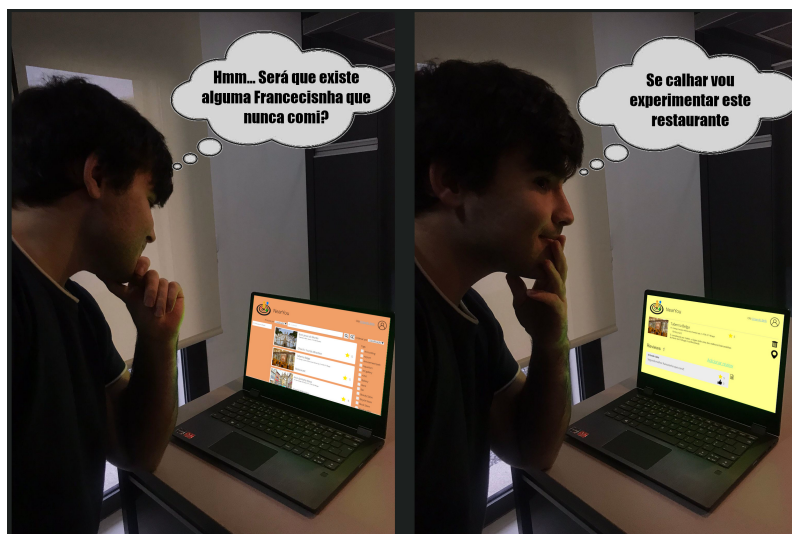


Figura 1.3: Maquete de Utilização do Sistema

## 1.8 Definição de um conjunto de medidas de sucesso

O grupo considerou os seguintes objetivos como condições para um projeto considerado de sucesso:

- O desenvolvimento e implementação do projeto devem ser concluídos dentro dos parâmetros de tempo e financiamento estabelecidos;
- A aplicação deve ser de uso fácil e intuitivo para a maior parte dos utilizadores;
- O guia deverá ser capaz de indicar pontos de interesse próximos do utilizador e de localizações fornecidas pelo mesmo;
- O guia deve ser completo, reunindo num só sítio uma grande diversidade de pontos de interesse bem como as suas respectivas descrições e avaliações.

## 1.9 Plano de desenvolvimento (diagrama GANTT)

No início de cada projeto, é importante planear o tempo dedicado a cada etapa, de forma a apresentar um produto final que seja terminado dentro dos prazos e que alcance todos os objetivos propostos. Para ajudar neste planeamento, o Diagrama de Gantt é uma excelente ferramenta que permite visualizar o tempo necessário para cada etapa, e cujo desenho implica um planeamento cuidadoso para garantir que o trabalho é dividido em etapas com um tempo compreensivo.

O grupo compromete-se a tentar cumprir todos os prazos definidos.

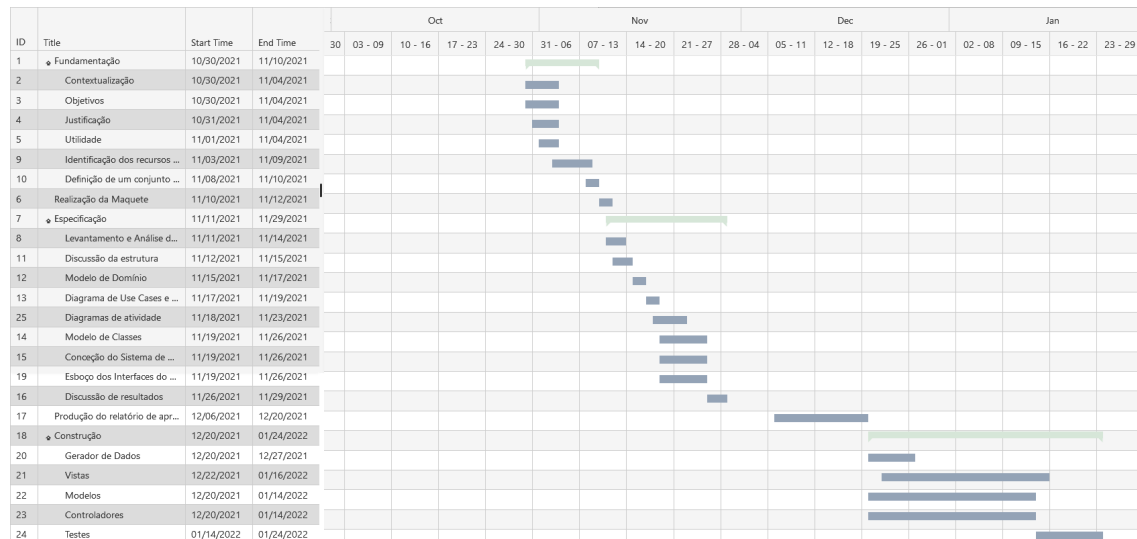


Figura 1.4: Diagrama de Gantt do planeamento do projeto

## 2 Levantamento e Análise de Requisitos

"O processo de levantamento e de análise de requisitos constitui uma das partes mais importantes (senão a mais importante) de um projeto de engenharia de software". Será nesta etapa que serão definidos em concreto todos os objetivos a alcançar para o sucesso da aplicação.

Como não temos um cliente ou qualquer outro tipo de entidade responsável por propor os requisitos, o grupo baseou-se em alguns inquéritos e nos debates internos, para criar objetivos concretos que criem uma base firme, de modo a obter um desenvolvimento eficaz e completo da aplicação.

### 2.1 Apresentação da estratégia e método

Para melhor conhecer e perceber o impacto que a aplicação proposta poderia ter na sociedade, o grupo de planeamento estudou o mercado de aplicações disponíveis, que trabalhassem na mesma área que a "NearYou" e realizou também alguns inquéritos a amigos e familiares, com uma amostra de cerca de 30 pessoas, de forma a recolher opiniões e possíveis funcionalidades para o guia proposto.

Através dos estudos realizados, o grupo concluiu:

- A principal concorrente é a aplicação fornecida pela Google, "Maps", que o grupo considera ser bastante completa. Não obstante, consideramos também que esta aplicação, não se enquadra como desejado nos requisitos do nosso guia: a aplicação deve ser muito simples de usar, deve ser compacta e deve ser apenas o que se propõe a ser, um guia que apresente de forma rápida e fácil todos os pontos de interesse perto de uma localização.
- Com os inquéritos, o grupo constata que a maioria dos entrevistados concorda com a afirmação "As áreas da cultura, restauração e lazer são das mais afectadas pela pandemia" e com a afirmação "Uma aplicação destas poderia ser uma grande ajuda à recuperação da economia". Estes inquéritos foram bastante informativos e deram novas ideias e perspectivas ao grupo, que permitiram desenvolver requisitos mais abrangentes.

## 2.2 Descrição geral dos requisitos (funcionais e não funcionais) levantados

### 2.2.1 Requisitos Funcionais

Os requisitos funcionais apresentam-se como um conjunto de afirmações acerca dos serviços que o sistema deverá oferecer, bem como ditar a forma como o sistema se deverá comportar perante situações específicas. Abaixo encontram-se todos os requisitos funcionais que o grupo acredita serem essenciais à aplicação desenvolvida.

#### Requisitos da Gestão de Utilizadores

1. A aplicação deverá ser capaz de criar contas para os seus utilizadores.
2. Ao iniciar, a aplicação deverá dar a opção ao utilizador de criar conta.
3. Para criar conta, o utilizador terá de providenciar o seu *email*, *User Name* e uma *Password*.
4. A aplicação deverá guardar numa base de dados todas as credenciais dos seus utilizadores, bem como se estão bloqueados ou não, e o seu nível de permissão.
5. Ao iniciar, a aplicação deverá pedir a autenticação do utilizador, solicitando o *email* e a *password*.
6. O utilizador deverá ser capaz de terminar a sua sessão.
7. Ao iniciar, a aplicação deverá dar a opção ao utilizador de recuperar conta.
8. A aplicação deve bloquear o *login* de contas bloqueadas por um moderador, e mostrar uma pequena mensagem informando o utilizador que foi bloqueado.
9. A aplicação deverá utilizar uma API externa para reconhecer a localização do utilizador.
10. A aplicação deverá dar a opção ao utilizador para procurar pontos de interesse noutra localização para além da atual.
11. A aplicação deverá dar capacidade aos utilizadores de adicionarem um e apenas um "gosto" a *reviews* de outras pessoas.
12. A aplicação deverá dar capacidade aos utilizadores de sinalizarem uma e uma só vez, as *reviews* que acharem inapropriadas, para um moderador as avaliar no futuro.
13. A aplicação deverá ser capaz de procurar por outros utilizadores e mostrar os seus perfis.
14. A aplicação deverá permitir ao utilizador alterar dados da sua conta.
15. A aplicação deverá permitir que um utilizador altere as *reviews* do seu perfil.
16. A aplicação deverá permitir ao utilizador remover a sua conta.
17. A aplicação deverá ser capaz de mostrar ao utilizador o seu perfil, onde deverão estar presentes as suas *reviews* e pontos de interesse guardados.



## Requisitos da Gestão de *Emails*

1. A aplicação, ao ser criada uma conta, deve enviar um *email* a notificar o utilizador que uma conta foi criada com esse *email*.
2. A aplicação, ao ser selecionada a opção de recuperar conta, deve pedir o *email* ao utilizador e, de seguida, enviar-lhe um *email* com a sua *password*.
3. A aplicação deverá notificar por *email* o utilizador quando uma *review* da sua autoria for removida.
4. A aplicação deverá notificar por *email* o utilizador quando a sua conta for bloqueada.

## Requisitos da Gestão dos Pontos de Interesse

1. A aplicação deverá utilizar uma API externa para obter pontos de interesse perto de uma localização.
2. A aplicação deverá utilizar uma API externa para procurar pontos de interesse pelo nome.
3. A aplicação deverá utilizar uma API externa para obter informações como a distância ao utilizador, uma descrição, uma fotografia, o endereço e identificadores ou *tags* desse ponto de interesse.
4. A aplicação deverá ser capaz de conseguir listar de forma ordenada quer por proximidade, quer por classificação média das *reviews*, todos os pontos de interesse próximos da localização fornecida.
5. A aplicação deverá ser capaz de aplicar filtros a lista de pontos de interesse. Os filtros disponíveis devem ser: raio de distância, *tags* e classificação (1 estrela, 2 etc).
6. A aplicação deverá mostrar a cada entrada da lista de pontos de interesse, o seu endereço, a sua classificação média, as suas *tags*, uma foto e uma breve descrição (caso exista).
7. A aplicação deverá, ao ser selecionada uma entrada na lista, expandir a informação a cerca do ponto de interesse escolhido, apresentando as suas *reviews* escritas de outros utilizadores, uma descrição mais completa, a classificação, as *tags* e uma forma de o utilizador obter as direções para este ponto de interesse a partir da sua localização.
8. A aplicação deverá ser capaz de guardar um ponto de interesse, permitindo ao utilizador visualizá-lo mais tarde.
9. A aplicação deverá ser capaz de gerar um link, que permita ao utilizador ser reencaminhado para fora da aplicação, de forma a que este obtenha as direções da sua posição para o ponto de interesse selecionado<sup>1</sup>

---

<sup>1</sup>Para criar este link usando Google maps, deve seguir o formato : "https://www.google.com/maps/dir/<latitudeOrigem>,<longitudeOrigem>/<latitudeDestino>,<longitudeDestino>"

### Requisitos da Gestão de *Reviews*

1. A aplicação deverá ser capaz de armazenar todas as *reviews* realizadas na aplicação, incluindo o identificador do utilizador, o identificador do ponto de interesse, o identificador da *review*, a *review* escrita e a classificação atribuída.
2. A aplicação deverá registar todos os "gostos" efetuado, pelo utilizador, a *reviews*.
3. A aplicação deverá registar todos os *reports* efetuados, pelo utilizador, a *reviews*.
4. A aplicação deverá alterar o identificador de todas as *reviews* de um utilizador, que removeu a sua conta, para "Utilizador Removido".
5. A aplicação deverá remover todas as *reviews* de um utilizador bloqueado.

### Requisitos da Gestão de Moderadores

1. A aplicação deverá suportar a criação de perfis moderadores que para além das funcionalidades de um utilizador comum, deverão ter acesso a mais ferramentas de moderação.
2. A aplicação deverá ser capaz de listar, apenas para os moderadores, todas as *reviews* reportadas
3. A aplicação deverá permitir a um moderador, remover pedidos de *reviews* reportadas.
4. A aplicação deverá permitir a um moderador, remover *reviews* do sistema.
5. A aplicação deverá permitir a um moderador, bloquear utilizadores.

### Requisitos da Gestão de Administradores

1. Deverá existir um perfil Administrador que possui todas as funções do moderador, com a adição da gestão de contas de moderadores.
2. A aplicação deverá permitir a um Administrador adicionar contas de moderador.
3. A aplicação deverá permitir a um Administrador remover contas de moderador.

## 2.2.2 Requisitos Não Funcionais

Os requisitos não-funcionais demonstram-se como limitações nos serviços ou funções oferecidas pelo sistema, principalmente relativos a factores externos ao sistema. Abaixo apresentam-se os requisitos não funcionais encontrados pelo grupo:

1. O utilizador precisa de possuir uma conta de *email*.
2. O equipamento onde a aplicação é executada, deverá ter acesso a *internet* e a serviços de geo-localização.
3. A aplicação deverá ser de uso fácil e intuitivo.
4. A aplicação deverá ser rápida e eficiente.
5. A aplicação deverá ser segura e *reliable* contra erros.

6. A aplicação deverá seguir as metodologias de implementação do modelo MVC (*Model View Controller*)

## **2.3 Validação dos requisitos estabelecidos**

Depois de uma atenta análise e várias versões dos requisitos, acreditamos ter alcançado uma base de orientação para as restantes fases de desenvolvimento, que alcança todos os objetivos definidos anteriormente neste relatório.

## 3 Especificação e Modelação do Software

### 3.1 Apresentação geral da especificação

Agora que os requisitos da aplicação estão bem definidos, o grupo irá basear-se em Diagramas UML para dar forma ao sistema que se pretende implementar. Estes modelos têm uma elevada importância no processo de desenvolvimento de sistemas de software, permitindo que possíveis clientes compreendam melhor as funcionalidades da aplicação e, no nosso caso em específico, servem como uma forma clara e objectiva de comunicação com a equipa de implementação, sobre quais as metas a alcançar para se obter uma implementação fiel aos requisitos e à ideia original.

Tendo isto em conta, o grupo suportar-se-á em 2 tipos de modelos: Estruturais e Comportamentais.

Os modelos estruturais demonstram a organização do sistema em termos dos seus componentes e das relações entre eles. Para estes modelos, iremos utilizar um diagrama de domínio, e um diagrama de classes.

Os modelos comportamentais ou interactivos representam a forma como o sistema deve reagir perante certos eventos ou estímulos do seu ambiente, como por exemplo a interacção com o utilizador. Os diagramas UML que irão ser usados são o diagrama de *Use Cases* e respectiva descrição dos *Use Cases*, e diagramas de Actividade.

### 3.2 Modelos estruturais

#### 3.2.1 Diagrama de Domínio

Um diagrama de Domínio é uma representação conceptual que pretende demonstrar, de forma resumida, as interações entre as entidades do sistema que se pretende implementar. Graças a esta simplicidade é uma ferramenta muito usada, na exposição de um projeto a um cliente, dado que, mesmo que este não tenha muito conhecimento técnico, consegue rapidamente compreender os relacionamentos entre as entidades

Para o modelo de Domínio consideramos relevantes as seguintes entidades:

- *Review*
- Utilizador
- Ponto de Interesse
- Moderador
- Administrador
- Gostos
- *Reports*

Depois de um estudo sobre as relações entre estas entidades, desenhamos o seguinte diagrama:

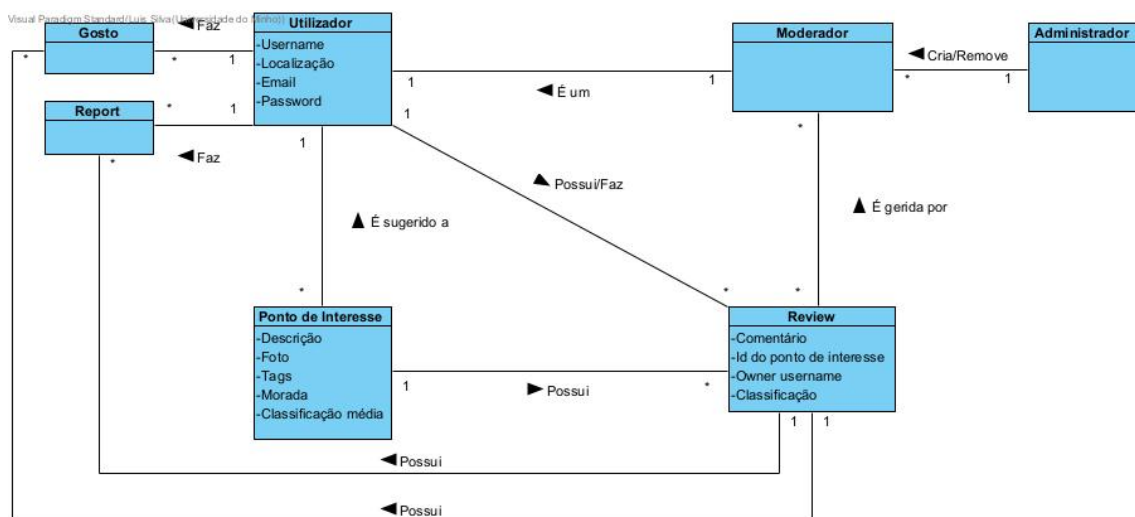


Figura 3.1: Modelo de domínio

### 3.2.2 Diagrama de Classes

O diagrama de Classes é um diagrama estrutural que descreve um sistema através das suas classes, apresentando assim em cada uma, os seus atributos, métodos e relações com as restantes classes.

Este tipo de diagrama, devido ao seu nível de detalhe e proximidade com o nível de programação, é considerado como um diagrama de desenho difícil e trabalhoso. Ainda assim, o grupo concluiu ser uma ferramenta importante para garantir a implementação idealizada de todas as funcionalidades da aplicação. O diagrama apresentado será referente apenas ao Modelo (do

modelo de implementação MVC), deixando as componentes de Vista e Controlador à mercê da equipa de implementação.

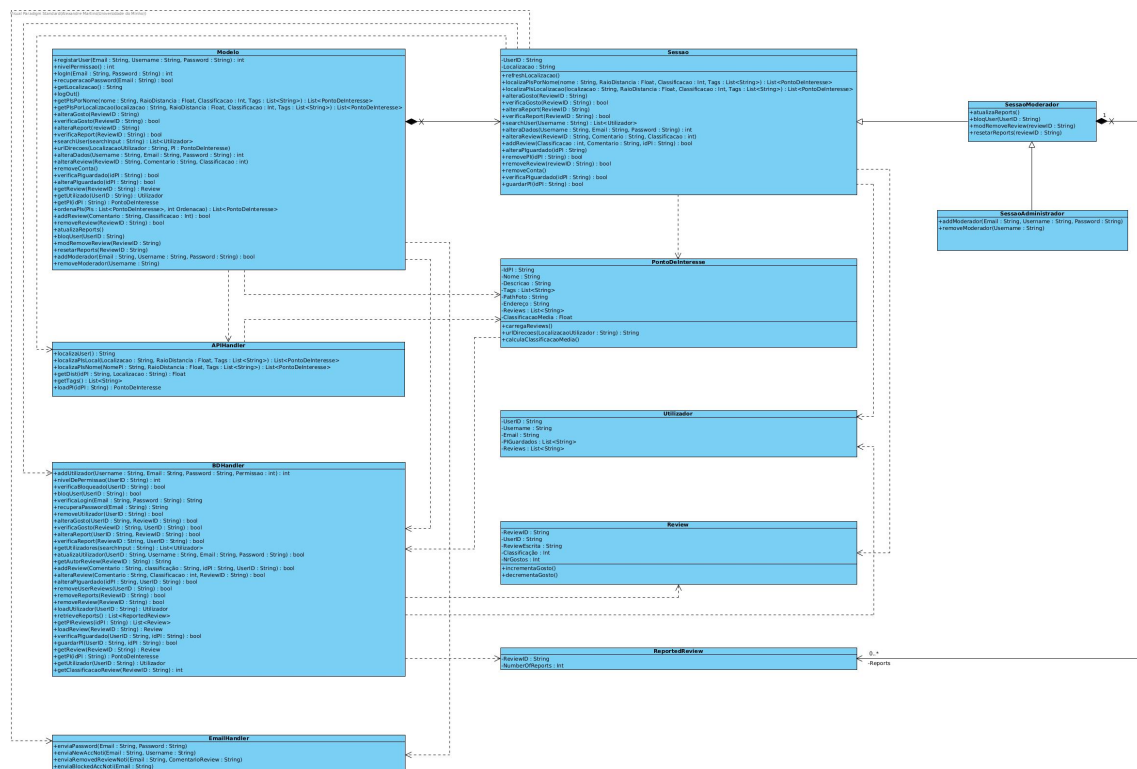


Figura 3.2: Diagrama de Classes

## Classe Modelo

**Descrição:**

Esta classe possui os métodos necessários para executar qualquer funcionalidade da aplicação.

**Atributos:**

- sessao - variável responsável por guardar a sessão atual

### Métodos:

- `registarUser` - utiliza o método `'addUtilizador'` da classe `BD Handler` para criar o novo utilizador na base de dados. É utilizado o método `'enviaNewAccNoti'`, da classe `Email Handler`, para enviar uma notificação da criação da nova conta.
- `nivelPermissao` - retorna o nível de permissão do utilizador com sessão iniciada, não é necessário aceder à base de dados, basta verificar o nome da classe `Sessao` em uso.

- `login` - recorrendo aos métodos '`verificaLogin`' e '`verificaBloqueado`' pertencentes à classe `BDHandler`, cria uma sessão para o utilizador, caso este não esteja bloqueado e as credenciais inseridas estejam corretas. Verifica qual o nível de permissão da conta, de modo a criar o tipo de sessão apropriado.
- `logout` - termina a sessão de um utilizador, i.e., elimina o objeto da classe `Sessao`
- `recuperacaoPassword` - é utilizado para enviar um email ao utilizador com a sua palavra-passe. Utiliza o método '`recuperaPassword`' (classe `BD Handler`) para obter a *password* do utilizador, que será enviada através do método '`enviaPassword`' (classe `Email Handler`)
- `getLocalizacao` - retorna a localização do utilizador em forma de string. Para a obter, chama o método '`refreshLocalizacao`' da classe `Sessão` e através de um método *getter*, obtém a localização atual.
- `getPlsPorNome` - retorna uma lista com os pontos de interesse cuja identificação se aproxime do nome fornecido. Os pontos de interesse são obtidos através do método '`localizaPlsPorNome`' presente na classe `Sessão`.
- `getPlsPorLocalizacao` - retorna uma lista com os pontos de interesse perto de uma localização. Os pontos de interesse são obtidos através do método '`localizaPlsLocalizacao`' presente na classe `Sessao`. Se a string de localização fornecida for '`null`' é usada a localização atual do utilizador.
- `alteraGosto` - utiliza o método '`alteraGosto`' da classe `Sessao`, de modo a adicionar um "gosto" a uma review, ou remover caso este já exista
- `verificaGosto` - recorre ao método '`verificaGosto`' da classe `Sessao` de modo a verificar a existência de um "gosto", do utilizador com sessão iniciada, para a review em questão
- `alteraReport` - chama o método '`alteraReport`' da classe `Sessao`, de forma a criar um report para uma review, ou eliminar caso já exista
- `verificaReport` - recorre ao método '`verificaReport`' da classe `Sessao` de forma a verificar se já existe um *report* feito pelo utilizador com sessão iniciada, à review fornecida
- `searchUser` - requisita à classe `BDHandler`, usando o método '`getUtilizadores`', utilizadores cujos dados se aproximem do *input* de procura
- `alteraDados` - pede à classe `Sessao`, que substitua os dados do utilizador pelos fornecidos, recorrendo ao método '`alteraDados`'
- `alteraReview` - recorre ao método '`alteraReview`' da classe `Sessao`, para alterar o comentário e a classificação de uma review, caso esta pertença ao utilizador
- `removeConta` - solicita a remoção da conta à classe `Sessao` (método '`removeConta`'), e

de seguida, efetua o termino de sessão

- `verificaPIguardado` - verifica se o utilizador já guardou o ponto de interesse utilizando o método '`verificaPIguardado`' da classe `Sessao`
- `alteraPIguardado` - guarda ponto de interesse na coleção do utilizador, chamando o método '`guardarPI`' da classe `Sessao`. Se já existir, remove-o.
- `getReview` - utiliza o método '`getReview`' do `BDHandler`, para obter a review com identificador fornecido
- `getPI` - utiliza o método '`loadPI`' do `APIHandler`, para obter o ponto de interesse com identificador fornecido
- `getUtilizador` - utiliza o método '`getUtilizador`' do `BDHandler`, para obter o utilizador com identificador fornecido
- `ordenaPIs` - ordena uma lista de pontos de interesse de acordo com a ordenação fornecida. Caso a '`Ordenacao`' seja 1, a lista deverá ser ordenada por proximidade, caso seja 2 deverá ser ordenada por classificação média das reviews. Retorna a lista ordenada.
- `addReview` - adiciona uma *review* do utilizador, recorrendo ao método '`addReview`' da classe `Sessao`
- `removeReview` - remove uma *review* do utilizador, recorrendo ao método '`removeReview`' da classe `Sessao`
- `atualizaReports` - recorre ao método '`atualizaReports`' da classe `Sessao Moderador` para atualizar a lista de '`ReportedReview`' do moderador.
- `loqUser` - recorrendo ao método '`loqUser`' da classe moderador bloqueia o utilizador e remove as suas reviews.
- `modRemoveReview` - recorrendo ao método '`modRemoveReview`' da classe moderador, remove a *review* reportada especificada do sistema
- `resetarReports` - recorrendo ao método '`resetarReports`' da classe moderador, tira os *reports* da *review* especificada.
- `addModerador` - utiliza o método '`addModerador`' da classe `Administrador` para criar o novo moderador na base de dados.
- `removeModerador` - utiliza o método '`removeModerador`' da classe `Administrador` para remover o utilizador na base de dados.



## Classe BDHandler

### Descrição:

Esta classe é a responsável pela interação com a base de dados.

### Atributos:

Não possui atributos.

### Métodos:

- addUtilizador - cria um utilizador na base de dados
- nivelPermissao - verifica na base de dados, qual o nível de permissão atribuído ao utilizado (utilizador, moderador ou administrador)
- verificaBloqueado - acede à base de dados e retorna o estado de bloqueio do utilizador
- bloqUser - atualiza na base de dados, o atributo Bloqueado para o valor 1, relativo ao utilizador fornecido, e elimina as *reviews* relativas a este
- verificaLogIn - procura uma entrada de utilizador na base de dados cujo email e a palavra-passe correspondam aos fornecidos. Em caso de sucesso, retorna o id de utilizador. Caso contrário, retorna '*null*'
- recuperaPassword - devolve *password* associado ao email fornecido, ou '*null*' caso não exista uma conta com o email fornecido.
- removeUtilizador - remove um utilizador da base de dados, retorna falso se o utilizador não existir.
- alteraGosto - adiciona uma entrada de "gosto" na base de dados. Se já existir, elimina-a.
- verificaGosto - verifica a existência de uma entrada de "gosto" associada ao utilizador e review fornecidos
- alteraReport - cria uma entrada de "report" na base de dados. Caso exista, elimina-a.
- verificaReport - verifica a existência de uma entrada de *report* associada ao utilizador e review fornecidos
- getUtilizadores - pede à base de dados, uma lista de utilizadores que possuam algum dado que se aproxime da *String* fornecida
- atualizaUtilizador - solicita à base de dados, a substituição dos dados do Utilizador pelos dados providenciados. Informando do insucesso caso o id do utilizador não exista ou caso já haja um email igual ao fornecido.

- `getAutorReview` - requisita o ID do autor da *review*, à base de dados
- `addReview` - cria uma *review* na base de dados.
- `alteraReview` - altera o comentário e a classificação de uma *review* existente na base de dados
- `alteraPlguardado` - adiciona uma nova entrada de Ponto de Interesse, se o *ID* do mesmo ainda não existir, caso exista é removido.
- `removeUserReviews` - remove *reviews* de um utilizador da base de dados, retornando falso caso o utilizador não exista.
- `removeReports` - remove todos os *reports* associados a uma *review* da base de dados, retornando falso caso a *review* não exista.
- `removeReview` - remove uma *review* da base de dados, retornando falso caso a *review* não exista.
- `loadUtilizador` - devolve um Utilizador da base de dados, caso o utilizador não exista *null* e devolvido.
- `retrieveReports` - devolve todas as *review* reportadas presentes na base de dados.
- `getPIReviews` - devolve todas as *review* ,presentes na base de dados, associadas ao ponto de interesse especificado.
- `loadReview` - devolve a *review* especificada, caso a *review* não exista e devolvido *null*.
- `verificaPlguardado` - pergunta à base de dados se o utilizador está associado ao ponto de interesse fornecido
- `guardarPI` - requisita a criação de uma entrada para um "Ponto de interesse guardado", e associa-a ao utilizador indicado
- `getReview` - retorna a *review* com o identificador fornecido, utilizando a base de dados
- `getUtilizador` - retorna o utilizador com o identificador fornecido
- `getPIReviews` - devolve os identificadores de todas as *reviews* associadas ao ponto de interesse
- `getClassificacaoReview` - devolve a classificação de uma *review* a partir do seu ID.

## Classe **APIHandler**

### Descrição:

Esta classe é caracterizada pelos métodos usados para contactar as APIs externas à aplicação.

**Atributos:**

Não possui atributos.

**Métodos:**

- `localizaUser` - descobre a localização do utilizador, e retorna-a em formato de string
- `localizaPIsPorLocal` - cria uma lista de Pontos de Interesse referentes à localização fornecido com os filtros das Tags e do raio de distancia.
- `localizaPIsPorNome` - usa o método '`localizaUser`' para procurar uma lista de Pontos de Interesse á volta do utilizador, com o raio de distancia do argumento fornecido, referentes ao nome introduzido e com os filtros das Tags.
- `getDist` - calcula a distancia entre um ponto de interesse e uma localização.
- `getTags` - retorna a lista de *tags* suportadas pela API.
- `loadPI` - retorna um ponto de interesse a partir do seu ID.

## EmailHandler

**Descrição:**

Esta classe possui os métodos necessários para enviar emails aos utilizadores.

**Atributos:**

Não possui atributos.

**Métodos:**

- `enviaPassword` - Envia a seguinte mensagem para o email fornecido: "Recebemos um pedido de recuperação de password para a tua conta de Near You. Se tu não fizeste um pedido de recuperação de password, por favor ignora este email, não serão feitas alterações à tua conta. Outra pessoa pode ter escrito o teu email por engano, mas no entanto, encorajamos-te a alterar a tua password.

A tua password é <password>. Bem vindo, sentimos a tua falta."

- `enviaNewAccNoti` - Envia a seguinte mensagem para o email fornecido: "Bem-vindo(a) <username>, Estamos felizes por teres criado uma conta Near You!

A aplicação Near You é capaz de listar os Pontos de Interesse próximos de si. Também

poderá fazer reviews destes e guarda-los no seu perfil. Será possível outras pessoas verem o teu perfil e vice versa, portanto não escrevas reviews inapropriadas, pois estas serão apagadas e poderás ter a tua conta bloqueada. Lamentamos informar mas ainda estamos em versão beta sendo que não temos muitas funcionalidades. Desfruta da nossa aplicação e, com o tempo lançaremos mais actualizações."

- `enviaRemovedReviewNoti` - Envia a seguinte mensagem para o email fornecido: "Lamentamos informar mas, uma das suas reviews foi apagada. A review está representada abaixo: <Review do utilizador>. A sua review pode ter sido apagada por motivos de spam, fazeres publicidade, expor conteúdo sexual, discurso de ódio ou violência explícita."
- `enviaBlockedAccNoti` - Envia a seguinte mensagem para o email fornecido: "Lamentamos informar mas tu foste bloqueado. Não poderás voltar a aceder a tua conta, sendo que os bloqueios são permanentes."

## Classe Sessao

**Descrição:** Esta classe encarrega-se de executar as ações básicas, comuns aos utilizadores, moderadores e administradores, da sessão actual, tendo como auxiliares principais as classes BD Handler e API Handler. É também incumbida de armazenar os dados necessários para o funcionamento de todos os métodos acessíveis por um usuário.

### Atributos:

- `UserID` - identificador do utilizador na base de dados
- `Localizacao` - *string* com as coordenadas (latitude e longitude) do utilizador

### Métodos:

- `refreshLocalizacao` - recorre ao método '`localizaUser`' da classe `APIHandler` para descobrir a localização do utilizador e atualizar a variável '`Localizacao`'
- `localizaPlsPorNome` - recorrendo ao método da classe '`localizaPlsNome`' da classe `APIHandler`, vai se obter uma lista de Pls relacionados ao nome fornecido e filtrados pelo raio de distancia e as Tags. Se o utilizador decidiu aplicar filtros de classificação, vai ser utilizado o método '`carregaReviews`' do `PontoDeInteresse` para se obter as listas de id das reviews e depois se utilizar o método '`calculaClassificacaoMedia`' para se poder aplicar o filtro.
- `localizaPlsPorLocalizacao` - este método será idêntico ao '`localizaPlsPorNome`' mas neste caso, vai utilizar o método '`localizaPlsLocal`' para aplicar os filtros e neste caso receber uma lista de Pls próximos da localização fornecida. O filtro de classificação será aplicado de igual forma. Se a string de localização fornecida for '`null`,' é usada a

localização atual do utilizador.

- *alteraGosto* - utiliza o método '*alteraGosto*' da classe *BDHandler*, para adicionar um "gosto" a uma *review*, caso este já exista, é removido.
- *verificaGosto* - utilizando a classe *BDHandler*, através do método '*verificaGosto*', verifica a existência de um "gosto" do utilizador para a *review* fornecida
- *alteraReport* - chama o método '*alteraReport*' da classe *BDHandler*, para criar um "report" de uma *review*, caso este já exista, é removido.
- *verificaReport* - evocando o método '*verificaReport*' da classe *BDHandler* verifica a existência de um *report*, efetuado pelo utilizador, à *review* fornecida
- *searchUser* - recorrendo à classe *BD Handler*, descobre através de uma string, os utilizadores que melhor se identificam com ela.
- *alteraDados* - através do método '*atualizaUtilizador*', pertencente à classe *BDHandler*, modifica os dados da conta de um utilizador
- *alteraReview* - recorre ao método '*alteraReview*' da classe *BDHandler*, para substituir o comentário e a classificação de uma *review*. Apenas pode alterar caso o utilizador seja o autor desta (método '*getAutorReview*', da classe '*BDHandler*')
- *removeConta* - utiliza o método '*removeUtilizador*', da classe *BDHandler* para eliminar a conta do utilizador.
- *verificaPlguardado* - invocando o método '*verificaPlguardado*' da classe *BDHandler*, descobre se o ponto de interesse foi guardado pelo utilizador
- *guardarPI* - guarda ponto de interesse na coleção do utilizador, recorrendo o método '*guardarPI*' da classe *BDHandler*
- *addReview* - adiciona uma *review* do utilizador, recorrendo ao método '*addReview*' da classe *BDHandler*
- *removeReview* - remove uma *review* do utilizador, recorrendo ao método '*removeReview*' da classe *BDHandler*

## Classe Utilizador

### Descrição:

Esta classe é a responsável por guardar em memória as informações de cada utilizador, retiradas da base de dados.

### Atributos:

- Username - nome do utilizador na aplicação.
- UsedID - identificador do utilizador na base de dados.
- Email - email do utilizador.
- PontosdeInteresseGuardados - lista com os objetos de classe PontosdeInteresse, que são referentes aos pontos de interesse que o utilizador guardou para ver mais tarde.
- Reviews - lista com os objetos de classe Review, referentes às reviews que foram realizadas pelo utilizador.

#### **Métodos:**

Não possui métodos.

### **Classe PontoDeInteresse**

**Descrição:** Esta classe é a responsável por guardar em memória as informações de Ponto de Interesse, retiradas da base de dados e da API externa. **Atributos:**

- idPI - identificador do Ponto de Interesse da base de dados e da API.
- Reviews - lista com os IDs das reviews, referentes ao PI.
- Nome - nome do Ponto de Interesse.
- Descrição - descrição do Ponto de Interesse.
- Tags - lista de strings que representarão as tags referentes ao PI
- PathFoto - url referente ao caminho para a foto do PI.
- Endereço - morada do PI.
- ClassificacaoMedia - classificação média das reviews do PI.

#### **Métodos:**

- carregaReviews - utiliza o método 'getPIReviews' para obter uma lista de identificadores das reviews referentes ao ponto de interesse.
- urlDirecoes - cria um link URL a partir do endereço do PI e da localização do utilizador.
- calculaClassificacaoMedia - utiliza o metodo 'getClassificacaoReview' para obter uma lista de classificações das reviews e depois faz a media destas e atualiza o atributo 'ClassificacaoMedia'.

## Classe *Review*

### Descrição:

Esta classe é a responsável por guardar em memória as informações de cada *review*, retiradas da base de dados.

### Atributos:

- UserID - identificador do utilizador que fez a *review* na base de dados.
- ReviewID - identificador da *review* na base de dados.
- ReviewEscrita - comentário na *review* sobre o ponto de interesse.
- Classificação - classificação da *review*.
- NrGostos - número de utilizadores que gostaram da *review*.

### Métodos:

- incrementaGosto - aumenta o NrGosto em 1 unidade.
- decrementaGosto - diminui o NrGosto em 1 unidade.

## Classe *ReportedReview*

**Descrição:** Esta classe é a responsável por guardar o id da *review* e o seu número de reportes, retiradas da base de dados.

### Atributos:

- ReviewID - identificador da *review* reportada.
- Reports - lista de *reviews* reportadas.

### Métodos:

Não possui métodos.

## Classe *SessaoModerador*

### Descrição:

Esta classe dá *extend* à classe *Sessão*, dando ao moderador as funcionalidades requeridas para moderar o sistema.

**Atributos:**

- Reports - lista de *review* reportadas.

**Métodos:**

- atualizaReports - recorre ao método 'retrieveReports' pertencente ao BDHandler, para preencher o 'Reports' com as 'ReportedReviews' presentes no sistema.
- bloqUser - recorrendo ao método 'bloqUser' pertencente à BDHandler, atualiza a base de dados removendo todas as *reviews* do utilizador, a bool do utilizador, relativa ao estado de bloqueio, e recorre ao método 'enviaNewAccNoti' pertencente à EmailHandler, para notificar o utilizador que a conta dele está bloqueada.
- modRemoveReview - remove a *review* reportada especificada do sistema
- resetarReports - recorre ao método 'removeReports' pertencente ao BDHandler, tirando os *reports* da *review* especificada.

**Classe SessaoAdministrador****Descrição:**

Esta classe dá *extend* à classe SessãoModerador, dando ao Administrador as funcionalidades requeridas para gerir os moderadores.

**Atributos:**

Não possui atributos. **Métodos:**

- addModerador - utiliza o método 'addUtilizador' da classe BDHandler para criar o novo moderador na base de dados. É o utilizado o método 'enviaNewAccNoti', da classe Email Handler, para enviar uma notificação da criação da nova conta. Devolvendo um estado de sucesso do método.
- removeModerador - utiliza o método 'removeUtilizador' da classe BD Handler para remover o utilizador na base de dados.



## 3.3 Modelos comportamentais

### 3.3.1 Diagrama de *Use Cases*

Use cases são utilizados para capturar de forma sistemática os requisitos funcionais de uma aplicação, descrevendo como se desenrolam as interações com o sistema. O diagrama de *Use Cases* representa os atores, entidades externas que interagem com o sistema, e irá associar visualmente um *Use Case* a cada um destes atores.

Estes modelos UML são essenciais para o planeamento de grandes projetos de *software*, dado que permitem a discussão entre clientes e equipa de desenvolvimento, a cerca da forma como o sistema deve interagir com os diferentes atores. No nosso caso, os *Use Cases* terão um grande impacto, sendo que serão o principal guia para o comportamento da aplicação, da equipa de implementação.

Para a implementação do guia NearYou foram considerados os seguintes atores:

- **Utilizador:** É o foco da aplicação, a maioria dos *Use Cases* são feitos com as ações deste ator em mente;
- **Moderador:** Trata-se de um utilizador com permissão para remover *reviews* e bloquear outros utilizadores;
- **Administrador:** É um ator com a permissão de executar tudo o que os atores acima podem, para além de ter o poder de criar e eliminar contas de moderador.

Com os atores definidos, resta-nos definir as suas interações com o sistema. Na figura abaixo encontra-se o Diagrama de *Use Cases* que retrata isto mesmo.

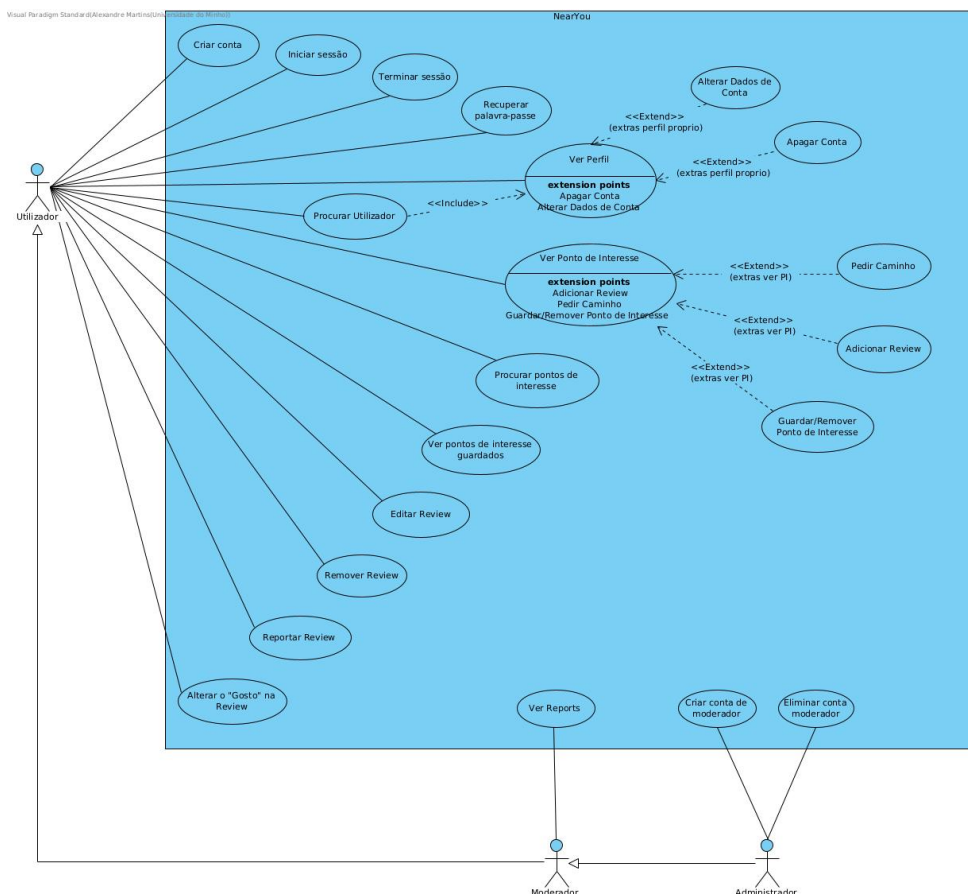


Figura 3.3: Diagrama de *Use Cases*

Na próxima secção entraremos em mais detalhes a cerca de cada Use Case em específico, onde serão detalhados o "Fluxo normal" (funcionamento esperado da aplicação), o "Fluxo Alternativo" (representa uma situação alternativa contemplada num funcionamento normal) e o "Fluxo de Exceção" (possíveis exceções que necessitem de tratamento próprio).

### 3.3.2 *Use Cases*

Nota: Encontram-se aqui os *Use Cases* sobre forma textual. Nos anexos do presente relatório, estão disponíveis os mesmos *Use Cases* apresentados sobre forma de tabela.

### 3.3.3 Use Cases do Utilizador

#### Criar Conta

De forma a que o Utilizador (não autenticado) possa usufruir da aplicação, este deverá efetuar a criação de uma conta. Na secção de criação de conta, deverá introduzir o nome de utilizador, *email*, palavra-passe e a confirmação da palavra-passe.

De seguida, o sistema valida as credenciais, verificando se não existe um utilizador com o mesmo *email* e se a palavra-passe coincide com a que se encontra no campo de confirmação.

Finalmente, o utilizador é redirecionado para a página inicial.

Existe ainda, um fluxo de exceção para o caso de as credenciais não cumprirem um dos requisitos: não existir um utilizador com o *email* introduzido, a palavra-passe coincidir com a confirmação e o número de caracteres não superar o limite imposto do campo. Neste fluxo, o utilizador é informado do problema com as credenciais introduzidas.

#### Iniciar Sessão

Para obter acesso às diversas funcionalidades da aplicação, o utilizador deve iniciar sessão, tendo como pré-condição não estar já autenticado.

Inicialmente, o utilizador introduz as suas credenciais (*email* e palavra-passe). Após serem confirmadas pelo sistema, é redirecionado para página inicial.

No caso de as credenciais não existirem no sistema, ou se o utilizador tiver sido bloqueado, trata-se de um caso de exceção em que as credenciais são recusadas, e o utilizador não é autenticado.

#### Terminar sessão

Caso o utilizador este autenticado, é-lhe permitido terminar a sessão, clicando no botão "Terminar sessão" que se encontra na página do próprio perfil.

#### Recuperar Palavra-Passe

O utilizador tem a opção de recuperar a palavra-passe da sua conta, selecionando a opção que se encontra na página de início de sessão.

O utilizador, começa por inserir o *email* da sua conta. O sistema verifica a existência da conta, e depois envia o *email* com a palavra-passe.

Existe um fluxo de exceção, para o caso do *email* introduzido não existir no sistema. O utilizador é informado da inexistência de uma conta com o *email* fornecido.

## **Ver Perfil**

O utilizador pode optar por ver o perfil de um utilizador.

Se seleccionar a opção de ver o próprio perfil, o sistema apresentará a página do perfil, em que é possível executar diversas ações, sendo as ações de alterar os dados da conta e apagar conta exclusivas desta página.

No caso de escolher a opção de ver o perfil de outro utilizador, o sistema apresentará a página do perfil desse utilizador, onde será possível observar os dados deste, assim como, as suas *reviews*.

## **Alterar Dados da Conta**

O utilizador, caso esteja autenticado, pode optar por mudar dados da sua conta. Estes dados podem ser o seu nome de utilizador, o seu *email* e a sua palavra-passe.

Para efetuar esta mudança, terá de seleccionar a opção de mudar dados que aparece no seu perfil. De seguida, o sistema apresentará a página onde o utilizador pode modificar os dados. Após alterar os campos pretendidos, o utilizador clica na opção para guardar. Por fim, o sistema atualiza os dados da conta.

Se já existir uma conta com o novo email introduzido ou os campos referentes à palavra-passe não coincidirem ou o número de caracteres dos campos superar o limite imposto, são casos que pertencem ao fluxo de exceção em que os dados introduzidos não cumprem os requisitos. O utilizador é informado da razão pela qual não foi feita a alteração.

## **Apagar Conta**

O utilizador, se tiver sessão iniciada, pode eliminar a sua conta, clicando na opção existente no próprio perfil.

É-lhe apresentada, pelo sistema, uma caixa de diálogo a pedir confirmação para eliminar a conta.

Após o utilizador confirmar a ação, o sistema remove todos os seus dados, excepto as suas *reviews*, cujo nome de utilizador passará a ser "Utilizador apagado". Por fim, redireciona o utilizador para a página de início de sessão.

## **Procurar utilizador**

A um utilizador autenticado, é dada a opção de procurar por outro utilizador, permitindo desta forma, ver as suas avaliações feitas a pontos de interesse.

Após seleccionar a opção de procurar utilizador e introduzir um dado identificador, o sistema efetua uma procura, e recolhe numa lista, os diversos perfis de utilizador que se aproximarem do dado fornecido. Depois de apresentar os resultados, o utilizador passa a escolher o perfil que pretender.

Não serem encontrados perfis de utilizador, que se aproximem do dado identificador introduzido, é um fluxo de exceção. Neste cenário, o sistema comunica o erro ao utilizador.

## **Ver Ponto de Interesse**

Quando um utilizador está a ver uma lista de pontos de interesse, este tem a possibilidade de clicar num dos Pontos de Interesse para o expandir e desbloquear mais opções.

## **Pedir caminho**

No menu expandido do Ponto de Interesse, o utilizador tem a opção de clicar no botão "Direções" que o vai redireccionar para um link gerado pelo sistema com as indicações/caminho para o seu destino.

## **Guardar/Remover Ponto de Interesse**

No menu expandido do Ponto de Interesse, o utilizador tem a opção de clicar no botão "Adicionar PI".

O sistema guarda o ponto de interesse se este ainda não tiver sido registado na coleção do utilizador, caso contrário, remove-o.

Também é possível remover directamente um *PI* se o utilizador estiver no seu perfil.

## **Procurar pontos de interesse**

No menu principal, é dado ao utilizador a opção de procurar pontos de interesse numa barra de pesquisa. O utilizador poderá seleccionar a opção de procurar pontos de interesse por nome, ou pode procurar pontos de interesse duma região, introduzindo, previamente, o nome da mesma ou seleccionando a opção que lhe permite utilizar a sua região atual.

Estes pontos de interesse serão gerados por uma API externa que deverá devolver uma lista dos mesmos consoante a procura feita. Esta usará um raio de distancia pré-definido de 2 km caso se tenha escolhido uma localização.

O utilizador terá a opção de aplicar filtros de procura (antes e depois da pesquisa), como o raio de distância (ao ponto da região), *tags* (como restaurante e museu) que serão fornecidas pela API e estarão guardadas em memória, ou classificação mínima (de 0 a 5). Os Pontos de Interesse que ainda não tenham *reviews*, terão classificação 0.

O utilizador terá a liberdade de organizar a lista de pontos de interesse por proximidade e classificação.

Caso a API não tenha encontrado nenhuma correspondência, deverá aparecer na lista de pontos a mensagem "Não existem Pontos de Interesse que correspondem à sua procura".

### **Ver Pontos De Interesse guardados**

No perfil de utilizador, o utilizador tem acesso a uma lista de pontos de interesse guardados que podem ser acedidos.

### **Adicionar *Review***

No menu expandido do Ponto de Interesse, o utilizador tem a opção de seleccionar o botão que o vai permitir escrever uma *review*. Este botão vai abrir um menu que possibilita a escrita de uma *review* e a atribuição de uma classificação. No fim de escrever e atribuir a classificação, o utilizador terá de seleccionar a opção "Publicar" para partilhar a sua *review*.

Caso o utilizador já tenha feito uma *review* sobre o Ponto de Interesse, aparecerá a mensagem "Você já possui uma *review* sobre este Ponto de Interesse (Máximo de uma *review*)".

### **Editar *Review***

O utilizador deverá poder alterar uma *review*, desde que esta seja da sua autoria. Após a seleção desta opção, o sistema apresenta uma caixa de edição da *review*. Após alterar o que pretende, o utilizador deve seleccionar a opção para guardar as alterações. De seguida, o sistema atualiza a entrada existente no sistema.

### **Remover *Review***

O utilizador tem a possibilidade de eliminar a própria conta, clicando na opção existente no próprio perfil. Quando seleccionada, o sistema pede a confirmação da ação, através de uma

caixa de diálogo. Depois da ação ser confirmada pelo utilizador, o registo da sua *review* é removido pelo sistema.

### **Reportar *Review***

Num perfil de outro utilizador, ou na página de um ponto de interesse, o utilizador tem a opção de reportar uma *review*.

O sistema adiciona o *report* da *review* se já não existir um associado ao utilizador. No caso de já existir, é removido.

### **Alterar o "Gosto" na *review***

Num perfil de outro utilizador ou num ponto de interesse, o utilizador tem a opção de clicar no botão de "Gosto".

O "Gosto" vai ser adicionado, pelo sistema, se não existir e removido caso já exista.

## **3.3.4 Use Cases do Moderador**

### **Ver *reports***

No menu exclusivo ao moderador, é-lhe apresentada uma lista de *reviews* reportadas, sendo possível ao moderador analisar uma das *reviews*.

Caso a *review* analisada seja considerada inofensiva, é marcada como um falso *report* dando o caso como resolvido.

Por outro lado, caso a *review* seja considerada ofensiva, o moderador irá proceder à análise do perfil do usuário que a publicou. Se considerar que existem mais *reviews* ofensivas no perfil do utilizador, o moderador seleciona a opção de bloquear a conta do utilizador, o que obriga o sistema a remover todas as suas informações do sistema. O sistema terá também de notificar o utilizador em questão via email.

Se porventura, o perfil não apresentar mais conteúdo ofensivo, a *review* reportada será removida e o usuário será notificado automaticamente por email do sucedido.

O sistema, independentemente do caso, remove todos os *reports* referentes à *review* analisada.

### **3.3.5 Use Cases do Administrador**

#### **Criar conta de moderador**

O Administrador tendo sessão iniciada, será capaz de criar uma nova conta de moderador, sendo esta, uma ação exclusiva do seu cargo.

Para gerar um novo moderador é necessário introduzir as credenciais desejadas para a nova conta. Caso sejam inválidas, é-lhe informado que os dados introduzidos são inválidos, e repetindo o processo de pedido de credenciais.

Após a conta ser registada pelo sistema, é enviado um email de confirmação para o email associado à nova conta.

#### **Remover conta de moderador**

O Administrador tendo sessão iniciada será capaz de remover uma nova conta de moderador, sendo que esta ação é exclusiva ao administrador.

O administrador após seleccionar a opção para remover um moderador, é-lhe apresentado um menu com uma listagem das contas de moderadores existentes, e com uma opção para procurar um moderador em específico através de um identificador do mesmo.

Quando o administrador seleccionar a opção para remover o moderador, ser-lhe-á apresentada uma caixa de diálogo, de forma a confirmar a ação. Mal o administrador confirme a remoção, a conta do moderador será eliminada do sistema.

### **3.3.6 Diagrama de Atividades**

Os diagramas de Atividades à semelhança dos Use Cases, são diagramas que retratam uma interação com um ator, mas diferenciam-se graças a uma ênfase na forma de "como" o sistema reage a interações. Apresentam-se assim como excelentes complementos aos Use Cases, permitindo através de uma forte componente visual, demonstrar o fluxo do sistema.

#### **Adicionar uma review**

Para a atividade "Adicionar uma review", o utilizador começa por tentar iniciar sessão. A partir daqui, se o iniciar sessão concluir com sucesso, o sistema verifica o nível de permissões da conta, cria a sessão correspondente à permissão, e prossegue para pedir às API externas a localização e os pontos de interesse perto desta. O sistema necessita agora de agregar as informações retiradas das API e criar os Pontos de Interesse que serão depois apresentados ao



utilizador. Quando o utilizador escolher um Ponto de Interesse, é lhe apresentada uma página com mais informação, onde está disponível uma opção para adicionar uma review. O utilizador seleciona-a e é lhe apresentado uma caixa de texto onde pode escrever a sua review. Quando acabar, e após atribuir as estrelas, seleciona a opção para concluir a review, e resta ao sistema, verificar se o utilizador não foi bloqueado enquanto escrevia a sua review, e se tal não tiver acontecido, a review é adicionada ao sistema.

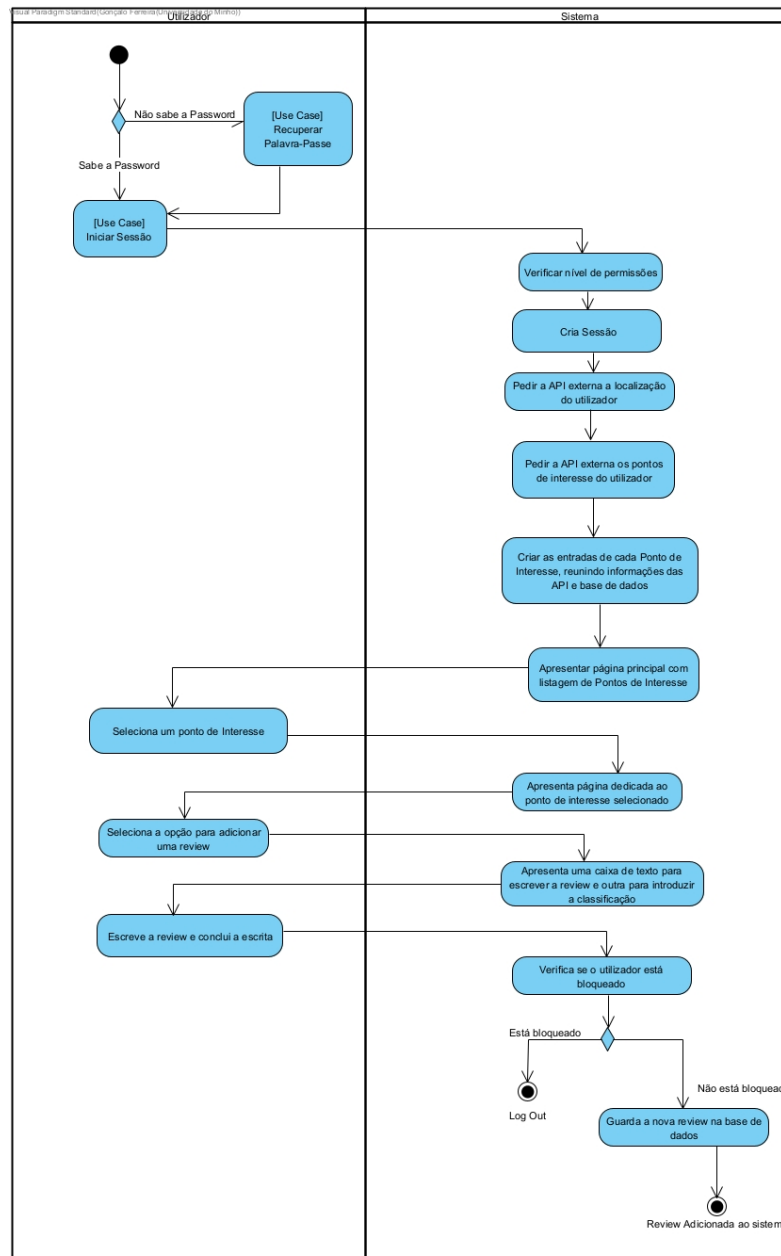


Figura 3.4: Atividade : Adicionar uma review

## Avaliação de Lista de Reports

De forma a concluir a atividade "Avaliação da lista de reports" o moderador deve primeiro estar autenticado na página inicial da aplicação NearYou. Para ver as reviews com reports, o moderador deve selecionar a opção para ver esta lista, e mal a seleciona, o sistema deverá aceder a base de dados, recolher todas as reviews que têm um número de reports superior a 10 e por fim apresentar visualmente a listagem. Resta agora ao moderador, analisar as reviews e escolher uma ação disponível:

- **Marcar como falso report** : se a review for cumpridora dos termos de serviço, o moderador apenas seleciona a opção para limpar todos os reports daquela review

- **Ir ao perfil do utilizador**: se considerar a review inapropriada, seleciona a opção de visualizar o perfil do utilizador, que será carregado da base de dados e apresentado ao moderador

- **Bloquear o utilizador**: se considerar que existem mais reviews que quebram os termos de serviço, o moderador seleciona a opção de bloquear utilizador, que o sistema interpreta como a necessidade de sinalizar este utilizador na base de dados como bloqueado e remover todas as reviews em seu nome da BD. O utilizador deverá ser notificado por email do seu estado de bloqueado.

- **Não bloquear o utilizador**: se considerar que apenas aquela review infringe os termos de serviço, o moderador seleciona a opção de eliminar esta review. O sistema deve assim eliminar apenas esta review do sistema e enviar uma notificação por email ao utilizador notificando-o da remoção da review.

Nas ações que alteram os dados do sistema, como a remoção da review, apresentadas em cima, é necessário verificar que o moderador não foi removido pelo administrador, para evitar que moderadores suspensos das suas ações alterem os dados da aplicação. Se um moderador bloqueado por um administrador tentar realizar qualquer uma das operações acima mencionadas, a sessão deverá ser terminada.

Por fim, se tudo correr como esperado, o moderador será reencaminhado para a página com a lista de reports, para acabar de analisar as restantes reviews.

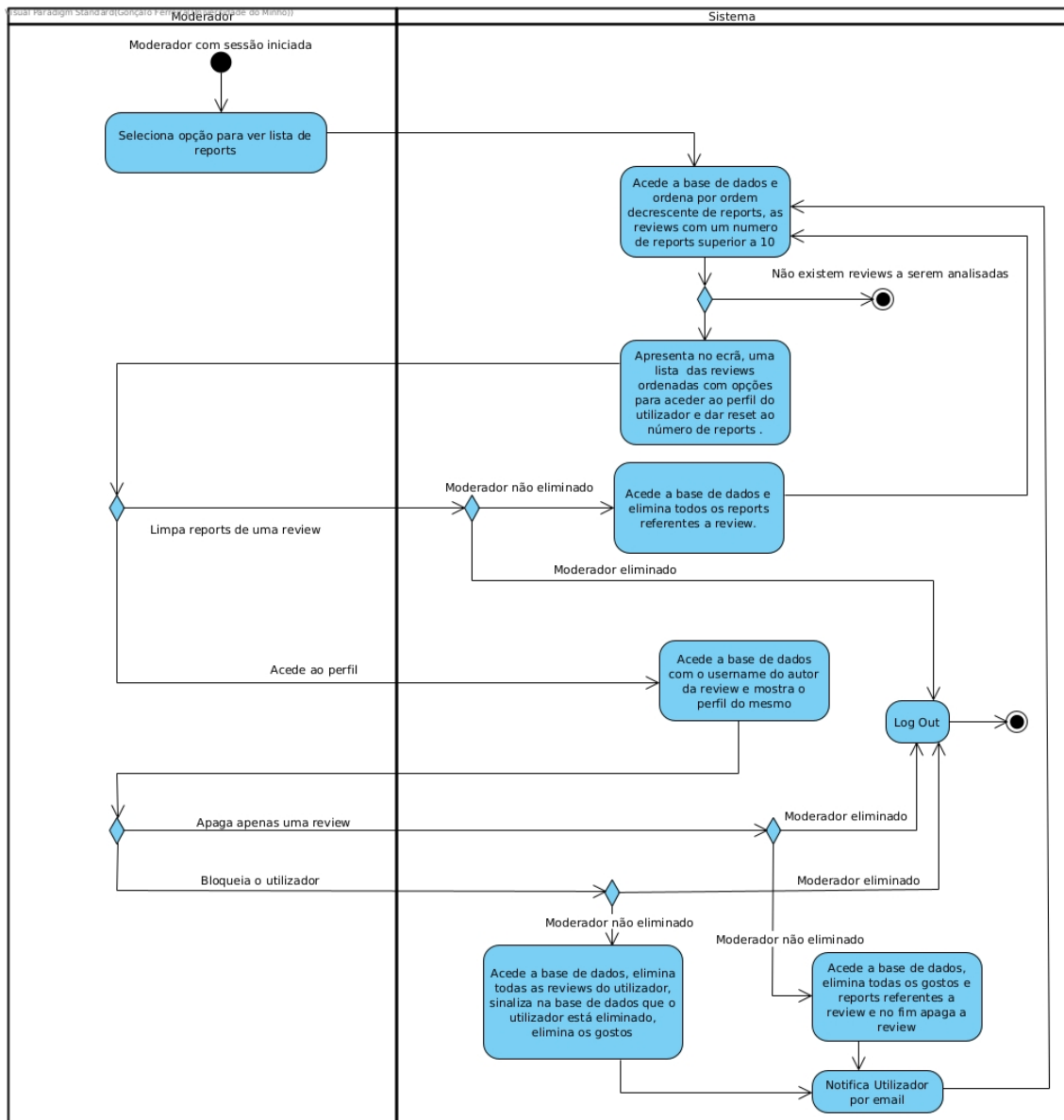


Figura 3.5: Atividade : Avaliação de Lista de Reports

## 4 Conceção do Sistema de Dados

### 4.1 Apresentação geral da estrutura (esquema) do sistema de dados

Para este projeto é essencial a utilização de um sistema de dados, que permita guardar de forma organizada, fiável e de rápido acesso, uma grande quantidade de dados relativa aos utilizadores e reviews. De forma a representar a estrutura e as entidades que pretendemos que sejam guardadas na BD, foi desenhado o seguinte diagrama de Entidade Relacionamento na figura 4.1.

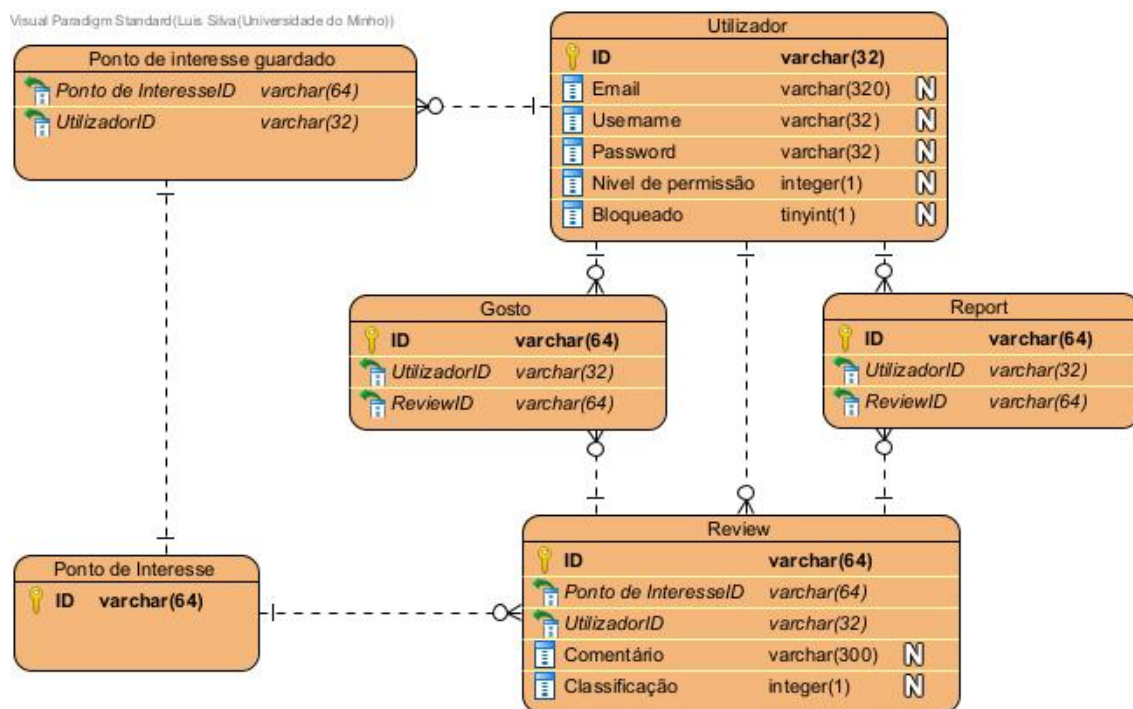


Figura 4.1: Diagrama de Entidade Relacionamento

Na próxima secção iremos explicar as decisões tomadas a cerca desta base de dados.

## 4.2 Descrição detalhada dos vários elementos de dados e seus relacionamentos

Como foi retratado no diagrama, podemos dividir a nossa base de dados em 6 entidades:

### 4.2.1 Utilizador

Nesta entidade estarem presentes os seguintes atributos:

- A chave primária, **ID**, que permite a distinção entre utilizadores na base de dados
- **Email**
- **Username**
- **Password**
- **Nível de permissão** que ditará as funcionalidades disponíveis para o utilizador. Quando a conta é criada, a aplicação deve definir automaticamente este valor, dependendo da forma como a entrada foi criada (se foi por um utilizador comum ou por um administrador). A conta administrador deverá ser uma conta predefinida no sistema de dados
- **Bloqueado** que se apresenta na forma de um *tiny int* que funcionará como um booleano, que indica se este utilizador está bloqueado ou não. Deverá ser inicializado a 0.

### 4.2.2 Review

Nesta entidade estarem presentes os seguintes atributos:

- A chave primária, **ID**, que permite a distinção das reviews na base de dados
- A chave estrangeira, **Ponto de InteresseID**, referente ao identificador do ponto de interesse, ao qual a *review* pertence
- A chave estrangeira, **UtilizadorID**, que se refere ao identificador do utilizador que realizou a *review*
- **Comentário**, até 300 caracteres
- **Classificação**, um inteiro de 1 a 5

### 4.2.3 Gosto e *Report*

Estas entidades são bastante similares, e o seu objetivo é registarem as pessoas que deram gosto ou *report* numa *review*, de forma a evitar gostos ou *reports* de utilizadores repetidos. Quanto aos seus atributos, possuem uma chave primária **ID**, e duas chaves estrangeiras **UtilizadorID** e **ReviewID**, de forma a indicarem quem e onde foi dado o *report*/gosto.

### 4.2.4 Ponto de Interesse

A entidade ponto de interesse foi criada para salvaguardar os identificadores dos pontos de interesse mencionados no sistema, quer por reviews, quer por pontos de interesse guardados. Este foi criado para ligar as informações relativas a um ponto de interesse na base de dados.

### 4.2.5 Ponto de Interesse guardado

Esta entidade relaciona duas chaves estrangeiras, **Ponto de InteresseID** e **UtilizadorID** que permitem associar a um utilizador um ponto de interesse que este decidiu guardar, sem existirem duplicados.

Como podemos ver nas entidades retratadas, os Pontos de Interesse, não serão guardados integralmente na base de dados, e existirá assim uma forte dependência da API, para obter estas informações. Com esta dependência, surge também a possibilidade de a nossa aplicação deixar de ser capaz de apresentar pontos de interesse, quando a API externa não estiver a funcionar ou for alterada. Apesar disto, o grupo acredita que as vantagens superam as desvantagens: o facto de não ser necessário guardar na nossa base de dados a informação de potencialmente todos os pontos de interesse do mundo, leva a que as nossas bases de dados sejam mais fáceis de gerir, evitando a necessidade de atualizar a base de dados sempre que é feita uma *query* à API.

## 5 Esboço dos Interfaces do Sistema

Depois da realização do levantamento e análise de requisitos, podemos dizer que temos conhecimento suficiente da aplicação para projetar uns esboços de como será a interface de comunicação do utilizador com o sistema.

Neste capítulo serão utilizadas as imagens desenvolvidas pelo grupo com o intuito acima descrito, acompanhadas de uma breve descrição acerca da forma de funcionamento dos pontos principais nela apresentados.

### 5.1 Estrutura geral das interfaces do sistema

A primeira interface com que o utilizador interage é a página responsável pela autenticação, onde se não tiver conta, poderá criá-la na interface do registo do utilizador.

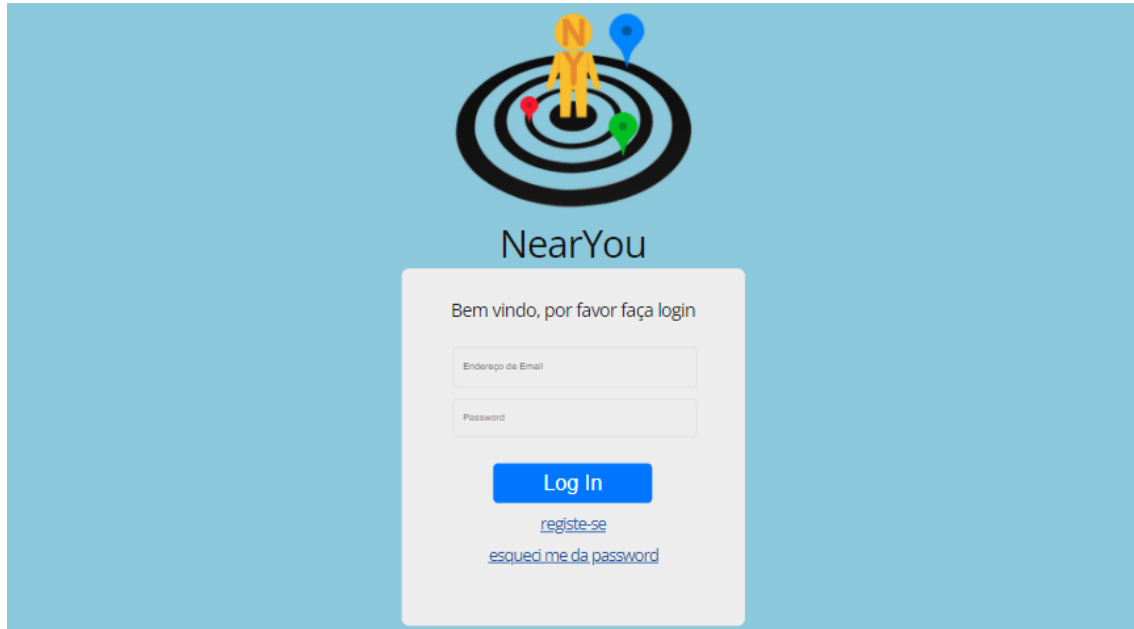



Figura 5.1: Interface de Login



NearYou

Registe-se

Endereço de Email

Nome do utilizador


Palavra-passe

Confirmar palavra-passe

Sign In

Figura 5.2: Interface de Signin

Se o utilizador se tiver esquecido da senha, terá a opção de recuperá-la na interface de autenticação.



NearYou

A nova password será enviada para o email que inserir

Endereço de Email

Recuperar

Figura 5.3: Interface de Recuperação de Password



Quando o utilizador é autenticado com sucesso, é direcionado para a interface principal da aplicação "NearYou", cujo estado *default* é a apresentar a lista com os pontos de interesse que estão próximos da localização geral do utilizador.

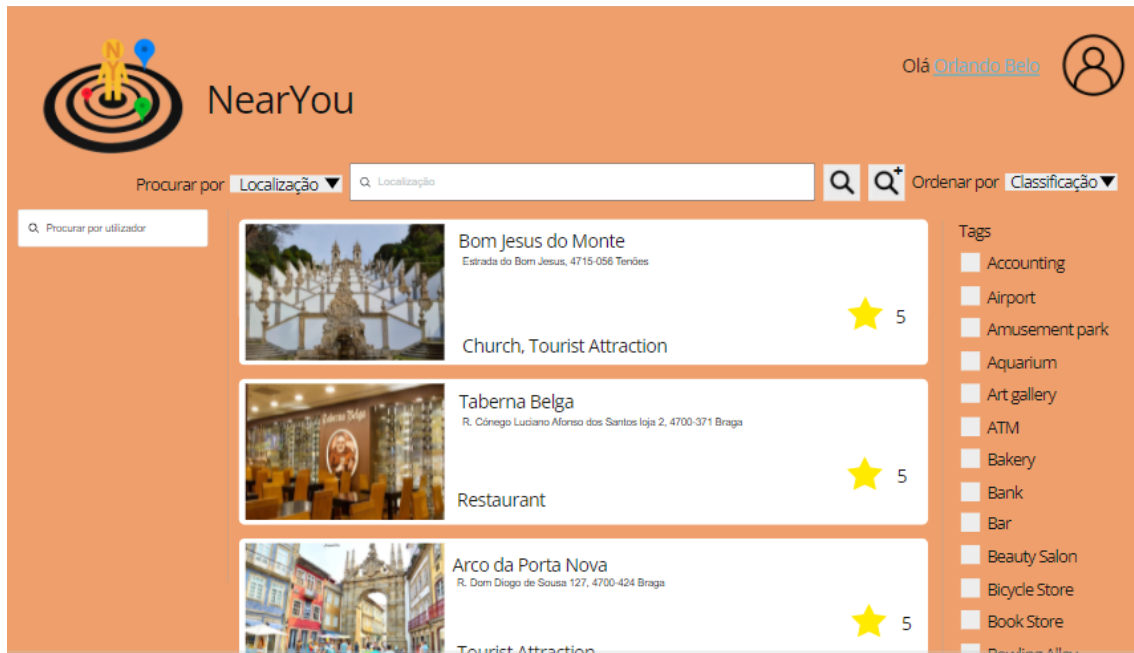


Figura 5.4: Interface Principal para o Utilizador Comum

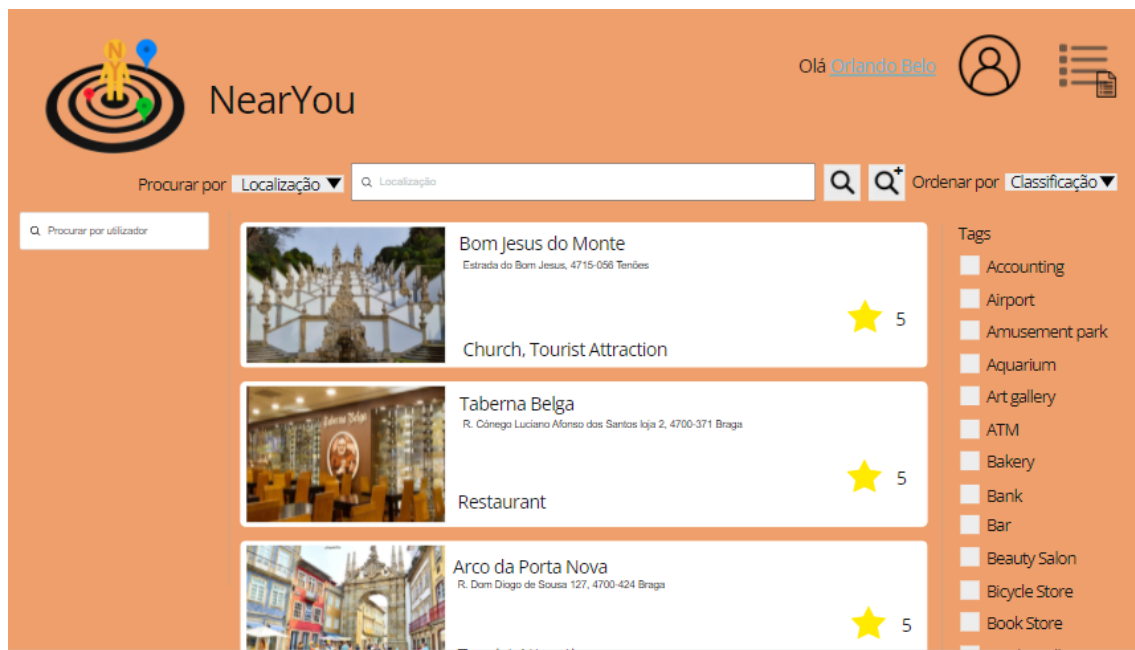


Figura 5.5: Interface Principal para os Moderadores

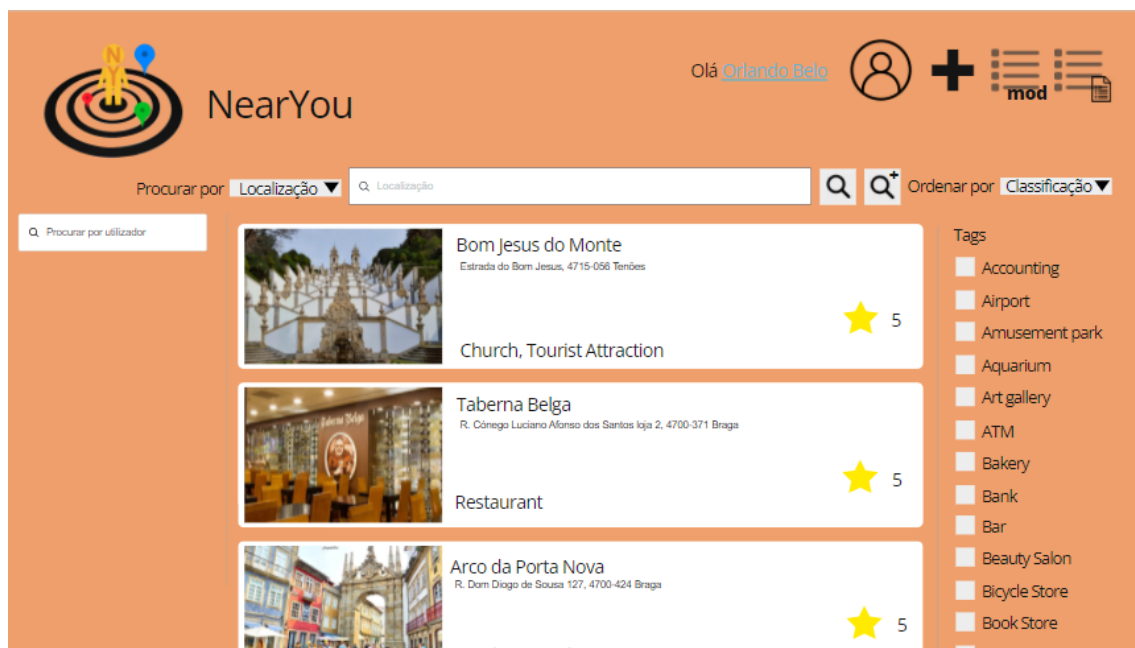


Figura 5.6: Interface Principal para os Administradores

Se o utilizador carregar num dos pontos de interesses listados é encaminhado para a interface desse ponto de interesse.



Figura 5.7: Interface do Ponto de Interesse

Na interface de um ponto de interesse, o utilizador pode escolher deixar uma review. Se for o caso, o utilizador é direcionado para a interface da criação da review.

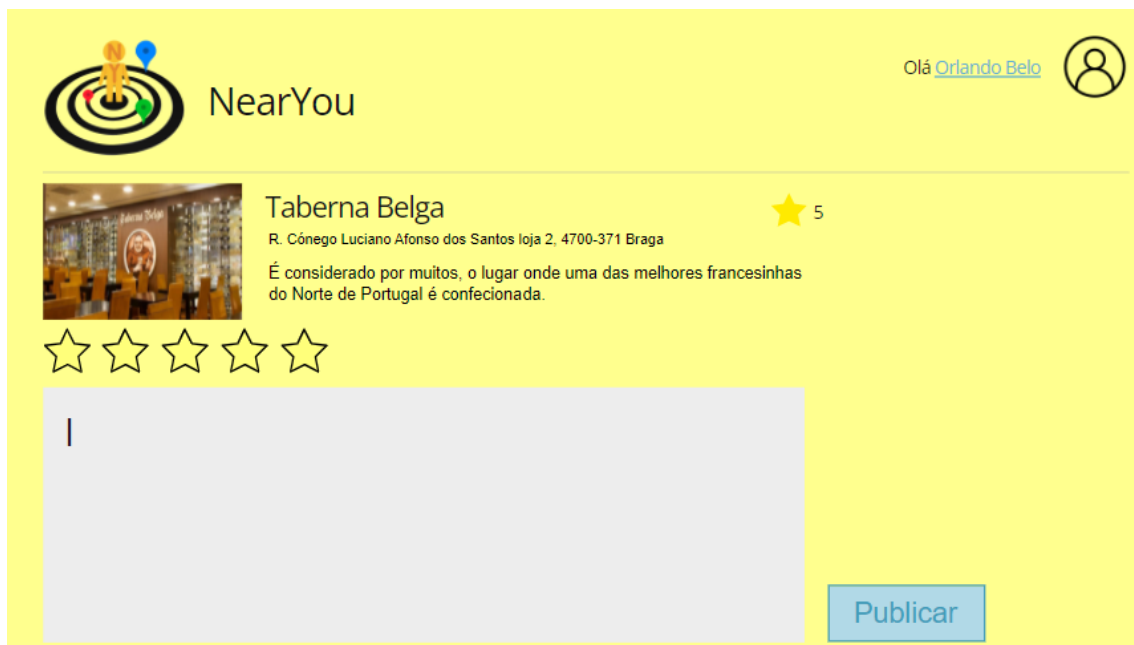



Figura 5.8: Interface de Criação de Review


Ainda autenticado, o utilizador, tem acesso ao seu perfil ao carregar no ícone de perfil no canto superior direito, presente em todas as interfaces após estar autenticado.



Figura 5.9: Interface Perfil

No perfil do utilizador há opção para mudar os dados da conta, o que direciona o utilizador para outra interface.



Olá [Orlando Belo](#) 

---

Email Address    constantino@\*\*\*.pt

Escreva aqui o novo email

Username        Orlando Belo

Escreva aqui o novo useranme

Password

Escreva aqui a password nova

Alterar




Figura 5.10: Interface de Alteração da Conta

A partir da interface principal, o moderador e o *admin* conseguem ver e avaliar os *reports* feitos pelos os utilizadores comuns.



Olá [Orlando Belo](#)  

---

Reportes por analisar    3

<p>Taberna Belga <small>R. Cónego Luciano Afonso dos Santos loja 2, 4700-371 Braga</small></p> <p>Mais valia comer m****!</p>	 1  3	  
<p>Bom Jesus do Monte <small>Estrada do Bom Jesus, 4715-056 Terões</small></p> <p>Prefiro olhar para o ralo da sanita</p>	 1  1	  
<p>Taberna Belga <small>R. Cónego Luciano Afonso dos Santos loja 2, 4700-371 Braga</small></p> <p>Segunda melhor francesinha que comi!!</p>	 5  3	  




Figura 5.11: Interface de Gerência de Reportes

Através da interface anterior, o moderador pode visitar o perfil do autor de uma *review*.




Figura 5.12: Interface de Análise de Perfil

Na interface principal, o *admin* ainda tem a opção de criar moderadores e de os gerir, o que o direciona para as seguintes interfaces, respetivamente.



NearYou

Olá [Orlando Belo](#)



Insira o email address do novo moderador

Insira o username do novo moderador

Password

Criar






Figura 5.13: Interface de Criação de Moderador




NearYou


Olá [Orlando Belo](#)



+




mod



Moderadores 2

**José Barros**

josebarros@\*\*\*\*\*.pt



**Antonio Nestor**

antonionestor@\*\*\*\*\*.pt




Figura 5.14: Interface de Gerência de Moderador

## 5.2 Caracterização das interfaces

Os esboços acima apresentados mostram o aspeto base das interfaces, embora as cores sejam subjetivas.

A interface de autenticação deve ter:

- logótipo NearYou, no centro em cima dos itens seguintes;
- o campo de preenchimento do nome de utilizador, no centro por cima dos itens seguintes;
- o campo de preenchimento da palavra-passe, no centro por cima dos itens seguintes;
- um botão para confirmar a autentificação, no centro por cima dos itens seguintes;
- um botão para direccionar o utilizador para a interface de registo, no centro por cima do item seguinte;
- um botão para direccionar o utilizador para a interface de recuperação da palavra-passe, no centro;

A interface de registo deve ter:

- logótipo NearYou, no centro por cima dos itens seguintes;
- o campo de preenchimento do email, no centro por cima dos itens seguintes;
- o campo de preenchimento do nome de utilizador, no centro por cima do item seguinte;
- o campo de preenchimento da palavra-passe, no centro por cima do item seguinte;
- o campo de preenchimento da confirmação da palavra-passe, no centro por cima do item seguinte;
- um botão para confirmar o registo, no centro.

A interface de recuperação da palavra-passe deve ter:

- logótipo NearYou, no centro por cima dos itens seguintes;
- o campo de preenchimento do email, no centro por cima do item seguinte;
- um botão para confirmar o registo, no centro.



A interface principal deve ter:

- logótipo NearYou, no canto superior esquerdo, que quando pressionado deve direcionar o utilizador para a interface principal no estado *default*;
- botão de perfil do utilizador, que se for moderador ou *admin*, deve ser acompanhado pelo botão da gerência de reportes, e se for só *admin*, deve ainda ser acompanhado pelos botões de criação de moderador e gerência de moderadores;
- um campo de opções de procura, um campo de preenchimento de procura, um botão para confirmar a procura, um botão de procura com filtros e um campo de opções de ordenação dos pontos de interesse, seguidos horizontalmente, logo após o logótipo, sendo o campo de preenchimento, o mais centrado possível;
- a faixa à direita deverá conter todas as *tags* dos pontos de interesse com *checkboxes* pela sua esquerda;
- a listagem dos pontos de interesse deverá ser no centro, por baixo do campo de preenchimento de procura ;
- a faixa à esquerda deverá ter um campo de preenchimento de procura de utilizadores, onde aparece a listagem de utilizadores que podem corresponder ao que já está escrito abaixo dele;

A interface de criação de moderadores deve ter:

- logótipo NearYou, no canto superior esquerdo, que quando pressionado deve direcionar o utilizador para a interface principal no estado *default*;
- botão de perfil do utilizador, que se for moderador ou *admin*, deve ser acompanhado pelo botão da gerência de reportes, e se for só *admin*, deve ainda ser acompanhado pelos botões de criação de moderador e gerência de moderadores;
- campo de preenchimento do endereço do email do moderador, logo depois do logótipo, à esquerda;
- campo de preenchimento do nome do moderador, logo depois do campo de preenchimento do endereço do email do moderador, à esquerda;
- campo de preenchimento da palavra-passe do moderador, logo depois do campo de preenchimento do nome do moderador, à esquerda;
- botão para confirmar a criação do moderador, logo depois do campo de preenchimento da palavra-passe do moderador, à esquerda;
- botão *logout*, no canto inferior direito.

A interface de gerência de moderadores deve ter:

- logótipo NearYou, no canto superior esquerdo, que quando pressionado deve direcionar o utilizador para a interface principal no estado *default*;
- botão de perfil do utilizador, que se for moderador ou *admin*, deve ser acompanhado pelo botão da gerência de reportes, e se for só *admin*, deve ainda ser acompanhado pelos botões de criação de moderador e gerência de moderadores;
- o número de moderadores e a sua listagem, seguidos verticalmente, logo depois do logótipo e à esquerda;
- cada moderador deve ser apresentado numa caixa branca, com o nome de utilizador por cima do endereço de email do moderador;
- botão para eliminar moderador à direita de cada caixa de moderador.
- botão *logout*, no canto inferior direito.

A interface de gerência de reportes deve ter:

- logótipo NearYou, no canto superior esquerdo, que quando pressionado deve direcionar o utilizador para a interface principal no estado *default*;
- botão de perfil do utilizador, que se for moderador ou *admin*, deve ser acompanhado pelo botão da gerência de reportes, e se for só *admin*, deve ainda ser acompanhado pelos botões de criação de moderador e gerência de moderadores;
- o número total de *reviews* reportadas e a sua listagem, seguidos verticalmente, sendo o formato das *reviews* o mesmo do que na interface perfil, logo depois do logótipo.
- botão (->) para ver página do autor da review à direita de cada; caixa da review;
- botão (caixote de lixo) para eliminar review à direita de cada caixa da review e em baixo do botão (->);
- botão (X) para eliminar report logo à direita do botão (caixote do lixo):
- botão *logout*, no canto inferior direito.

A interface de análise do perfil de utilizador deve ter:

- logótipo NearYou, no canto superior esquerdo, que quando pressionado deve direcionar o utilizador para a interface principal no estado *default*;

- botão de perfil do utilizador, que se for moderador ou *admin*, deve ser acompanhado pelo botão da gerência de reportes, e se for só *admin*, deve ainda ser acompanhado pelos botões de criação de moderador e gerência de moderadores;
- o nome e o endereço de email do utilizador a analisar, seguidos verticalmente e logo depois do logótipo, à esquerda.
- o número total de *reviews* do utilizador seguido verticalmente da listagem vertical delas, logo depois do endereço de email e à esquerda;
- botão de bloqueio do utilizador, no mesmo nível que o nome de utilizador e à direita;
- botão *logout*, no canto inferior direito.

A interface de perfil deve ter:

- logótipo NearYou, no canto superior esquerdo, que quando pressionado deve direccionar o utilizador para a interface principal no estado *default*;
- botão de perfil do utilizador, que se for moderador ou *admin*, deve ser acompanhado pelo botão da gerência de reportes, e se for só *admin*, deve ainda ser acompanhado pelos botões de criação de moderador e gerência de moderadores;
- a localização atual do utilizador, logo depois do logótipo e à esquerda;
- botão de alteração de conta, logo depois da localização e à esquerda;
- botão de eliminação de conta, ao mesmo nível do que o botão de alteração de conta e à direita imediata do centro;
- o número total de *reviews* do utilizador seguido verticalmente da listagem vertical delas, logo depois do botão de alteração de conta e à esquerda;
- cada review deve ser uma caixa branca, com o nome, morada e comentário da review seguidos verticalmente, à esquerda na caixa, e a classificação e o número de gostos, seguidos verticalmente, à direita na caixa;
- botões para editar e apagar, junto e à direita de cada *review* e seguidos verticalmente;
- um campo de procura de pontos de interesse guardados, o seu número total e uma listagem deles, que se deve alterar à medida que se escreve no campo de procura, seguidos verticalmente, logo por baixo dos botões no campo superior direito.
- botão *logout*, no canto inferior direito.

A interface de alteração de dados da conta deve ter:

- logótipo NearYou, no canto superior esquerdo, que quando pressionado deve direcionar o utilizador para a interface principal no estado *default*;
- botão de perfil do utilizador, que se for moderador ou *admin*, deve ser acompanhado pelo botão da gerência de reportes, e se for só *admin*, deve ainda ser acompanhado pelos botões de criação de moderador e gerência de moderadores;
- campo de preenchimento do novo email com o email atual por cima, logo depois do logotipo e à esquerda;
- campo de preenchimento do novo nome de utilizador com o nome de utilizador atual por cima, logo depois do campo de preenchimento do novo nome de utilizador e à esquerda;
- campo de preenchimento da nova palavra-passe, logo depois do campo de preenchimento do novo email e à esquerda;
- botão de confirmação das alterações, logo depois do campo de preenchimento da nova palavra-passe e à esquerda;
- botão *logout*, no canto inferior direito.

A interface de ponto de interesse deve ter:

- logótipo NearYou, no canto superior esquerdo, que quando pressionado deve direcionar o utilizador para a interface principal no estado *default*;
- botão de perfil do utilizador, que se for moderador ou *admin*, deve ser acompanhado pelo botão da gerência de reportes, e se for só *admin*, deve ainda ser acompanhado pelos botões de criação de moderador e gerência de moderadores;
- as informações do ponto de interesse por baixo do logótipo, começando pela imagem à esquerda, seguido do nome, morada e descrição verticalmente e a classificação média do ponto de interesse no centro ao mesmo nível que o nome;
- uma lista das *reviews* e o seu número total, logo depois da informação do ponto de interesse, à esquerda;
- um botão para dar *report* à direita de cada *review*;
- botão para gostar em cada *review* e o seu número de gostos;
- botão para adicionar *review*;
- botão para guardar ponto de interesse ou eliminar se já estiver guardado, ao mesmo

nível do nome do ponto de interesse e à direita.

- botão para direcionar o utilizador para a API guia em que o destino é ponto de interesse, logo abaixo do botão para guardar ou eliminar.

A interface para adicionar uma *review* deve ter:

- logótipo NearYou, no canto superior esquerdo, que quando pressionado deve direcionar o utilizador para a interface principal no estado *default*;
- botão de perfil do utilizador, que se for moderador ou *admin*, deve ser acompanhado pelo botão da gerência de reportes, e se for só *admin*, deve ainda ser acompanhado pelos botões de criação de moderador e gerência de moderadores;
- as informações do ponto de interesse por baixo do logotipo, começando pela imagem à esquerda, seguido do nome, morada e descrição verticalmente e a classificação média do ponto de interesse no centro ao mesmo nível que o nome;
- 5 estrelas "vazias", seguidas horizontalmente, logo por baixo das informações do ponto de interesse e à esquerda, e se o utilizador pairar sobre uma delas com o rato, essa estrela e as que estão para trás ficam preenchidas durante essa posição do rato, e se clicar numa delas, essa aparência é permanente ou até o utilizador clicar noutra estrela;
- campo de preenchimento do comentário da *review*, que deve começar logo por baixo das estrelas até o canto inferior esquerdo e ao centro;
- botão para publicar a review, que deve estar junto ao canto inferior direito do campo de preenchimento.

## 6 Conclusões e Trabalho Futuro

Ao longo da primeira fase do projeto, o grupo de planeamento desenvolveu a especificação da futura implementação da aplicação "NearYou": um guia que permitirá encontrar todos os pontos de interesse próximos de uma localização, juntamente com as suas *reviews* e classificações de outros utilizadores. Esta fase é talvez das mais importantes para toda a realização do projeto, dado que foram aqui definidas as bases para a toda a implementação da aplicação. De forma a garantir a correta implementação de todas as funcionalidades, foi exigido ao grupo um plano detalhado, concreto e completo acerca do trabalho a realizar. Para completar esta tarefa o grupo elaborou uma lista de requisitos da aplicação e a partir destes serviu-se de diagramas UML para melhor idealizar e visualizar a estrutura da futura aplicação.

Após todo o trabalho desenvolvido, inúmeras vezes reescrito e retificado, o grupo conclui estar perante uma especificação completa e fiel à ideia inicial do projeto. Várias ideias novas foram introduzidas, como o aspeto da moderação das *reviews*, mas globalmente concluímos que estas seriam de possível realização no período de tempo estabelecido. Ainda assim, entendemos que poderíamos ter desenhado mais tipos de diagramas UML (como diagramas de sequência) para tornar a fase de implementação mais fácil e restritiva, de forma a obter todos os resultados desejados. Outro aspeto a ser melhorado, seria o planeamento da Conceção do Sistema de dados, que o grupo admite estar incompleto, muito devido, à inexperiência para com a utilização e representação das mesmas.

No futuro, este relatório terá de ser novamente analisado pela equipa de implementação, e será alvo de discussão, por ambas as partes, de forma a obter um melhor resultado final.

# Referências

- [1] Ian Sommerville, 2015, Software Engineering, 10th ed, <https://iansommerville.com/software-engineering-book/> [Accessed 22 November 2021]
- [2] Luís Miguel Queirós, 2021, "Covid-19 provocou quebra de 31% no sector cultural da União Europeia", *Público-Ípsilon*, <https://www.publico.pt/2021/01/26/culturaipsilon/noticia/covid19-provocou-quebra-31-sector-cultural-uniao-europeia-1947969> [Accessed 22 November 2021]
- [3] Inês Subtil (Jornalista), Joana Melo (Edição), 2021, "Estudo europeu. Cultura sofre com a pandemia", *RTP Notícias*, [https://www.rtp.pt/noticias/mundo/estudo-europeu-cultura-sofre-com-a-pandemia\\_v1300268](https://www.rtp.pt/noticias/mundo/estudo-europeu-cultura-sofre-com-a-pandemia_v1300268) [Accessed 22 November 2021]
- [4] Observador (autor não creditado), 2021, "Covid-19: OMS avisa para "impacto prolongado" da pandemia na saúde mental mundial", *Observador*, <https://observador.pt/2021/07/22/covid-19-oms-avisa-para-impacto-prolongado-da-pandemia-na-saude-mental-mundial/> [Accessed 22 November 2021]

# Lista de Siglas e Acrónimos

**BD** Base de Dados

**PI** Ponto de Interesse

**UML** Unified Modeling Language

**API** Application Programming Interface



# Anexos

## Use Cases de Utilizador

Use Case			
Use Case:	Criar conta		
Ator:	Utilizador		
Pré-condição:	Não estar autenticado		
Pós-condição:	Conta é registada e utilizador é autenticado		
	Passo	Ator	Sistema
Fluxo Normal	1.	Seleciona a opção de criar conta	
	2.		Apresenta página com requisitos para criar conta
	3.	Introduz as credenciais que pretende para a sua conta: nome de utilizador, email, palavra-passe e confirmação de palavra-passe.	
	4.		Valida e guarda as credenciais, e envia confirmação para o email fornecido.
	5.		Autentica o utilizador na aplicação.
	6.		Apresenta página inicial ao novo utilizador.
Fluxo Alternativo [credenciais não cumprem requisitos] (passo 4)	4.1.		Verifica que as credenciais não cumprem algum dos requisitos: email não existe, palavras-passe coincidem e número de caracteres não ultrapasse o limite imposto. Informa o utilizador do requisito não cumprido.

Figura 6.1: Criar Conta

Use Case			
Use Case:	Iniciar Sessão		
Ator:	Utilizador		
Pré-condição:	Não ter sessão iniciada		
Pós-condição:	Sessão iniciada		
	Passo	Ator	Sistema
Fluxo Normal	1.		Apresenta a página de início de sessão
	2.	Digita o seu email e a sua password	
	3.		Verifica que as credenciais existem
	4.		Verifica que a conta não está bloqueada
	5.		Autentica utilizador
	6.		Apresenta página inicial ao utilizador
Fluxo de exceção 1 [Credenciais não existem no sistema ou utilizador está bloqueado] (passo 3)	3.1		Recusa as credenciais e informa o utilizador da razão.

Figura 6.2: Iniciar Sessão

Use Case			
Use Case:	Terminar a sessão		
Ator:	Utilizador		
Pré-condição:	O utilizador tem sessão iniciada		
Pós-condição:	Sessão terminada		
	Passo	Ator	Sistema
Fluxo Normal	1.	Seleciona a opção de terminar sessão.	
	2.		Termina sessão do utilizador.
	3.		Apresenta página de início de sessão.

Figura 6.3: Terminar Sessão

Use Case			
Use Case:	Recuperar palavra-passe		
Ator:	Utilizador		
Pré-condição:	-		
Pós-condição:	Utilizador recebe palavra-passe por email.		
	Passo	Ator	Sistema
Fluxo Normal	1.	Seleciona a opção de recuperar palavra-passe.	
	2.		Apresenta página de recuperação da palavra-passe.
	3.	Introduz email da conta que pretende recuperar a palavra-passe.	
	4.		Verifica existência de uma conta com o email fornecido.
	5.		Envia palavra-passe para o email do utilizador.
Fluxo de Exceção [Não existe uma conta com o email fornecido] (passo 4)	4.1.		Verifica que não existe uma conta com o email fornecido.
	4.2		Apresenta informação sobre a inexistência de uma conta com o email fornecido.

Figura 6.4: Recuperar palavra-passe

Use Case			
Use Case:	Ver Perfil		
Ator:	Utilizador		
Pré-condição:	Utilizador autenticado		
Pós-condição:	Apresenta perfil de um utilizador.		
	Passo	Ator	Sistema
Fluxo Normal	1.	Seleciona a opção para ver o próprio perfil.	
	2.		Apresenta perfil do utilizador e funcionalidades extra.
	3.	Visualiza o perfil ( <i>extras perfil proprio</i> ).	
Fluxo Alternativo [Visualização de perfil de outro utilizador] (passo 1)	1.1.	Seleciona a opção para ver o perfil de outro utilizador.	
	1.2.		Apresenta perfil do utilizador.
	1.3.	Visualiza o perfil.	

Figura 6.5: Ver Perfil

Use Case			
Use Case:	Alterar Dados da Conta		
Ator:	Utilizador		
Pré-condição:	Utilizador autenticado e não está bloqueado		
Pós-condição:	Atualiza dados do utilizador		
	Passo	Ator	Sistema
Fluxo Normal	1.	Seleciona opção de alterar o perfil.	
	2.		Apresenta página com os campos de todas as informações que é possível alterar: email, nome de utilizador e palavra-passe. Apresenta um campo extra que é o de confirmação de palavra-passe.
	3.	Altera informações pretendidas, e clica em "Gravar".	
	4.		Se o utilizador introduziu um novo email verifica que não pertence a uma conta já existente. No caso de ter preenchido os campos relativos à palavra-passe verifica se são iguais.
	5.		Atualiza dados do utilizador.
Fluxo de Exceção [Dados introduzidos não cumprem requisitos] (passo 4)	4.1.		Verifica que existe algum erro nos dados introduzidos: já existe uma conta com o email introduzido, palavras-passes não coincidem ou o número de caracteres supera o número imposto para o campo. Informa o utilizador.

Figura 6.6: Alterar Dados da Conta

Use Case			
Use Case:	Apagar Conta		
Ator:	Utilizador		
Pré-condição:	Utilizador autenticado e não está bloqueado		
Pós-condição:	Dados do utilizador removidos.		
	Passo	Ator	Sistema
Fluxo Normal	1.	Seleciona a opção para remover conta.	
	2.		Apresenta caixa de diálogo para confirmação.
	3.	Confirma a intenção de eliminar a conta.	
	4.		Remove dados do utilizador, excepto as reviews. Atualiza autor das reviews deste para "Utilizador apagado".
	5.		Redireciona utilizador para página de início de sessão.

Figura 6.7: Apagar Conta

Use Case			
Use Case:	Procurar utilizador		
Ator:	Utilizador		
Pré-condição:	Utilizador autenticado		
Pós-condição:	Apresentado perfil do utilizador pesquisado		
	Passo	Ator	Sistema
Fluxo Normal	1.	Seleciona a opção para procurar utilizador.	
	2.		Apresenta menu de procura de utilizador.
	3.	Introduz dado identificador do utilizador a procurar.	
	4.		Procura utilizadores e apresenta lista.
	5.	Seleciona perfil que pretende visualizar (<<include>> Ver Perfil).	
Fluxo de exceção [Não foram encontrados resultados] (passo 4)	4.1.		Procura utilizadores, e verifica que não foram encontrados resultados.
	4.2.		Informa da falha na procura por utilizadores.

Figura 6.8: Procurar Utilizador

Use Case			
Use Case:	Ver Ponto de Interesse		
Ator:	Utilizador		
Pré-condição:	Utilizador autenticado e está no menu com a lista dos Pontos de Interesse		
Pós-condição:	Apresentado página do Ponto de Interesse		
	Passo	Ator	Sistema
Fluxo Normal	1.	Clica num ponto de interesse	
	2.		Apresenta o página do Ponto de Interesse
	3.	Observa página (extras ver PI)	

Figura 6.9: Ver Ponto de Interesse

Use Case			
Use Case:	Pedir Caminho		
Ator:	Utilizador		
Pré-condição:	Utilizador autenticado e encontra-se na página do Ponto de Interesse		
Pós-condição:	Utilizador redirecionado para aplicação externa que o encaminha para o ponto de interesse.		
	Passo	Ator	Sistema
Fluxo Normal	1.	Seleciona a opção pedir caminho.	
	2.		Redireciona para aplicação externa que consegue calcular o caminho entre o ponto de interesse e a localização pretendida pelo utilizador.

Figura 6.10: Pedir Caminho

Use Case			
Use Case:	Guardar/Remover Ponto de Interesse		
Ator:	Utilizador		
Pré-condição:	Utilizador autenticado, não está bloqueado e utilizador encontra-se na página do Ponto de		
Pós-condição:	Ponto de Interesse adicionado/removido da lista do utilizador.		
	Passo	Ator	Sistema
Fluxo Normal	1.	Ponto de interesse ainda não foi guardado. Clica na opção para guardar o Ponto de Interesse.	
	2.		Adicionado ponto de interesse à lista do utilizador.
Fluxo Alternativo [Pretende remover ponto de interesse]	1.1.	Ponto de interesse já está guardado. Clica na opção para remover o Ponto de Interesse da lista de pontos de interesse guardados.	
	1.2.		Removido ponto de interesse da lista do utilizador.

Figura 6.11: Guardar/Remover Ponto de Interesse

Use Case			
Use Case:	Procurar Pontos de Interesse		
Ator:	Utilizador		
Pré-condição:	Utilizador autenticado		
Pós-condição:	Sistema lista pontos de interesse		
	Passo	Ator	Sistema
Fluxo Normal	1.	Seleciona a opção de procura de pontos de interesse.	
	2.	Insere nome específico para procura.	
	3.	Não pretende aplicar filtros de procura.	
	4.	Clica em "Procurar".	
	5.		Procura os pontos de interesse na base de dados que obedecem aos filtros e mostra os resultados
	6.	Procura na lista pelo ponto de interesse.	
Fluxo Alternativo 1 [Pretende procurar pontos próximos de uma localização específica] (passo 1)	1.1.	Seleciona a opção de procura de pontos proximos de uma localização.	
	1.2.		Apresenta barra de procura de localização.
	1.3.	Insere nome da localização se não pretender usar a sua localização atual.	
	1.4.	Volta ao passo 3.	
Fluxo Alternativo 2 [Pretende aplicar filtros de procura] (passo 3)	3.1.	Seleciona opção de escolher filtros.	
	3.2.		Apresenta filtros.
	3.3.	Seleciona os filtros pretendidos.	
	3.4.		Aplica lista de filtros.
Fluxo Alternativo 3 [Pretende ordenar a procura] (passo 6)	3.5.	Volta ao passo 4.	
	6.1.	Aplica uma ordenação á sua escolha	
Fluxo Exceção [Não existem pontos de interesse que obedecem aos filtros] (passo 5)	6.2.		Ordena a lista dos pontos de interesse
	5.1.		Não encontra pontos de interesse que obedecem aos filtros. Avisa utilizador que nada foi encontrado.

Figura 6.12: Procurar Pontos de Interesse

Use Case			
Use Case:	Adicionar Review		
Ator:	Utilizador		
Pré-condição:	Utilizador autenticado, não está bloqueado e encontra-se no menu expandido do Ponto de Interesse		
Pós-condição:	Review registada.		
	Passo	Ator	Sistema
Fluxo Normal	1.	Clica em "Adicionar Review".	
	2.		Apresenta página de escrita de review.
	3.	Escreve review e clica em "Publicar".	
	4.		Regista review
Fluxo Exceção [Já possui review para este PI] (passo 2)	2.1.		Deteta que o utilizador já fez uma review sobre o Ponto de Interesse
	2.2.		Informa o utilizador da impossibilidade de escrever mais do que uma review.

Figura 6.13: Adicionar Review

Use Case			
Use Case:	Editar Review		
Ator:	Utilizador		
Pré-condição:	Utilizador autenticado, não está bloqueado e trata-se de uma das suas reviews		
Pós-condição:	Review atualizada.		
	Passo	Ator	Sistema
Fluxo Normal	1.	Seleciona a opção para alterar a review.	
	2.		Apresenta menu de escrita de review.
	3.	Altera os campos da review, e clica em "Gravar".	
	4.		Atualiza os dados da review.

Figura 6.14: Editar Review

Use Case			
Use Case:	Reportar Review		
Ator:	Utilizador		
Pré-condição:	Utilizador autenticado e não está bloqueado		
Pós-condição:	"Report", referente à review, adicionado		
	Passo	Ator	Sistema
Fluxo Normal	1.	Clica no botão de "Report" referente a uma review	
	2.		Regista report da review.
Fluxo de Exceção [O utilizador já deu "Report" na Review] (passo 2)	2.1.		Remove report.

Figura 6.15: Reportar Review

Use Case			
Use Case:	Alterar o "Gosto" na Review		
Ator:	Utilizador		
Pré-condição:	Utilizador autenticado e não está bloqueado		
Pós-condição:	"Gosto" adicionado/retirado no sistema		
	Passo	Ator	
Fluxo Normal	1.	Utilizador não tem "Gosto" na review e clica no botão de "Gosto" referente a uma review.	
	2.		Registado "gosto" no sistema.
Fluxo alternativo [Utilizador já tem gosto na review] (passo 2)	2.1.		Remove registo do "gosto" da review do sistema.

Figura 6.16: Alterar o "Gosto" na Review

## Use Cases de Moderador

Use Case			
Use Case:	Ver reports		
Ator:	Moderador		
Pré-condição:	O moderador tem sessão iniciada		
Pós-condição:	Não existem mais reviews com reports		
	Passo	Ator	Sistema
Fluxo Normal	1.	Seleciona opção para ver lista de reports	
	2.		Apresenta a lista de reports
	3.	Seleciona uma review reportada	
	4.		Apresenta a review em causa
	5.	Marca como falso report	
	6.		Retira da review a sinalização de reportada
	7.	Volta ao passo 1	
Fluxo alternativo 1 (considera a review inapropriada)	5.1.	Considera a review inapropriada	
	5.2.	Seleciona o perfil do utilizador	
	5.3.		Apresenta o perfil utilizador
	5.4.	Considera que é um perfil normal	
	5.5.	Seleciona a opção de remover a review	
	5.6.		Remove a review e envia notificação por email ao autor
	5.7.		Remove todos os reports sobre esta review
Fluxo de alternativo 2 (considera que o utilizador é repetente)	5.8.	Volta ao passo 1	
	5.4.1.	Considera que o perfil contém várias reviews inapropriadas	
	5.4.2.	Seleciona a opção de bloquear o utilizador	
	5.4.3.		Bloqueia conta do utilizador, remove todas as reviews do utilizadores e notifica utilizador por email.
	5.4.4.		Elimina registos de reports sobre esta review
	5.4.5.	Volta ao passo 1	
	3.1.	Não existem mais reviews	
Fluxo alternativo 3 (Não existem mais reviews reportadas)	3.2.	Volta a pagina principal	
	3.3.		Apresenta página principal

Figura 6.17: Ver Reports

## Use Cases de Administrador

Use Case			
Use Case:	Criar conta de moderador		
Ator:	Administrador		
Pré-condição:	Ator autenticado		
Pós-condição:	Conta de moderador registada		
	Passo	Ator	Sistema
Fluxo Normal	1.	Seleciona a opção de registar conta de moderador.	
	2.		Apresenta página de criação de conta de moderador.
	3.	Introduz credenciais.	
	4.		Guarda as credenciais e envia confirmação para o email fornecido.
Fluxo Alternativo [email já existe ou palavras-passe não coincidem] (passo 4)	4.1.		Credenciais não cumprem os requisitos. Informa o utilizador do requisito não cumprido.
	4.2.	Volta ao passo 3.	

Figura 6.18: Criar Conta de Moderador

Use Case			
Use Case:	Eliminar conta de moderador		
Ator:	Administrador		
Pré-condição:	Ator autenticado		
Pós-condição:	Conta de moderador eliminada		
	Passo	Ator	Sistema
Fluxo Normal	1.	Seleciona a opção de eliminar conta de moderador	
	2.		Apresenta menu de eliminação de conta de moderador
	3.	Seleciona opção de listar moderadores	
	4.		Lista moderadores
	5.	Seleciona conta de moderador que pretende eliminar	
	6.		Apresenta caixa de diálogo, a pedir confirmação
	7.	Confirma a eliminação da conta	
	8.		Conta de moderador eliminada
Fluxo Alternativo [Seleciona opção de procurar moderador] (passo 3)	3.1.	Seleciona opção de procurar moderador.	
	3.2.		Mostra barra de procura
	3.3.	Introduz dados identificadores da conta	
	3.4.		Mostra resultados da procura
	3.5.	Volta ao passo 5.	

Figura 6.19: Eliminar conta de moderador