



Universidade do Minho
Escola de Engenharia
Mestrado Integrado em Engenharia Informática

Unidade Curricular de Laboratórios de Informática IV

Ano Letivo de 2021/2022

RestauraçãoUM

Guilherme Fernandes Gonçalo Soares Tiago Costa Diogo Pires Pe-
dro Tavares

29 de novembro de 2021

LI4

Data de Receção	
Responsável	
Avaliação	
Observações	

RestauraçãoUM

Guilherme Fernandes Gonçalo Soares Tiago Costa Diogo Pires Pedro Tavares

29 de novembro de 2021

Resumo

Este relatório foi elaborado no contexto de um guia de localização de restaurantes que facilita a descoberta e posterior escolha de um serviço de restauração.

Numa primeira fase foi feita uma abordagem ao problema de modo a descobrir o tema sobre o qual nos iríamos debruçar. Após a escolha do tema, foi feita uma descrição do modelo do sistema, realizada uma descrição do contexto e identificação de métricas para a avaliação do desempenho do sistema e do seu sucesso.

A segunda fase marcou o início do processo de materialização das ideias e a sua passagem para algo mais formal. Assim, foram levantados e analisados os requisitos funcionais e não funcionais do sistema. Em seguida, estes requisitos foram transpostos para casos de uso concretos que, quando apresentados e especificados, em conjunto com o modelo de domínio, também desenvolvido, estabelecem as bases deste sistema. Em seguida, num processo iterativo e, com a preocupação de não restringir as possibilidades de escolha de linguagem para a implementação, foram desenvolvidos os diagramas de classe e sequência, que representam a estrutura do sistema e as interações das diferentes entidades. De modo a sustentar o armazenamento dos diferentes componentes presentes na aplicação, foi desenvolvido um esquema geral da estrutura do sistema de dados e, também, uma descrição detalhada dos seus vários elementos.

Por último, de forma a estudar a interação entre o nosso software e um utilizador delineamos algumas interfaces que consideramos necessárias. Com todo este extenso processo de modelação do projeto, foi nos possível antecipar problemas no sistema e desenvolver de um modo formal todo o longo trajeto que é o efetuado na engenharia de *software*.

Área de Aplicação: Guia de localização de restaurantes

Palavras-Chave: Engenharia de software, Restauração, Localização

Índice

1	Introdução	1
1.1	Contextualização	1
1.2	Motivação e Objectivos	2
1.3	Justificação e Utilidade do Sistema	2
1.4	Estabelecimento da Identidade do Projeto	3
1.5	Identificação dos Recursos Necessários	3
1.6	Maqueta do Sistema	4
1.7	Definição de um Conjunto de Medidas de Sucesso	4
1.8	Plano de desenvolvimento	5
2	Levantamento e Análise de Requisitos	6
2.1	Levantamento de Requisitos	6
2.2	Requisitos	7
2.2.1	Requisitos Funcionais	7
2.2.2	Requisitos Não Funcionais	9
3	Especificação e Modelação do Software	11
3.1	Modelo de Domínio	11
3.2	Diagrama de <i>Use Cases</i>	12
3.3	Especificação dos <i>Use Cases</i>	13
3.3.1	Registar na Aplicação	13
3.3.2	Autenticar na aplicação	14
3.3.3	Consultar restaurantes no local atual	15
3.3.4	Consultar restaurantes de um local personalizado	16
3.3.5	Consultar especificação do restaurante	17
3.3.6	Consultar lista de restaurantes visitados	18
3.3.7	Marcar restaurante como visitado	18
3.4	Arquitetura da Aplicação	19
3.5	Diagramas de Sequência	20
3.5.1	Registar na Aplicação	20
3.5.2	Autenticar na Aplicação	21
3.5.3	Consultar Restaurantes numa Área	22
3.5.4	Consultar Lista de Restaurantes Visitados	23
3.5.5	Marcar Restaurante como Visitado	23
3.6	Diagrama de Classes da Camada de Negócio	24

4	Conceção do Sistema de Dados	25
4.1	Apresentação Geral da Estrutura do Sistema de Dados	25
4.2	Descrição Detalhada dos Vários Elementos de Dados e Seus Relacionamentos .	26
5	Esboço das interfaces do sistema	28
6	Conclusões e Trabalho Futuro	33

Lista de Figuras

1.1	Maqueta do Sistema	4
1.2	Diagrama GANTT	5
3.1	Modelo de Domínio	11
3.2	Diagrama de Use Cases	12
3.3	Arquitetura da Aplicação	19
3.4	Diagrama de Sequência - Registrar na Aplicação	20
3.5	Diagrama de Sequência - Autenticar na Aplicação	21
3.6	Diagrama de Sequência - Consultar Restaurantes numa Área	22
3.7	Diagrama de Sequência - Consultar Lista de Restaurantes Visitados	23
3.8	Diagrama de Sequência - Marcar Restaurante como Visitado	23
3.9	Diagrama de Classes	24
4.1	Diagrama ER (Conceptual)	26
4.2	Diagrama Lógico	27
5.1	Interface da página inicial	28
5.2	Interface de registo	29
5.3	Interface de autenticação	29
5.4	Interface de escolha de onde procurar	30
5.5	Interface de escolha da localização e raio	30
5.6	Interface do mapa, com restaurantes	31
5.7	Interface da informação de um restaurante	31
5.8	Interface do perfil do utilizador	32

1 Introdução

1.1 Contextualização

O turismo é uma das principais fontes de rendimento de várias zonas e países na Europa, principalmente no sul. Este fenómeno tem crescido graças aos baixos preços de voos e uma maior facilidade nas viagens, como ausência de passaportes dentro da União Europeia.

Tirando o período da pandemia, o turismo tem sido uma área em expansão, o que faz aumentar os investimentos neste setor. Também os países já se promovem como destinos de férias de modo a cativarem turistas estrangeiros.

Zonas como Paris e Roma dependem do turismo, sendo visível pela quantidade de lojas dedicadas a viajantes tanto nacionais como estrangeiros. A média da União Europeia em termos de receitas vindas do turismo, proporcionalmente ao PIB, é 5.7% em 2019. Nesse ano, o país com mais receitas provenientes do turismo é a Croácia, com 18.44% do PIB, e o país com menos é a Alemanha, com 1.51%.

Analisando a situação portuguesa, em 2019 (antes da pandemia), existiam 7155 estabelecimentos de alojamento turístico, mais 19.2% que no ano anterior, segundo o Instituto Nacional de Estatística, 2020. Para além disso, este setor empregava 834 mil pessoas de forma indireta, 18% do emprego total a nível nacional. Diretamente, empregava 6% do total nacional.

Mas existem dificuldades que se mantêm sempre neste setor, como a língua e o desconhecimento dos locais a visitar por parte dos visitantes.

Apesar de existirem postos de turismo em várias cidades, a maior parte dos turistas já utiliza aplicações para planear as suas férias. Quer seja para escolher o que visitar, onde dormir, ou consultar os espetáculos na zona, é possível procurar tudo isto online. Desta forma, os turistas podem planear mais confortavelmente, em sua casa, e na sua língua materna, o que pretendem fazer.

Por isso, também as plataformas relacionadas com o turismo cresceram em termos de utilizadores, com sites/aplicações bem conhecidos como *TripAdvisor* e *Airbnb* a servirem de exemplo.

1.2 Motivação e Objectivos

Os problemas que um viajante tem ao chegar a um local que não conhece são bastantes, podendo prejudicar a imagem que terá desse local. Por outro lado, a quantidade de hipóteses existentes também dificulta a escolha dum restaurante em específico. Um desses casos é Lisboa, que nas zonas mais turísticas (como a Rua Augusta), tem vários restaurantes próximos uns dos outros. Os vários estabelecimentos são bastante parecidos entre si, tendo como principal fator distintivo as ementas. Não conhecendo o local, a escolha do turista será influenciada por critérios que não a qualidade do restaurante.

Isto origina escolhas mal informadas, que fazem o turista ter uma ideia errada do local visitado, neste caso Lisboa, levando-o a não aproveitar tão bem as suas férias. Por isso, uma das nossas metas é apresentar mais criteriosamente as opções que o turista tem.

Do ponto de vista dos estabelecimentos, também existe uma concorrência muito forte nessas situações. Tirando guias oficiais e concursos, os restaurantes não têm forma de se destacarem da concorrência nos meios tradicionais. Por outro lado, vários restaurantes apresentam pouca informação online, publicando apenas o nome e localização. Esta ausência de publicidade prejudica-os por comparação a outros restaurantes que tenham mais informação disponibilizada. Por isso, também pretendemos dar a hipótese de destacar os restaurantes pela sua qualidade, através da opinião dos utilizadores.

Em suma, estes problemas dos estabelecimentos e dos turistas são a motivação para o desenvolvimento deste software, e pretendemos ajudar estas duas entidades a terem mais conhecimento uma da outra.

1.3 Justificação e Utilidade do Sistema

A RestauraçãoUM pretende corrigir os problemas enumerados anteriormente, indicando ao utilizador os vários locais de restauração à sua volta. Ao ter mais conhecimento sobre os vários locais, poderá sentir mais seguranças nas suas escolhas. Assim, terá uma experiência mais positiva nas suas férias.

Este software também pode ajudar os estabelecimentos a serem conhecidos, incentivando-os a publicarem informações sobre si. Com uma presença online podem-se destacar através de comentários de utilizadores, quando realmente tiverem qualidade. Assim, têm mais uma forma de atrair clientes.

Para facilitar esse processo, o utilizador poderá criar uma conta e consultar os restaurantes que visitou.

Assim, tanto os clientes como os proprietários de restaurantes podem ganhar com este software em vários locais do mundo.

1.4 Estabelecimento da Identidade do Projeto

A RestauraçãoUM é um software que fornece informações sobre os vários estabelecimentos de restauração à volta do utilizador, utilizando a sua localização. Mas existe como alternativa, caso o utilizador queira, a escolha de uma localização e a RestauraçãoUM apresenta os restaurantes perto de si, de uma forma prática e intuitiva. Algumas das informações apresentadas são: tipos de serviços (por exemplo, se permitem *takeaway*), horário de funcionamento, e contactos.

Também será possível criar uma conta do utilizador. Assim, poderá consultar os restaurantes que já visitou. Opcionalmente, o software desenvolvido pode armazenar os *ratings* que os utilizadores dão aos vários estabelecimentos. Por questões de privacidade, a conta terá uma palavra-passe protegida. Com este software, pretendemos facilitar as escolhas de um turista ao chegar a um local novo e desconhecido, de forma gratuita.

1.5 Identificação dos Recursos Necessários

Para o desenvolvimento desta aplicação foi necessário alocar vários recursos, de modo a cumprir os requisitos esperados. No nosso caso em particular, além dos recursos fundamentais para o desenvolvimento de qualquer software, também foram necessários recursos alusivos à área da restauração.

Inicialmente, foi necessário encontrar uma equipa de engenheiros de *software* experientes, responsáveis pela especificação do sistema de software. Com a especificação feita, foi preciso ter à disposição uma equipa de programadores para construir o software, como especificado. Todo este processo foi gerido por um gestor de projetos, garantindo a coerência, qualidade e cumprimento de prazos estabelecidos.

Também foi fundamental falar com indivíduos comuns, os utilizadores alvo do software, de modo a entender qual seria a melhor forma de interagir com o sistema e a sua utilidade; assim como com donos de restaurantes, para saber como preferiam que o seu restaurante fosse apresentado aos utilizadores.

1.6 Maqueta do Sistema

O sistema será constituído por duas componentes: *Back-End* e *Front-End*.

O *Front-End* corresponde ao website ao qual o utilizador terá acesso, este funciona em qualquer dispositivo com um *browser*, o que facilita a utilização do sistema em qualquer lugar.

O *Back-End* é o servidor e toda a lógica que sustenta o *website*, estando sempre ativo e pronto para responder aos pedidos dos utilizadores. Armazena também e manipula todos os dados de utilizadores e restaurantes do sistema.

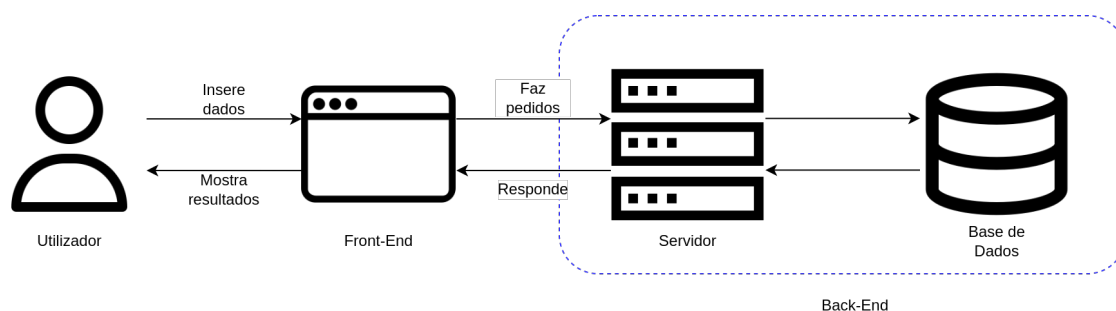


Figura 1.1: Maqueta do Sistema

1.7 Definição de um Conjunto de Medidas de Sucesso

A principal medida de sucesso será o número de contas criadas, e a sua utilização do nosso software. Usamos como termo de comparação outro site de recomendações turísticas, o “tripadvisor.com”, com vinte anos de existência, adequando os números à realidade nacional. Definimos como medida de sucesso termos 140 mil restaurantes listados, 385 mil visitas por mês e 2 milhões de utilizadores, ao fim de 20 anos. Também prevemos ter 2.47 milhões de visitas acumuladas em 20 anos.

Para além disso, esperamos um aumento da qualidade de informação disponibilizada pelos restaurantes, como as horas de funcionamentos e ementas.

Desta forma, prevemos melhorar a experiência dos turistas nas suas férias, utilizando o nosso software, e ser uma referência nas aplicações do setor turístico em Portugal.

1.8 Plano de desenvolvimento

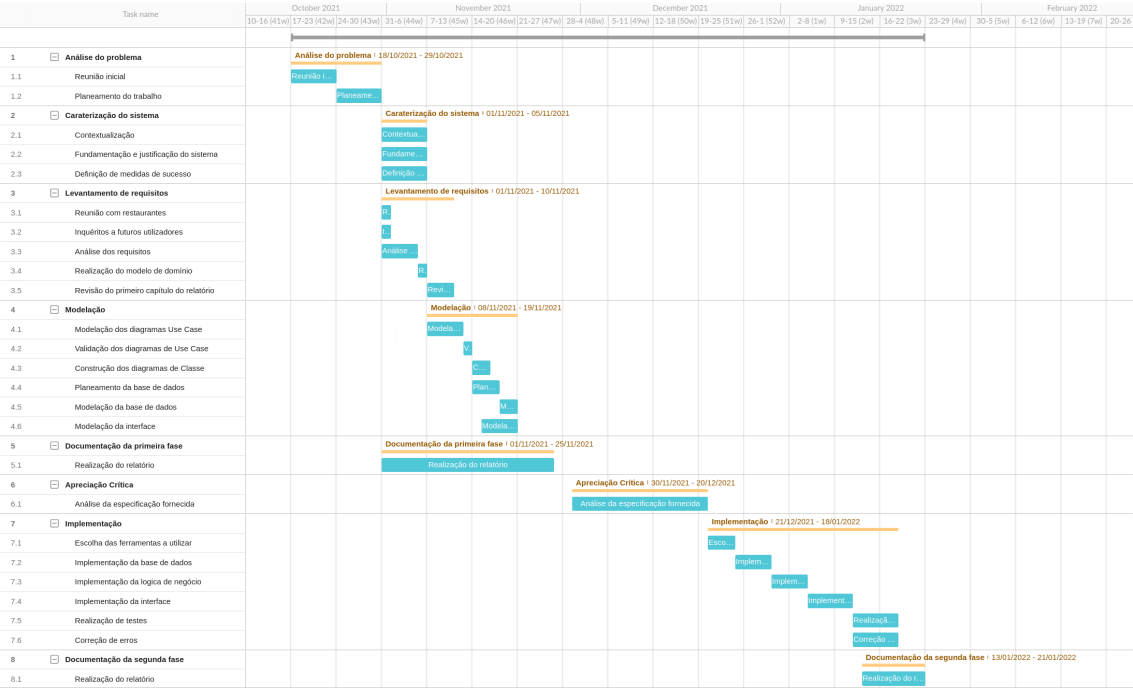


Figura 1.2: Diagrama GANTT

2 Levantamento e Análise de Requisitos

2.1 Levantamento de Requisitos

Os requisitos de um sistema são o que um sistema deve fazer, ou seja, os serviços que este proporciona e as restrições que derivam dessa oferta. Estes requisitos devem refletir as necessidades dos utilizadores mas também dos promotores (*stakeholders*).

Deste modo são de destacar alguns pontos que permitiram fazer uma recolha abrangente da grande maioria das exigências funcionais.

Em primeiro lugar, uma vez que o nosso sistema se foca em promover restaurantes, foram feitas várias entrevistas com donos de serviços de restauração de modo a perceber melhor quais as melhores formas de promover os seus restaurantes.

Em segundo, foram analisadas as diferentes estratégias que plataformas semelhantes à que vamos desenvolver implementam de modo a proporcionar ao utilizador final a maior facilidade possível nas suas deslocações.

Por fim, dado que também são o foco do sistema, realizamos alguns inquéritos a futuros utilizadores do sistema por forma a proporcionar uma experiência mais do seu agrado e também de modo a perspetivar quais as maiores necessidades que deveriam ser implementadas.

2.2 Requisitos

De acordo com Sommerville, 2011, os requisitos funcionais devem ser especificados em 2 tipos de requisitos dado que um dos maiores problemas que resulta deste processo de determinação de requisitos é a dificuldade de conseguir distinguir claramente os níveis de abstração. Por isso propõe 2 termos:

1. Requisitos do utilizador
2. Requisitos do sistema

Os requisitos do utilizador serão expostos numa linguagem natural e clara, de modo a explicitar o que o sistema deve providenciar ao utilizador.

Por outro lado, os requisitos do sistema são descrições mais detalhadas das funções, serviços e restrições operacionais do mesmo. Estes devem especificar exatamente o que o software tem que implementar.

De seguida, apresentamos então os requisitos funcionais de acordo como explicado anteriormente.

2.2.1 Requisitos Funcionais

2.2.1.1 Registo na aplicação

- Definição de requisitos de utilizador
 1. O utilizador tem que se registar na aplicação para a poder utilizar.
- Especificações de requisitos do sistema
 - 1.1. Aquando do registo, o sistema deve solicitar os dados necessários ao utilizador, nomeadamente o *username* e *password*.
 - 1.2. Não deverá haver registos de novos utilizadores com *usernames* já registados.
 - 1.3. O sistema deve armazenar os dados introduzidos, criando assim o registo do utilizador.

2.2.1.2 Autenticação na aplicação

- Definição de requisitos de utilizador
 2. O utilizador deverá introduzir o seu *username* e a respetiva *password* para iniciar sessão na aplicação.
- Especificações de requisitos do sistema
 - 2.1. O sistema deve verificar a validade da tentativa de autenticação, verificando se o par *username/password* inseridos correspondem a algum utilizador existente no sistema e a *password* fornecida é correspondente à deste utilizador. Caso não correspondam, o sistema não deve permitir a autenticação.

2.2.1.3 Consultar restaurantes de um local personalizado

- Definição de requisitos de utilizador
 3. O utilizador deverá especificar a área onde pretende ver os restaurantes, indicando a localização central e o raio.
- Especificações de requisitos do sistema
 - 3.1. O sistema deve validar as informações introduzidas pelo utilizador e mostrar um mapa com os restaurantes circundantes à localização inserida.

2.2.1.4 Consultar restaurantes no local atual

- Definição de requisitos de utilizador
 4. O utilizador deverá conseguir visualizar os restaurantes que o rodeiam.
- Especificações de requisitos do sistema
 - 4.1. O sistema deve localizar o utilizador e proporcionar uma vista dos serviços de restauração que o envolvem.

2.2.1.5 Consultar especificação dos restaurantes

- Definição de requisitos de utilizador
 5. O utilizador deve conseguir consultar a descrição de cada restaurante.
- Especificações de requisitos do sistema
 - 5.1. Cada restaurante deve ter o tipo de serviços disponibilizados.
 - 5.2. Cada restaurante deve ter o horário de funcionamento associado.
 - 5.3. Cada restaurante deve disponibilizar as informações de contacto e morada.
 - 5.4. Cada restaurante deve disponibilizar o nível de preços praticado.
 - 5.5. O sistema deve também disponibilizar a distância à qual o restaurante está do utilizador.

2.2.1.6 Consultar lista de restaurantes visitados

- Definição de requisitos de utilizador
 6. O utilizador deve ter a possibilidade de poder observar todos os restaurantes por ele já visitados e também de marcar restaurantes como visitados.
- Especificações de requisitos do sistema
 - 6.1. O sistema deve conseguir guardar na base de dados os restaurantes marcados como visitados pelo utilizador.
 - 6.2. O sistema deve também conseguir mostrar todos os restaurantes visitados.

2.2.2 Requisitos Não Funcionais

Requisitos não funcionais são, tal como o nome indica, requisitos que não estão diretamente ligados com a especificação dos serviços dados aos utilizadores. São requisitos que definem as propriedades e restrições do sistema.

Foram então definidos os seguintes requisitos não-funcionais:

1. A nível de performance a aplicação deve ser o mais eficiente de modo a que os utilizadores não necessitem de esperar por uma resposta do sistema.
2. A aplicação deverá ser de fácil uso, com um *layout* o mais simples possível.

3. O sistema deve ser suportado por todos os *browsers*.
4. Por fim, a aplicação deverá estar disponível a qualquer altura.

3 Especificação e Modelação do Software

3.1 Modelo de Domínio

Com base na especificação, desenvolvemos este diagrama de domínio, para representar melhor as várias entidades e as relações entre elas, e explicamos algumas das nossas decisões. Este diagrama também permite mostrar algum do vocabulário usado neste relatório e no software. Desta forma, torna-se mais fácil para qualquer pessoa compreender o nosso software duma forma fácil e visualmente apelativa. Em primeiro lugar, o utilizador está registado numa

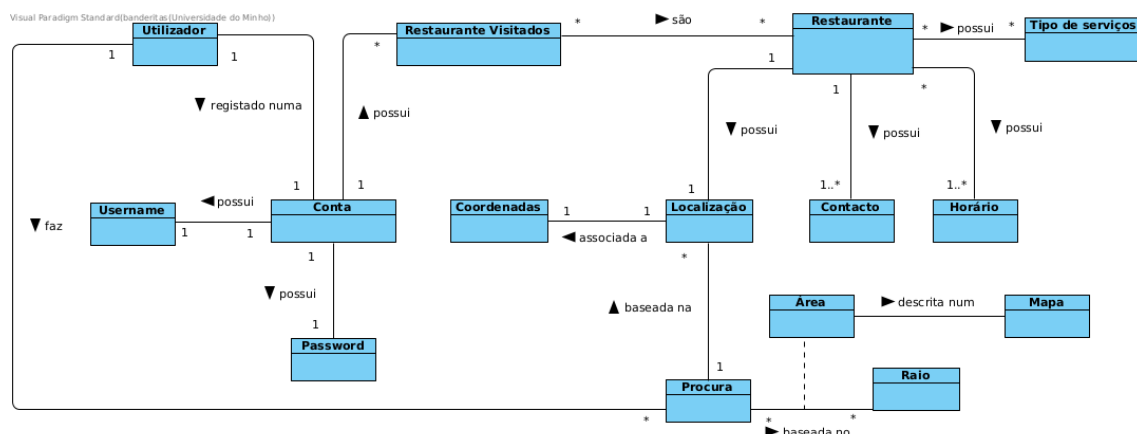


Figura 3.1: Modelo de Domínio

conta do nosso software. Essa conta tem de ter: username e password. Nesta conta é possível o utilizador guardar os restaurantes já visitados e fazer procuras de restaurantes. O restaurante também é uma entidade importante no nosso software, porque tem de ter as seguintes características: tipo de serviços; horário de funcionamento; um ou mais contactos, e localização. Essa localização é descrita por coordenadas. Outra característica importante do nosso software é a realização de procuras, realizadas pelo utilizador. A procura é baseada numa área, e detetam-se os restaurantes incluídos nessa área, representada num mapa. A área é calculada a partir dum raio à volta de uma localização.

3.2 Diagrama de *Use Cases*

O diagrama de Use Cases representado abaixo permite-nos ver as várias funcionalidades do programa. Para um utilizador não autenticado temos os casos de uso de registar e autenticar, que permitem ao utilizador ficar autenticado. Registrar e autenticar possibilitam ao utilizador criar uma conta e fazer login na mesma, correspondentemente.

Depois de estar dentro da conta, o utilizador pode consultar restaurantes na sua zona atual ou personalizada. Também pode obter informações sobre um restaurante, como nome, número de telefone, morada, descrição, etc. Por fim, o utilizador pode marcar um restaurante como visitado e, posteriormente, consultar a lista dos seus restaurantes visitados.

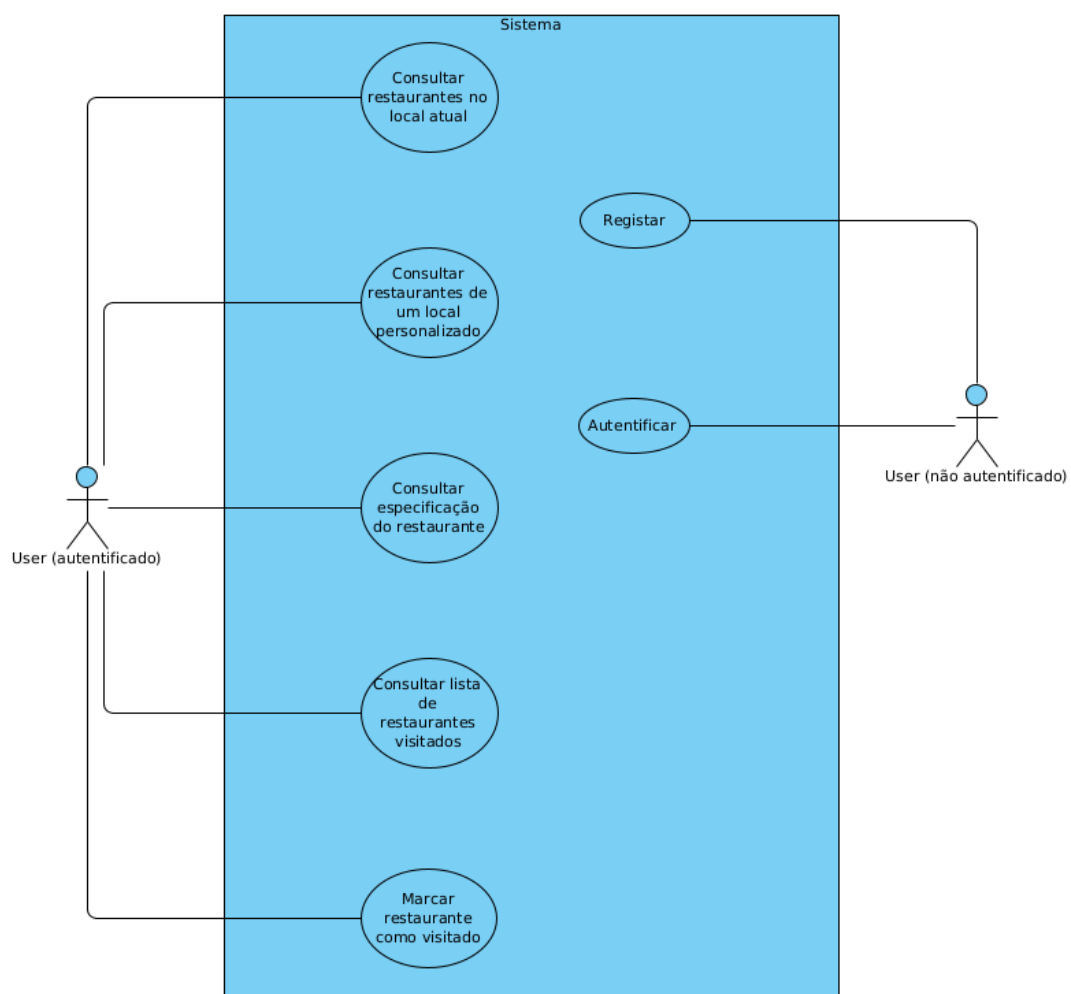


Figura 3.2: Diagrama de Use Cases

3.3 Especificação dos *Use Cases*

Procedemos então à descrição detalhada de cada caso de uso.

3.3.1 Registar na Aplicação

Descrição: O registo na aplicação permite ao utilizador criar uma conta com os dados necessários, de modo, a que possa utilizar a aplicação.

Pré-condição: True

Pós-condição: O utilizador fica com uma conta registada.

Fluxo normal:

1. Utilizador fornece o *username* e *password* duas vezes, para confirmação.
2. Sistema verifica que as *passwords* são iguais.
3. Sistema verifica que um utilizador com o mesmo *username* não existe.
4. Sistema efetua o registo do utilizador.
5. Sistema informa que conseguiu criar a conta.

Fluxo de exceção: [*passwords* não são iguais] (passo 2)

- 2.1 Sistema informa que as *passwords* não são iguais.

Fluxo de exceção [*username* já existe] (passo 3)

- 3.1 Sistema informa que um utilizador com o mesmo *username* já existe.

3.3.2 Autenticar na aplicação

Descrição: Qualquer utilizador que deseje utilizar a aplicação tem que está autenticado. Logo terá que fazer login com o seu username e password.

Pré-condição: True

Pós-condição: O utilizador fica autenticado.

Fluxo normal:

1. Utilizador fornece o *username* e *password*.
2. Sistema verifica se o *username* corresponde a alguma conta.
3. Sistema encripta a *password* fornecida.
4. Sistema verifica se a *password* encriptada corresponde à *password* (também encriptada) associada ao *username* fornecido pelo utilizador.

Fluxo de excepção: [*username* não corresponde a nenhuma conta] (passo 2)

- 3.1 Sistema indica que o *username* não corresponde a nenhuma conta.

Fluxo de excepção: [*password* não corresponde] (passo 3)

- 4.1 Sistema indica que a *password* introduzida pelo utilizador está errada.

3.3.3 Consultar restaurantes no local atual

Descrição: O utilizador poderá ver os restaurantes à sua volta, permitindo o acesso à sua localização.

Pré-condição: O utilizador está autenticado.

Pós-condição: Os restaurantes de uma dada área são representados para o utilizador.

Fluxo normal:

1. Utilizador permite que o sistema acesse à sua localização.
2. Sistema pede raio ao utilizador.
3. Utilizador fornece o raio.
4. Sistema valida a área.
5. Sistema mostra mapa com todos os restaurantes que pertencem à área.

Fluxo alternativo: [Utilizador não fornece raio] (passo 3)

- 3.1 Sistema usa raio *default*.
- 3.2 Voltar ao passo 4.

Fluxo de exceção: [Área inválida] (passo 4)

- 4.1 Sistema indica que a área é inválida.

3.3.4 Consultar restaurantes de um local personalizado

Descrição: O utilizador poderá ver os restaurantes num local personalizado e com um raio à sua escolha.

Pré-condição: O utilizador está autenticado.

Pós-condição: Os restaurantes de uma dada área são representados para o utilizador.

Fluxo normal:

1. Utilizador fornece a localização central e o raio.
2. Sistema valida a localização.
3. Sistema valida o raio.
4. Sistema calcula a área através da localização e do raio.
5. Sistema mostra mapa com todos os restaurantes que pertencem à área.

Fluxo alternativo: [Localização inválida] (passo 2)

- 3.1 Sistema indica que a localização é inválida.
- 3.2 Volta ao passo 1.

Fluxo alternativo: [Raio inválido] (passo 3)

- 3.1 Sistema indica que o raio é inválido.
- 3.2 Volta ao passo 1.

3.3.5 Consultar especificação do restaurante

Descrição: Sistema indica todos as informações de um restaurante à escolha do utilizador.

Pré-condição: O utilizador está autenticado.

Pós-condição: A especificação do restaurante é apresentada ao utilizador.

Fluxo normal:

1. Utilizador seleciona o restaurante.
2. Sistema indica o tipo de serviços disponibilizados.
3. Sistema indica horários de funcionamento associado.
4. Sistema indica as informações de contacto e morada.
5. Sistema indica o nível de preços praticado no restaurante.
6. Sistema indica a distância à qual o restaurante está do utilizador.

3.3.6 Consultar lista de restaurantes visitados

Descrição: O sistema mostra a lista de restaurantes visitados pelo utilizador até ao momento.

Pré-condição: O utilizador está autenticado.

Pós-condição: Lista com os restaurantes visitados é apresentada ao utilizador.

Fluxo normal:

1. Utilizador solicita lista de restaurantes visitados.
2. Sistema apresenta lista com os restaurantes visitados pelo utilizador até ao momento.

3.3.7 Marcar restaurante como visitado

Descrição: Utilizador marca restaurante como visitado e sistema armazena a informação.

Pré-condição: O utilizador está autenticado.

Pós-condição: Sistema guarda registo do restaurante marcado como visitado.

Fluxo normal:

1. Utilizador marca restaurante como visitado.
2. Sistema guarda registo do restaurante marcado como visitado pelo utilizador.

3.4 Arquitetura da Aplicação

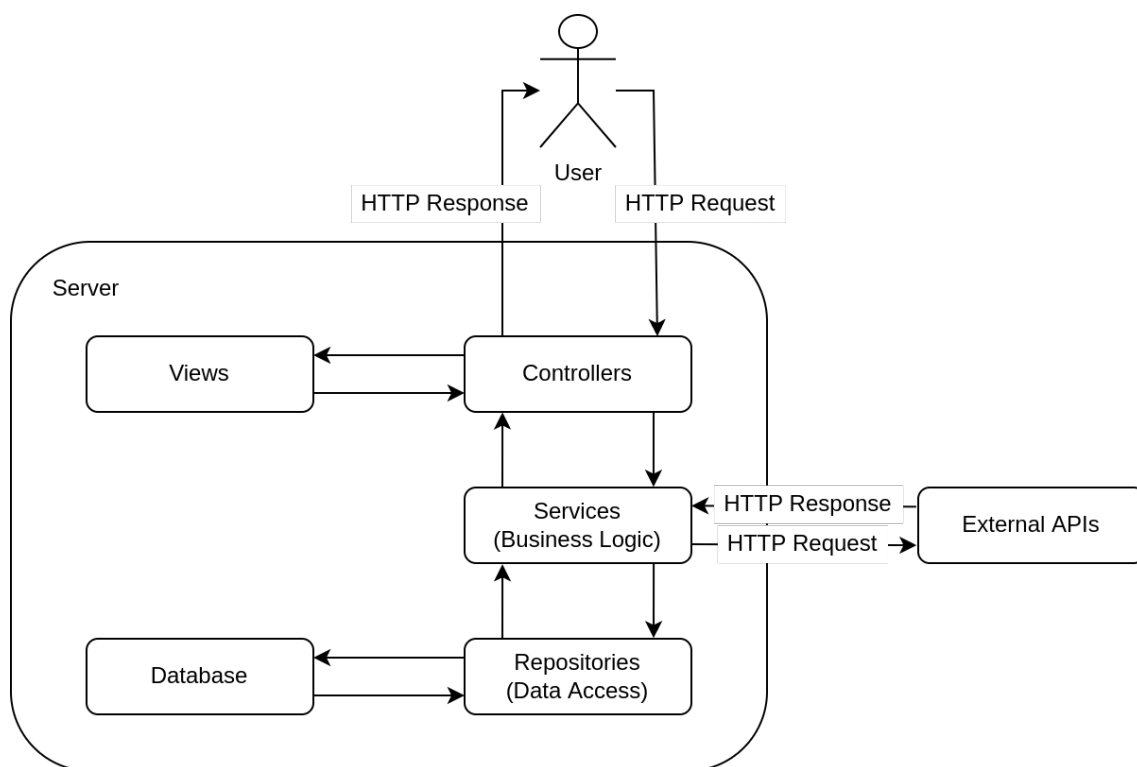


Figura 3.3: Arquitetura da Aplicação

3.5 Diagramas de Sequência

Tendo por base a especificação dos *Use Cases* e a arquitetura da aplicação, os diagramas de sequência permitem-nos representar a maior pormenor e de forma mais clara a interação entre os vários subsistemas da aplicação, assim como o fluxo de dados.

3.5.1 Registar na Aplicação

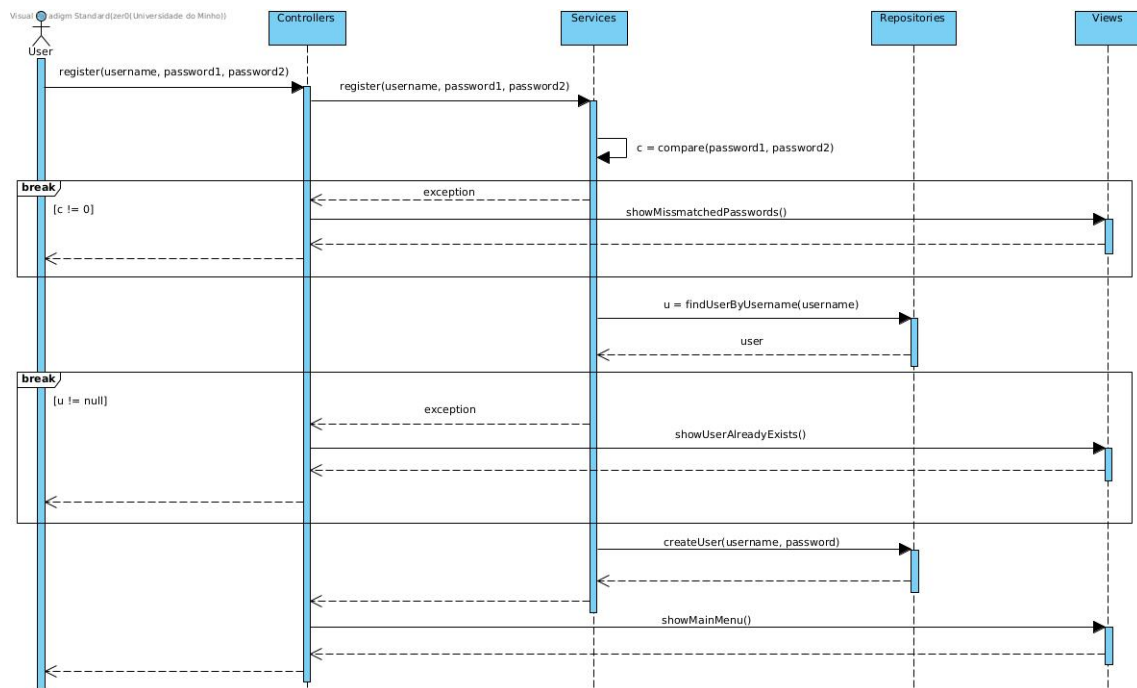


Figura 3.4: Diagrama de Sequência - Registar na Aplicação

3.5.2 Autenticar na Aplicação

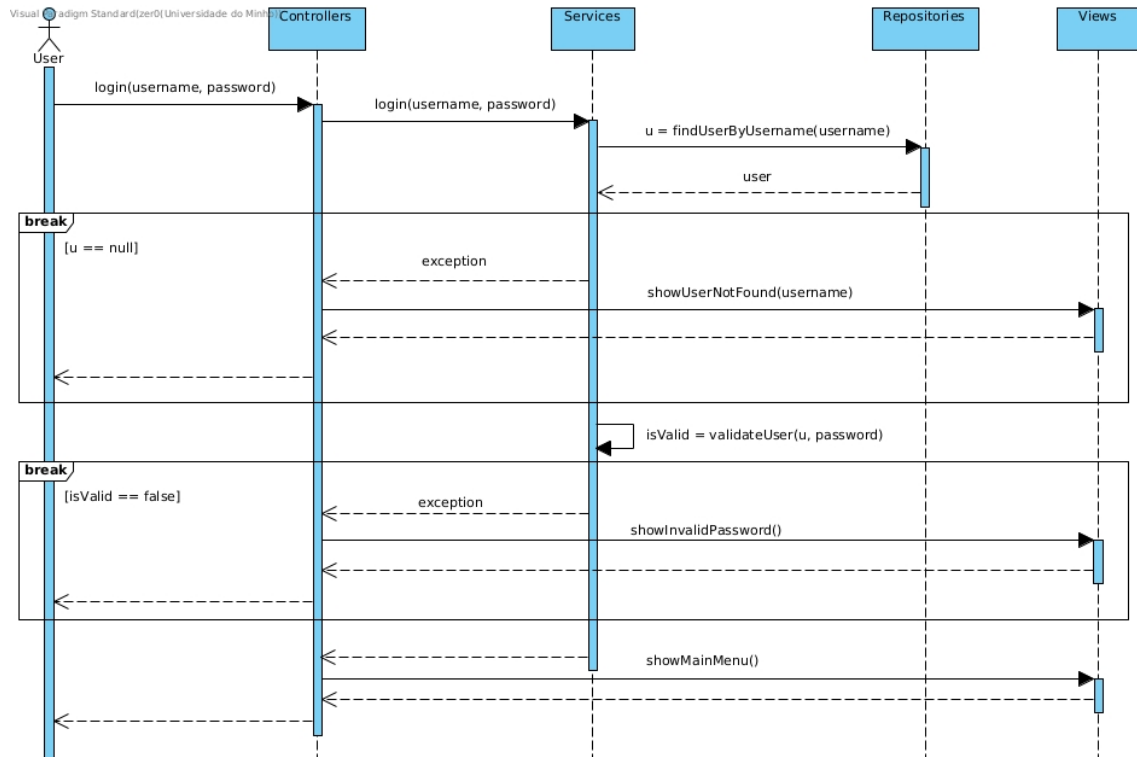


Figura 3.5: Diagrama de Sequência - Autenticar na Aplicação

3.5.3 Consultar Restaurantes numa Área

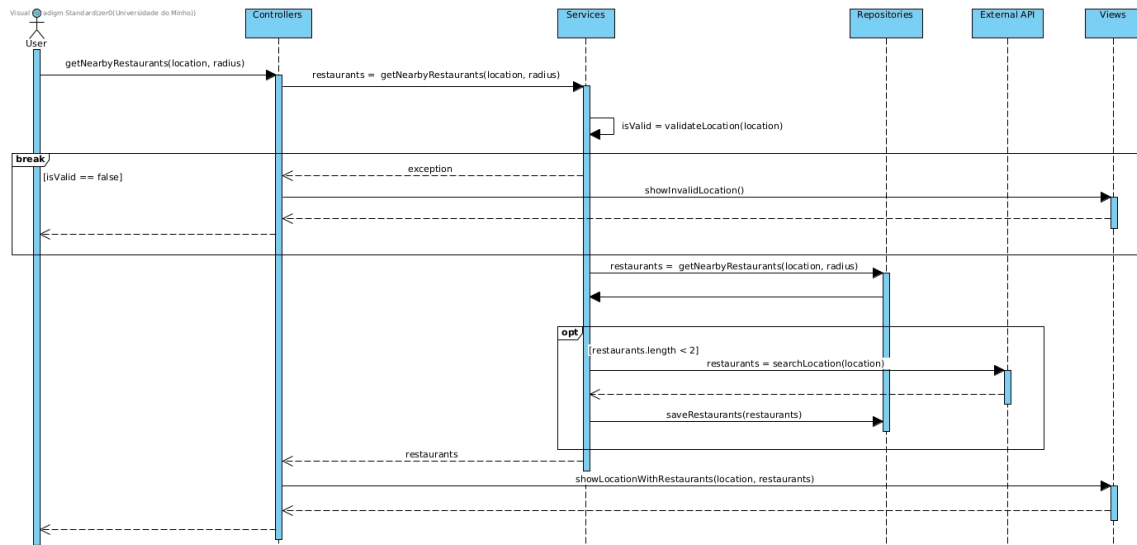


Figura 3.6: Diagrama de Sequência - Consultar Restaurantes numa Área

3.5.4 Consultar Lista de Restaurantes Visitados

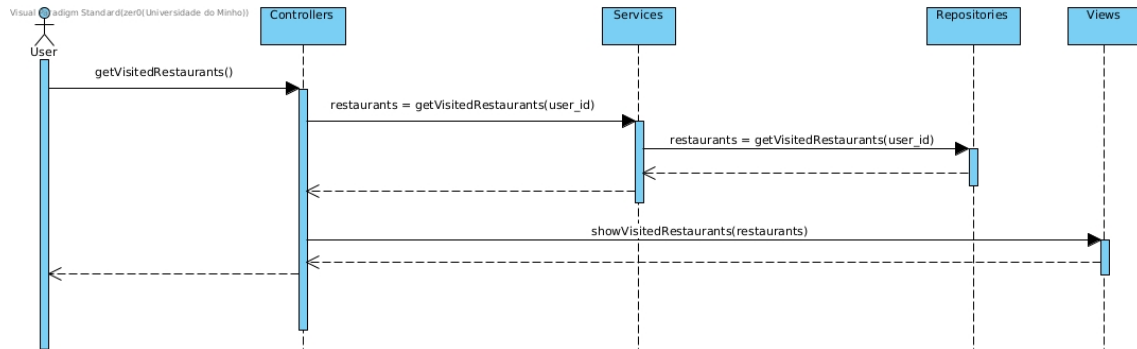


Figura 3.7: Diagrama de Sequência - Consultar Lista de Restaurantes Visitados

3.5.5 Marcar Restaurante como Visitado

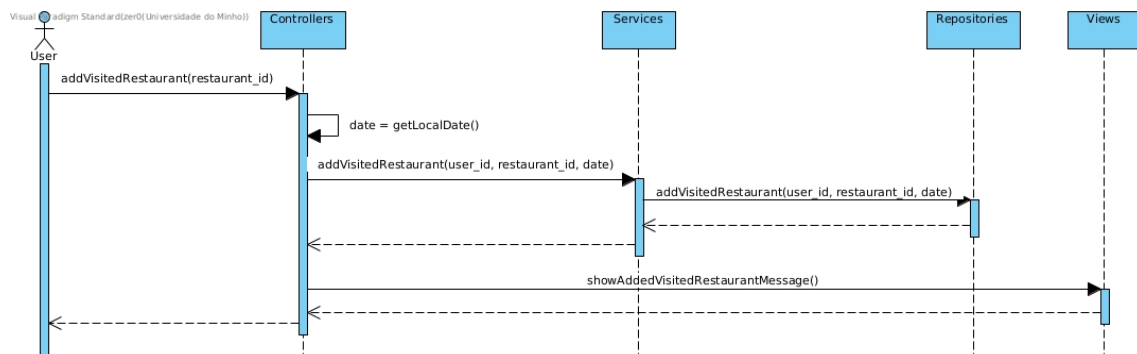


Figura 3.8: Diagrama de Sequência - Marcar Restaurante como Visitado

3.6 Diagrama de Classes da Camada de Negócio

Após a análise dos requisitos e do modelo de domínio apresentados, identificamos e caracterizamos as classes principais do sistema. Assim, temos o seguinte grupo de classes principais:

- **User** - Corresponde à representação no sistema dos utilizadores da aplicação e contém o seu username, password e id. Para além disto, possui também uma lista com os restaurantes visitados.
- **Restaurant** - Representa a informação de um restaurante, como por exemplo o id, nome, morada e contactos. Para a representação da localização é usada a classe Point que tem como atributos a latitude e longitude.
- **RestauraçãoUM** - É o facade que tem como objetivo mediar toda a camada de negócios para que a solução apresentada fosse modular.

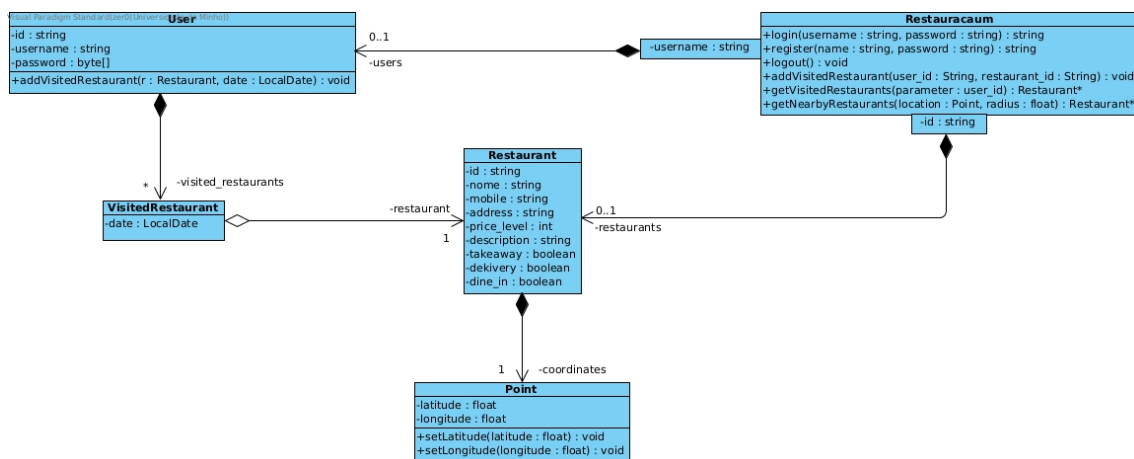


Figura 3.9: Diagrama de Classes

4 Conceção do Sistema de Dados

4.1 Apresentação Geral da Estrutura do Sistema de Dados

O planeamento da base de dados consistiu, primeiramente, na decisão da base de dados a usar, de seguida na realização de um Diagrama ER (conceptual), apesar deste não ser muito detalhado, é útil devido à sua fácil compreensão sendo assim possível discutir facilmente possíveis alterações com o cliente final.

Optamos por escolher uma base de dados relacional, uma vez que existem relações entre os dados que pretendemos guardar e, além disso, como não vai haver alterações dinâmicas futuras na estrutura, conseguimos facilmente estruturá-la. De entre as base de dados relacionais optamos por MySQL pela sua simplicidade, boa performance, *open-source*, segurança, utilização elevada e sem custos.

Após a realização do Diagrama ER, procedemos à realização do diagrama lógico que contém informação acerca dos tipos de cada atributo para além da informação já previamente obtida pelo Diagrama ER (relações, entidades e os seus atributos).

Aquando da realização de ambos os diagramas, são feitas constantemente verificações de modo a cumprir os requisitos. No final executamos alguns testes em *SQL* para provar que o nosso esquema de base de dados está implementado como planeado.

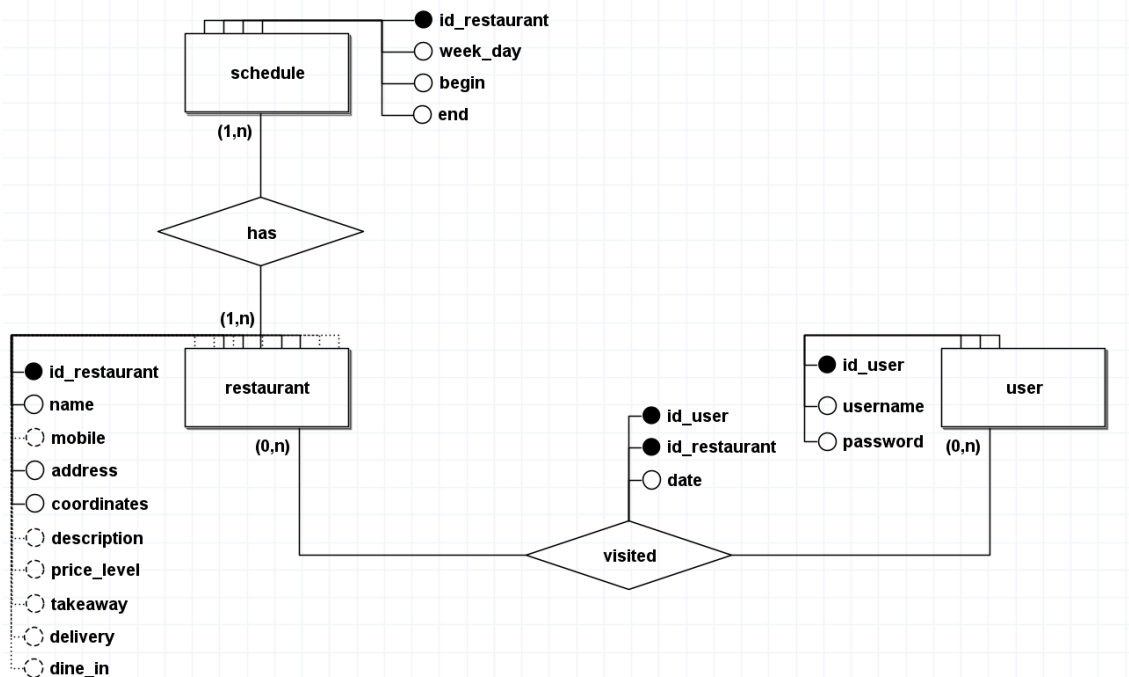


Figura 4.1: Diagrama ER (Conceptual)

4.2 Descrição Detalhada dos Vários Elementos de Dados e Seus Relacionamentos

A estruturação da base dados divide-se em duas tabelas principais relativas aos restaurantes e aos utilizadores.

Quanto aos restaurantes, guardamos o ID único, nome, telefone, morada, coordenadas, descrição, nível do preço e alguns valores lógicos tal como a possibilidade de consumir uma refeição dentro do restaurante, serviço de *takeaway* e entrega ao domicílio.

Uma vez que os restaurantes podem ter vários horários de funcionamento ao longo do dia, decidimos implementar uma tabela à parte para tal. Assim, a tabela dos restaurantes estabelece uma relação de *one to many non-identifying* contendo informação do dia e o respetivo horário. Sendo também possível criar vários horários de funcionamento diferentes para um único restaurante.

Em relação ao utilizador, além do ID único existirá também um `username` e uma `password` que deverá ser, previamente, passada por uma função de hash (BCrypt).

Adicionalmente criamos uma relação de *many to many identifying* entre os utilizadores e os restaurantes com o objetivo de salvaguardar os restaurantes visitados pelo utilizador e a sua última data respetiva.

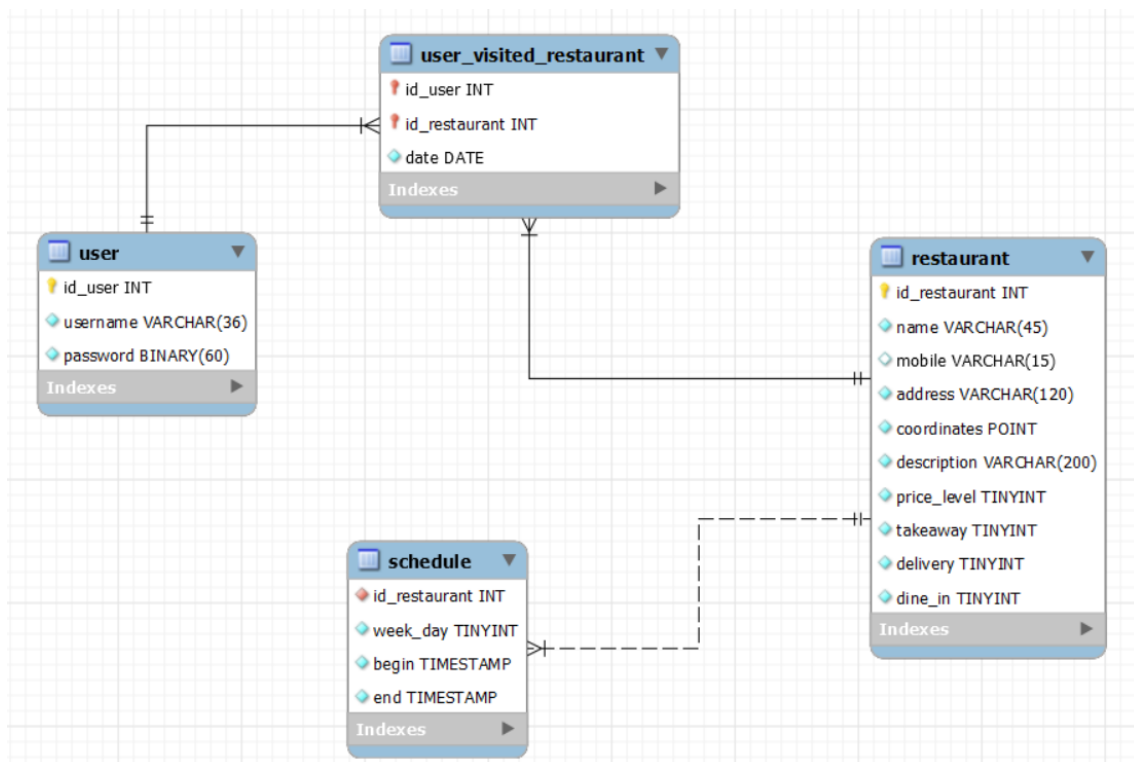


Figura 4.2: Diagrama Lógico

Optamos pela limitação dos caracteres no uso do tipo VARCHAR de modo a evitar a ocupação de espaço desnecessário.

Quanto às coordenadas, decidimos utilizar o tipo POINT, específico de *MYSQL*, para facilitar os cálculos geométricos em *SQL*.

5 Esboço das interfaces do sistema



Figura 5.1: Interface da página inicial



Bem vindo ao RestauraçãUM!

Nome de Utilizador

Palavra-passe [show](#)

Confirmar Palavra-passe [show](#)

Registrar

[Já tem uma conta? Autentique-se](#)

Figura 5.2: Interface de registo



Bem vindo de volta!

Nome de Utilizador

Palavra-passe [show](#)

Autenticar

[Criar uma conta](#)

Figura 5.3: Interface de autenticação

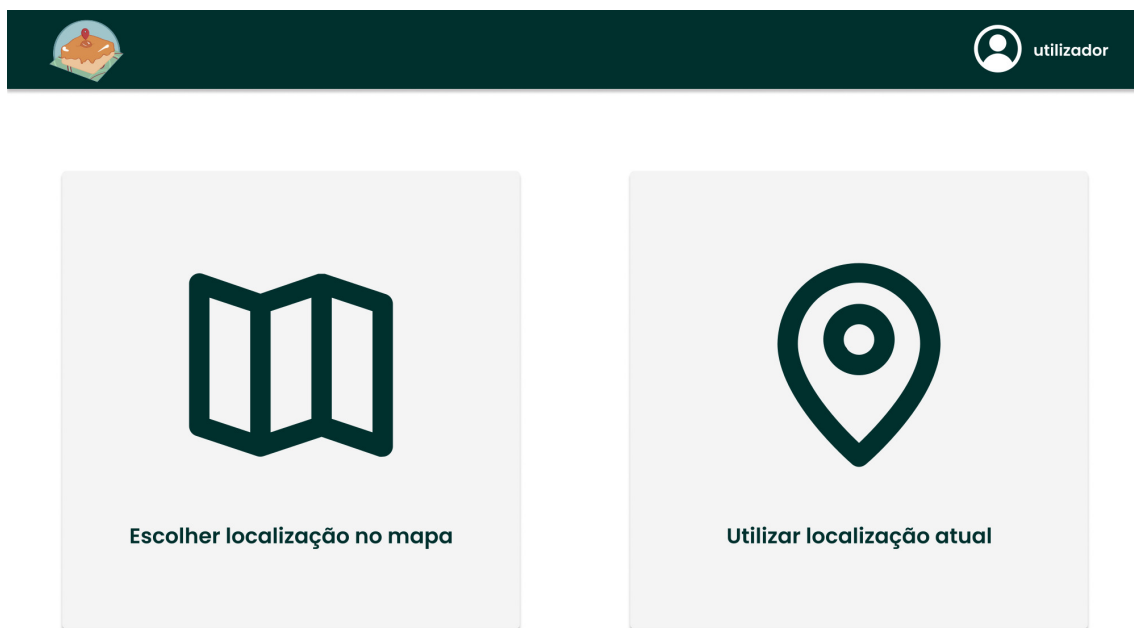


Figura 5.4: Interface de escolha de onde procurar

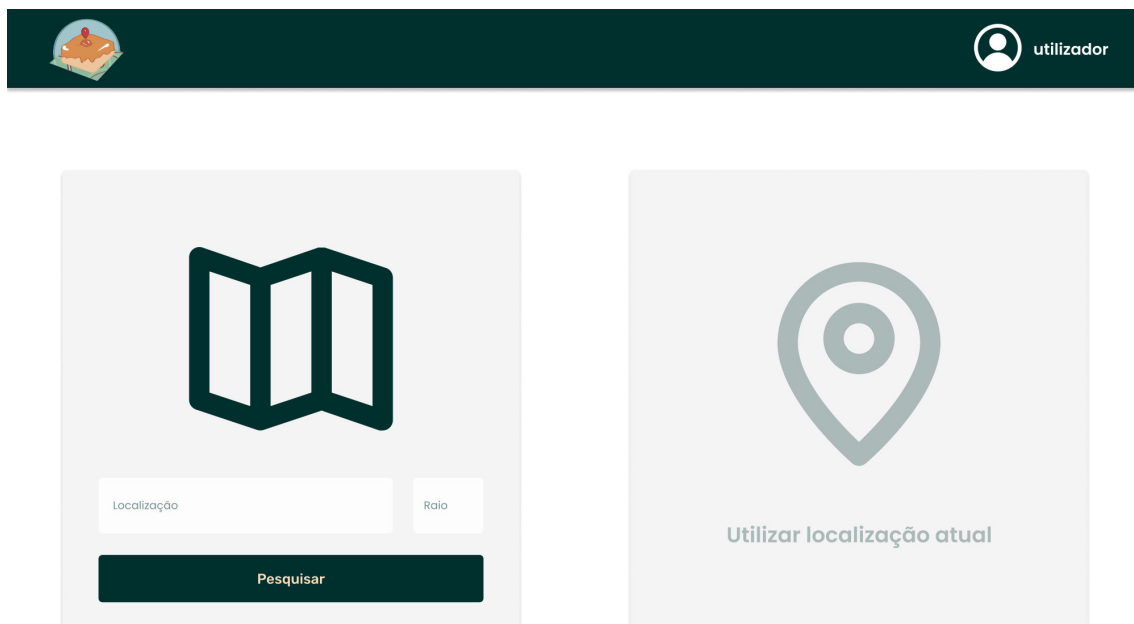


Figura 5.5: Interface de escolha da localização e raio

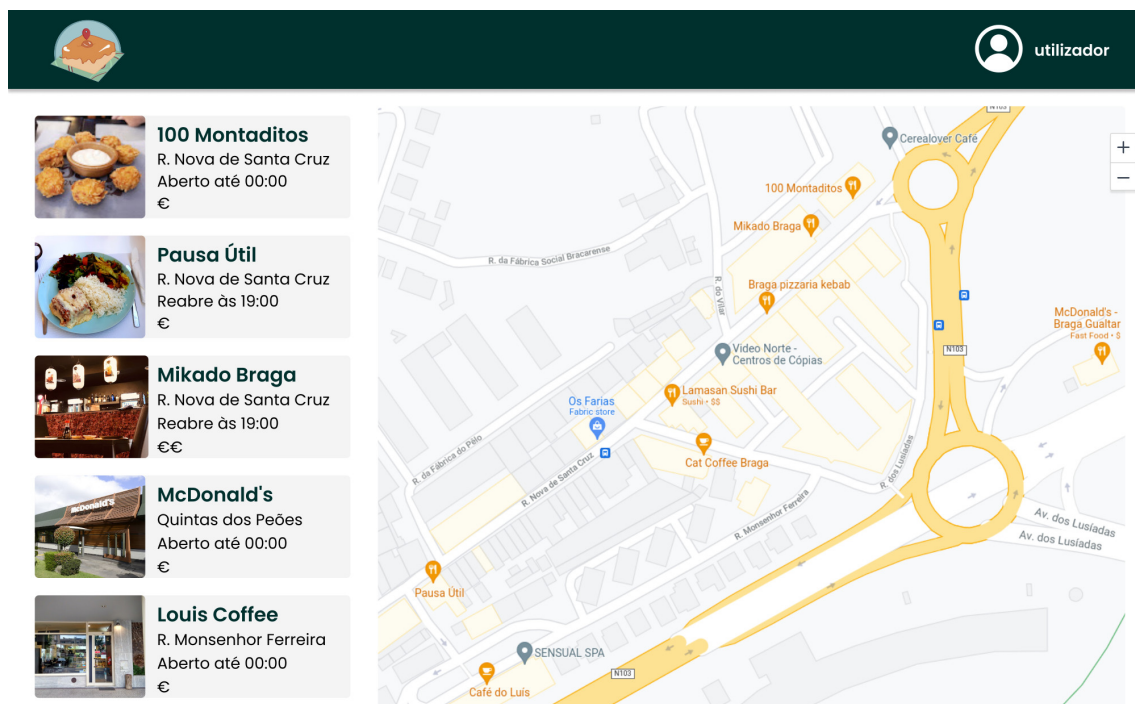


Figura 5.6: Interface do mapa, com restaurantes

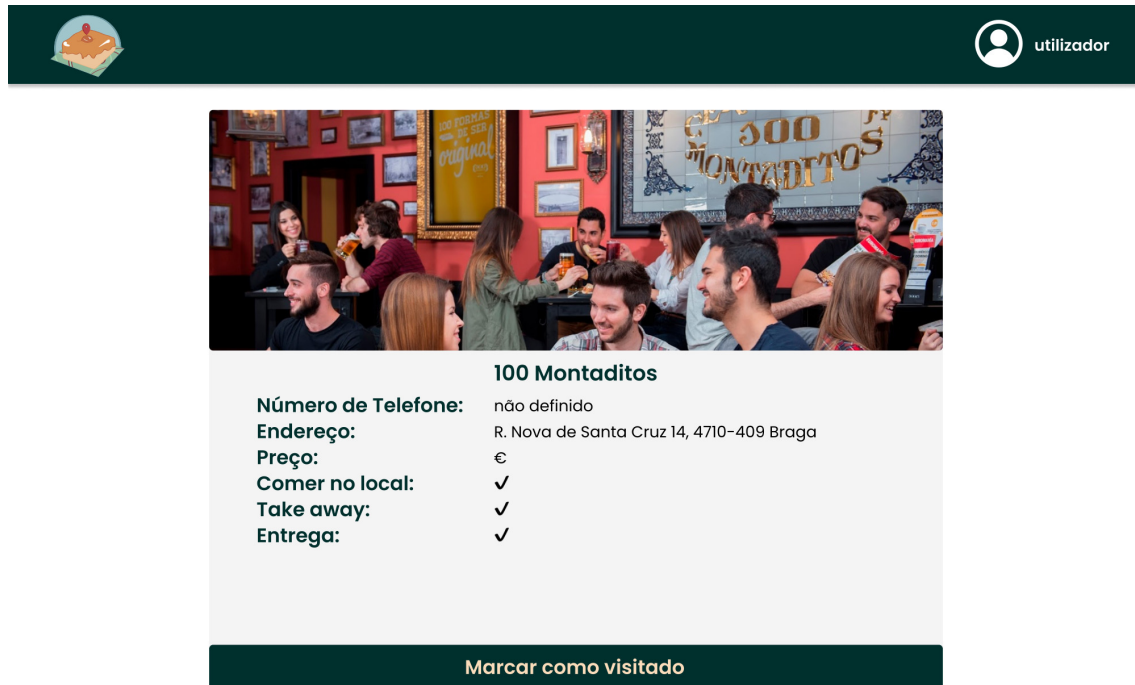


Figura 5.7: Interface da informação de um restaurante

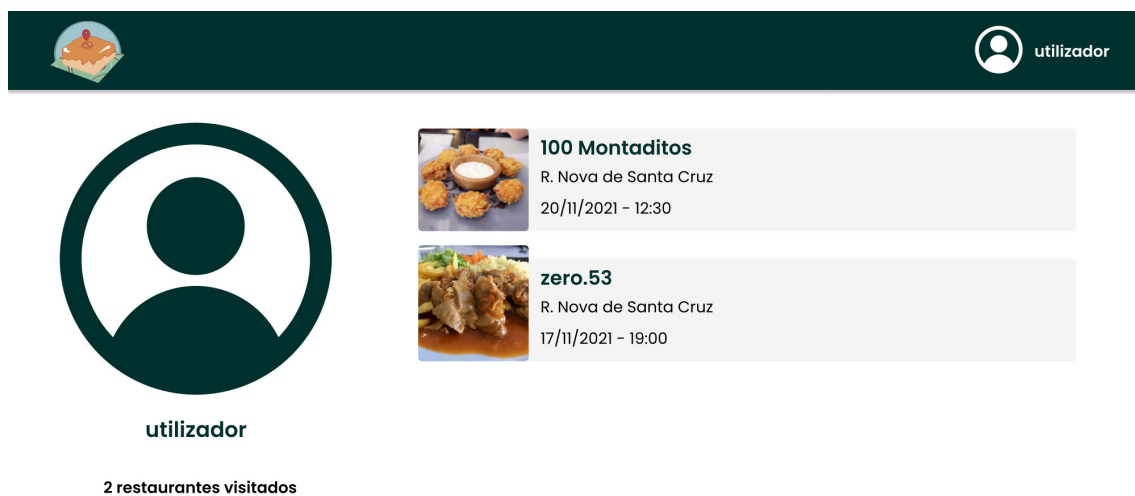


Figura 5.8: Interface do perfil do utilizador

6 Conclusões e Trabalho Futuro

A conclusão deste relatório marca também o término da especificação e da caracterização do nosso sistema.

Consideramos que, no geral, conseguimos modelar e especificar um software que se enquadre com o pedido pelos utilizadores.

Em primeiro lugar, caracterizamos o nosso sistema de uma forma consistente e também definimos qual a utilidade e medidas de sucesso. Posteriormente, e com alguma contenção devido ao tempo existente para a implementação do sistema, fizemos o levantamento dos requisitos que consideramos fundamentais para a nossa aplicação. No global, achamos que estes requisitos funcionais e não funcionais são simples, mas robustos e concisos de modo a definirmos uma boa estrutura de base para a implementação. No que diz respeito à especificação do sistema implementamos alguns diagramas UML que considerámos relevantes para uma melhor modelação e compreensão do sistema que irá ser desenvolvido. A representação do diagrama de casos de uso permitiu-nos exemplificar as relações do utilizador com o sistema e, assim, mostrar quais as possíveis interações que existem na aplicação. Com o diagrama de domínio podemos, de modo geral, avaliar quais os elementos que fazem parte do sistema. Com os diagramas de classes e de sequência que foram representados foi nos possível determinar que métodos devem ser implementados e como se processam as ações do utilizador com o sistema. Uma vez que este sistema será implementado por um grupo de outros engenheiros, desenvolvemos quer o diagrama de classes quer o diagrama de sequência de modo a não limitar as escolhas de ferramentas. O esboço das interfaces que foi apresentado serve também como um guia para o tipo de interface que queremos que seja implementada e como entendemos que o sistema se deve apresentar ao utilizador.

Para a próxima fase do projeto consideramos que apresentamos uma boa base sobre a qual é possível desenvolver o sistema que satisfaça os requisitos do utilizador.

Referências

Instituto Nacional de Estatística. (2020). *Estatísticas do Turismo - 2019*.

Similiar Web. (2021). *tripadvisor.com*. <https://www.similarweb.com/website/tripadvisor.com/>

Smith, C. (2021). *TripAdvisor Statistics, User Counts and Facts*. <https://expandedramblings.com/index.php/tripadvisor-statistics/>

Sommerville, I. (2011). *Software Engineering* (9ª ed.). Addison-Wesley.