



Universidade do Minho
Escola de Engenharia
Mestrado Integrado em Engenharia Informática

Unidade Curricular de Laboratórios de Informática IV

Ano Lectivo de 2016/2017

Eat Up

Daniel Teixeira Militão – A74557

Hugo Alves Carvalho – A74219

João Ismael Barros Reis – A75372

Abril, 2017

Data de Receção	
Responsável	
Avaliação	
Observações	

Eat Up

Daniel Teixeira Militão – A74557

Hugo Alves Carvalho – A74219

João Ismael Barros Reis – A75372

Abril, 2017

Resumo

O presente relatório descreve a primeira etapa do projeto “Eat Up” que surgiu devido à necessidade de implementar uma solução informática de recomendação e localização de restaurantes em Portugal, capaz de satisfazer as preferências do utilizador.

Esta primeira fase consiste na fundamentação do projeto, sendo iniciada pela contextualização, caso de estudo, motivação e objetivos. Nestes tópicos, para além de descrevermos o contexto onde o projeto está enquadrado e o seu modo de funcionamento, são expostas as principais motivações que nos levaram a implementar o projeto, bem como os objetivos a alcançar no término da construção da solução.

Em seguida, são apresentados a justificação e utilidade do sistema, onde explicamos em termos práticos como funciona o sistema e quais as utilidades que este oferece.

No estabelecimento da identidade do projeto, para além de referirmos o nome da aplicação, fazemos referência à categoria em que está inserida, bem como uma descrição dos seus serviços.

Com o tópico da identificação dos recursos necessários e da maqueta do sistema, descrevemos a forma como vamos obter dados para a aplicação e representamos a arquitetura do sistema idealizado, onde identificamos também os principais constituintes do sistema e a suas respetivas interações.

Seguidamente, elaboramos um conjunto de medidas de sucesso que futuramente nos permitem concluir se o sistema foi implementado de o êxito pretendido.

Por último, é descrito um planeamento das tarefas que constituem as três fases do projeto com base no tempo que prevemos ser necessário para as realizar, tendo este processo sido elaborada através de um diagrama de *Gantt*.

Área de Aplicação: Desenvolvimento de um sistema de recomendação e localização de restaurantes.

Palavras-Chave: Desenvolvimento de *Software*, Engenharia de *Software*, aplicação, restaurantes, localização, pesquisa, gestão de projeto, diagrama de *Gantt*.

Índice

1. Introdução	1
1.1. Contextualização	1
1.2. Apresentação do Caso de Estudo	1
1.3. Motivação e Objetivos	2
1.4. Justificação do sistema	4
1.5. Estabelecimento da identidade do projeto	4
1.6. Identificação dos recursos necessários	5
1.7. Maqueta do sistema	6
1.8. Definição de um conjunto de medidas de sucesso	7
1.9. Plano de desenvolvimento	8
2. Requisitos	10
3. Modelação em UML	14
3.1.1 Especificação de use case – Procurar produto – via texto	15
3.1.2 Especificação de use case – Selecionar produto	16
3.2. Diagramas de Sequência	16
3.3. Diagrama de Classe	18
4. Base de Dados	20
4.1. Modelo Conceptual	20
4.2. Modelo Lógico	26
4.2.1 Entidades	26
4.2.2 Relacionamentos	27
4.2.3 Atributos Multi-valor	27
4.2.4 Normalização	29
4.2.5 Elaboração e validação do esquema lógico da base de dados	30
5. <i>Mockups</i>	32
6. Implementação	38
6.1. Ferramentas utilizadas	38
6.1.1 Plataforma de desenvolvimento	38
6.1.2 Tecnologia de suporte	38
6.1.3 Visual Studio 2015	38
6.2. Estrutura inicial	39
6.3. Preenchimento da Base de Dados	39
6.4. Conexão Base de Dados	40
6.5. Funcionalidades	40
7. Conclusões e Trabalho Futuro	44

Anexos

I. Anexo 1 – Especificação de <i>Use Case</i>	49
II. Anexo 2 – Diagramas de Sequencia	50

Índice de Figuras

Figura 1 – Maqueta do sistema	6
Figura 2 – Diagrama de <i>Gantt</i>	8
Figura 3 - Diagrama de Use Case	14
Figura 4 - Especificação de use case – Procurar produto – via texto	15
Figura 5 - Especificação de use case – Selecionar produto	16
Figura 6 – Diagrama de Sequência – Procurar produto via texto	17
Figura 7 – Diagrama de Sequência – Selecionar Produto	18
Figura 8 – Diagrama de Classes	19
Figura 9 - Transformação de Relacionamento em Tabela	27
Figura 10 - Denúncia multi-valor em tabela	27
Figura 11 - Transformação atributo multi-valor 'Produto' em tabela	28
Figura 12 - Transformação atributo multi-valor 'Etiquetas_favoritas' em tabela	28
Figura 13 - Transformação do atributo multi-valor 'Historico_produto' em tabela	29
Figura 14 - Modelo Lógico	30
Figura 15 - <i>Mockup</i> Página Principal	32
Figura 16 - <i>Mockup</i> Menu	33
Figura 17 - <i>Mockup</i> Resultados de Pesquisa	34
Figura 18 - <i>Mockup</i> Adicionar Favoritos	35
Figura 19 - <i>Mockup</i> Ver Histórico	36
Figura 20 - <i>Mockup</i> Ver avaliações	37
Figura 21 - Estrutura básica	39
Figura 22 - Exemplo ligação a base de dados	40
Figura 23 - Erro de preenchimento inválido	40
Figura 24 - Exemplo de pesquisa	41
Figura 25 - Exemplo de redirecionamento	41
Figura 26 – Resultados	42
Figura 27 - Percurso a efetuar	43

Índice de Tabelas

Tabela 1 – Tabela das Entidades	20
Tabela 2 – Tabela dos Relacionamentos	20
Tabela 3 – Associação de Atributos e Entidades	24
Tabela 4 – Desenho do Diagrama ER	26

1. Introdução

1.1. Contextualização

Existe um número considerável de pessoas que regularmente necessitam de procurar, num determinado local, restaurantes para poderem almoçar ou jantar. Esta procura pode acontecer quando, por exemplo, uma pessoa está numa região em que não tem conhecimento dos restaurantes existentes na área e como tal precisa de encontrar o local mais adequado para a sua refeição. Por outro lado, por vezes é necessário procurar restaurantes com ementas que se adequem aos seus gostos pessoais e à sua capacidade financeira, bem como saber a distância e o percurso até chegar ao restaurante pretendido. Considerando estas situações bastante presentes no nosso quotidiano, surge o projeto *“Eat Up”*.

Este projeto consiste numa aplicação de recomendação e localização de restaurantes em Portugal, onde a qualquer momento o sistema é capaz de fornecer uma indicação precisa de um local onde podemos degustar algo do nosso agrado, assim como indicar o percurso a realizar até ele. Para além disso, o sistema tem em consideração as preferências do utilizador, a opinião de outros, o preço que este está disposto a pagar pela sua refeição entre outros filtros.

Os motivos para este enquadramento é, em primeiro ser o local onde nós nos encontramos e isto facilita o desenvolvimento do projeto dado a nossa rede de conhecimentos; em segundo, pela observação realizada, chegámos à conclusão que grande parte da informação que consideramos importante, nomeadamente ementas e preços, não se encontra disponível nos serviços existentes para restaurantes em Portugal porque ainda é um mercado não muito desenvolvido, sendo por isso uma excelente oportunidade para a nossa empresa.

Ainda por mais, a aplicação enquadra-se perfeitamente no território nacional uma vez que Portugal é um excelente ponto turístico.

1.2. Apresentação do Caso de Estudo

O projeto a ser desenvolvido divide-se em duas interfaces de utilização – uma para utilizadores e outra para proprietários de negócio – em distintos tipos de plataforma.

No que diz respeito interface de utilizador, será criada uma aplicação móvel assim como um *website* que recorrendo às várias componentes existentes em *smartphones* e *browsers* permitirá aos utilizadores:

- Encontrar restaurantes perto de si, com possível preferência de ementas e limite monetário;
- Avaliar restaurantes visitados;
- Partilhar restaurantes e/ou avaliações;
- Reportar restaurantes com informação incorreta;
- Adicionar restaurantes aos favoritos;
- Criar preferências.

Com todas estas funções, a aplicação terá uma rica e poderosa capacidade de satisfazer as necessidades e vontades do utilizador. As ementas, preços, enfim, todas as características relativas a restaurante são obtidas de uma base de dados online. Base de dados essa que será sustentada por proprietários de negócio que poderão:

- Adicionar restaurante;
- Editar informações relativas a restaurante;
- Adicionar ementas;
- Editar ementas;
- Apagar ementas.

No entanto, nenhuma destas funções está disponível na aplicação, mas sim numa página online exclusiva para a propósito.

1.3. Motivação e Objetivos

1.3.1. Motivação

O principal motivo que nos levou à realização deste projeto consistiu no facto de existirem dificuldades ao pesquisar restaurantes que satisfaçam, em tempo real, os gostos pessoais do utilizador.

Durante este processo de pesquisa, uma dificuldade que os utilizadores encontram é a falta de informação sobre as ementas existentes nos restaurantes, assim como do seu respetivo preço. Deste modo, quando alguém deseja procurar um restaurante numa determinada área, as informações que encontra são simplesmente a morada e, em alguns casos, a especialidade gastronómica do restaurante, ou seja, informações importantes como diferentes ementas, opiniões de outros utilizadores, percurso até ao restaurante, ou são inexistentes, ou têm de ser procuradas de outro modo. Assim, a pesquisa torna-se difícil e faz com que o utilizador tenha que perder muito tempo até encontrar algo do seu agrado.

O facto de um utilizador poder realizar uma pesquisa sem desperdiçar muito do seu tempo, é uma das nossas motivações para a realização deste projeto. Para além disso, na

maioria das situações em que se recorre a uma pesquisa na internet ou algo semelhante, nem sempre se encontra informação verídica.

O facto dos atuais competidores no mercado serem uma fonte constante de controvérsia devido às suas práticas anti consumidor e abusadores de pequenos negócios é então uma das motivações. Queremos proporcionar um produto de confiança para ambos consumidores bem como para os comerciantes.

Pela observação realizada chegámos à conclusão que em serviços concorrentes a informação mais importante a ver da nossa equipa, nomeadamente ementas e preços, não se encontra disponível para restaurantes em Portugal, isto porque ainda é um mercado não muito desenvolvido, sendo por isso uma excelente oportunidade para a nossa empresa.

Esta informação falsa pode nem ser da responsabilidade dos restaurantes e assim estes estão também a ser prejudicados, uma vez que o utilizador fica com uma má imagem do estabelecimento. Assim, a oportunidade de restaurantes poderem adicionar e partilhar as suas ementas, e de os utilizadores poderem avaliar e comentar a sua experiência gastronómica, representa outra das nossas principais motivações.

1.3.2. Objetivos

O principal objetivo da solução informática apresentada é disponibilizar uma interface intuitiva e de fácil utilização, que simplifique o processo de pesquisa de restaurantes de acordo com as preferências do utilizador.

Com a possibilidade de um utilizador pesquisar um restaurante através de texto ou de voz, indicando as suas preferências pessoais, pretendemos tornar mais simples e aumentar a rapidez do processo de pesquisa.

Além disso, a localização geográfica disponibilizada à aplicação, tem como objetivo possibilitar ao utilizador programar o seu percurso e até mesmo evitar trajetos que não sejam do seu agrado.

O facto de todos os utilizadores poderem avaliar, comentar e reportar um determinado restaurante, leva a que as informações presentes na aplicação sejam as mais verídicas possíveis. Para além disso, sempre que um utilizador estiver indeciso entre vários restaurantes, pode ter em consideração as opiniões dos outros utilizadores, bem como consultar uma lista ordenada de acordo com o índice de avaliação de cada restaurante.

Outro objetivo também considerado é a possibilidade de um utilizador poder partilhar a sua experiência gastronómica através de fotos ou texto, em redes sociais como *Facebook*, *Google+*, *Twitter* e *Instagram*. Este ponto, para além de permitir ao utilizador registar e guardar momentos da sua experiência, é também do agrado dos restaurantes, uma vez que permite a divulgação do estabelecimento.

Assim, esta solução apresentada, é também bastante útil para os restaurantes, uma vez que facilita o processo de divulgação do estabelecimento, bem como das suas ementas e preços. O sistema de avaliação por parte dos utilizadores, apresenta também uma vantagem para os

restaurantes uma vez que se um determinado restaurante tem boa reputação, atrairá mais facilmente potenciais clientes. Deste modo, cada restaurante pode registrar a sua morada, ementas e respetivos preços e que, após serem validados pela empresa, são adicionados ao sistema, ficando assim disponíveis a todos os utilizadores.

1.4. Justificação do sistema

Após realizarmos uma análise e recolha de informação do mercado das aplicações correspondentes a sistemas de recomendação e localização de restaurantes, deparamo-nos com algumas lacunas que limitam a experiência do utilizador. Por exemplo, na maioria das aplicações disponíveis, apenas é possível localizar restaurantes, não existindo qualquer indicação das ementas e dos preços que estes disponibilizam.

Outra situação que encontramos é que grande parte das aplicações semelhantes não possui uma lista ordenada de acordo com as avaliações dos utilizadores sobre cada restaurante. Esta ação é, no nosso entender, fundamental não só para permitir uma melhor escolha por parte do utilizador como também para promover os restaurantes com melhor reputação. Para além disso, a possibilidade de um utilizador poder procurar um restaurante de acordo com as suas preferências através de texto e sobretudo através da voz, bem como a utilização de *GPS* para obter a localização e direção para os restaurantes é algo incomum entre os nossos competidores.

Do ponto de vista dos restaurantes, foram identificadas falhas na dificuldade que estes têm na divulgação das suas ementas e preços, bem como da sua localização e opinião dos seus clientes.

Com base nestas lacunas encontradas, a nossa empresa desenvolveu uma nova aplicação, em que todas estas situações são resolvidas, oferecendo assim um sistema simples e interativo, capaz de suportar as preferências do utilizador. Por outro lado, qualquer restaurante tem acesso a um website de uso simplificado, onde pode inscrever-se no sistema e fazer o registo dos seus serviços, sendo assim possível uma divulgação verídica dos dados.

1.5. Estabelecimento da identidade do projeto

Nome: “*Eat Up*”

Categoria: “Alimentação e bebida”

Descrição: A *Eat Up* é uma aplicação que o vai ajudar quando a fome começa a apertar; apenas utilize o seu microfone, diga o que deseja comer e rapidamente lhe será dito onde e como chegar a um local indicado de acordo com as suas preferências. Não volte a aventurar-se no desconhecido sem saber o que lhe está à espera: o nosso sistema de avaliação permite também consultar a opinião do público geral de um determinado lugar.

Características:

- Reconhecimento de texto e voz;
- Mapeamento de diversos restaurantes;
- Navegação GPS;
- Avaliações dos restaurantes;
- Partilhas em redes sociais;

1.6. Identificação dos recursos necessários

Uma parte fulcral do projeto será de que maneira se irá obter recursos e como serão tratados e validados. Inicialmente iremos utilizar serviços existentes para recolha de dados. Estes são o *Yelp*, *TripAdvisor* e *Foursquare*. Deles pretendemos extrair uma lista de restaurantes de todo o país, em particular, a sua localização, contactos e horários. A partir desta lista iremos, numa fase inicial, contactar os mais conceituados com o intuito de fazer parceria com eles.

Pela observação realizada por nós chegámos à conclusão que grande parte da informação que consideramos importante, nomeadamente ementas e preços, não se encontra disponível nos serviços existentes para restaurantes em Portugal porque ainda é um mercado não muito desenvolvido, sendo por isso uma excelente oportunidade para a nossa empresa.

É por isso que a aplicação a ser criada terá uma especial interação com os proprietários de negócio, ou seja, qualquer responsável por um negócio de restauração terá a oportunidade de utilizar a aplicação para publicitar o seu restaurante.

Não obstante, numa fase inicial e com vista à publicidade e arranque com sucesso da aplicação, será necessário criar parcerias com os principais restaurantes das principais regiões do país – Lisboa, Porto e Braga. Tratar-se-á de um importante impulso porque ninguém iria publicitar o seu restaurante onde não existe atividade, assim como ninguém usaria uma aplicação que não cumprisse a sua função. Desta maneira, a aplicação ganha fama e os restantes restaurantes também sentirão a necessidade de entrar no serviço. Estes podem fazer solicitação para colocar o seu restaurante no sistema comprovando previamente que realmente representam o restaurante.

Os utilizadores não poderão adicionar restaurantes ao sistema, poderão apenas realizar avaliações e reportar informação incorreta.

Caso a página seja reportada demasiadas vezes pelos consumidores o dono irá ter que acarretar com as consequências. Estas poderão ser desligar temporariamente a página e em caso extremo expulsão do serviço.

A aplicação servirá exclusivamente para a procura de restaurantes. Consequentemente, os proprietários terão de ter um local onde terão possibilidade de inscreverem e gerirem as suas ementas, e a solução passará por um pequeno serviço online.

Ainda relativamente à aplicação, iremos também recorrer *API's* da *Google+*, *Facebook* e *Twitter* para login, de modo a preservar preferidos, preferências pessoais e para partilhas.

Para além disto, várias ferramentas irão ser usadas ao longo deste projeto, estas são:

- *Microsoft Project* para o planeamento do projeto;

- *Microsoft Word* para escrita dos relatórios;
- *Microsoft Power Point* para realização das apresentações;
- *Visual Paradigm* para modelação do sistema;
- *Microsoft SQL Server* para criação da base dados;
- *Microsoft Visual Studio* irá ser o nosso IDE;
- *Microsoft .NET C#* para escrever a nossa aplicação;
- *Microsoft Speech Recognition* para reconhecimento daquilo que o utilizador disser à aplicação, em concreto, palavras chave sobre a comida queira comer;
- *Bing Maps* que será utilizado para mostrar num mapa a localização do restaurante à escolha do utilizador e para planear a rota até esse local.

1.7. Maqueta do sistema

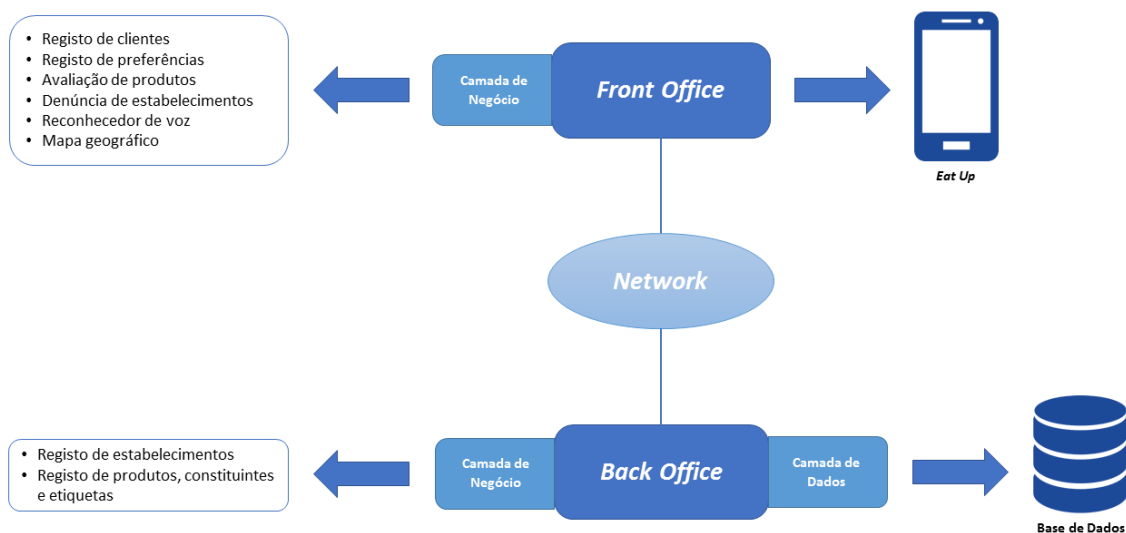


Figura 1 – Maqueta do sistema

O projeto divide-se em três camadas – utilizador, negócio e proprietário – onde será possível trabalhar-se independentemente de modo a obter o produto final.

Existem uma interface de utilizador e outra de proprietário, ambas que enviarão pedidos ao servidor de modo a obter ou comunicar novas informações, seja para localizar o restaurante mais próximo ou para registar um novo restaurante. Ainda nesta camada estará presente, para o utilizador, as funções de reconhecimento de voz e localização GPS.

Assim que os pedidos chegam ao servidor (camada de negócio) serão tratados de modo a responder às suas necessidades, que podem eventualmente ter de aceder à camada de dados caso surja a necessidade de obter dados. Nesta camada terá de existir forte segurança de modo a evitar problemas de acessos indesejados.

Finalmente, a camada de dados irá conter os diferentes dados relativos ao projeto, como os restaurantes e derivadas características, utilizadores e informação relevante, que terá também possuirá medidas de segurança como a criação de diferentes vistas para o sistema.

1.8. Definição de um conjunto de medidas de sucesso

Na fase de fundamentação do projeto, é crucial elaborar um conjunto de medidas que mais tarde possam indicar se este foi implementado com o êxito pretendido. Assim, neste tópico apresentamos uma lista com os principais pontos que consideramos fundamentais para obter sucesso com este projeto.

- Em primeiro lugar, é fundamental que todas as etapas do projeto cumpram os prazos estipulados. Esta medida, para além de nos dar indicativo que o projeto esteja realmente a avançar, permite também uma melhor gestão de tempo dedicado ao projeto por cada um dos elementos da equipa de trabalho.
- Para uma implementação correta do sistema, todos os requisitos identificados devem estar presentes na aplicação final implementada. Assim, podemos considerar que todos os objetivos propostos foram cumpridos do êxito.
- Com o objetivo de verificar que sistema é devidamente escalável, foi considerado que uma das medidas de sucesso seria ter, no mínimo, um número total de 200 utilizadores no primeiro mês do aplicativo. Por outro lado, é essencial também que pelos menos 10 empresas estejam certificadas no sistema.
- De modo a verificarmos que as variadas funcionalidades do aplicativo estão a ser usadas pelos utilizadores, o sistema deve ter um número mínimo de 10 comentários e 10 partilhas em redes sociais durante o primeiro mês.
- Durante o primeiro ano do sistema, é fundamental que todos os números anteriormente descritos para o primeiro mês, sejam superiores nos meses seguintes.

1.9. Plano de desenvolvimento

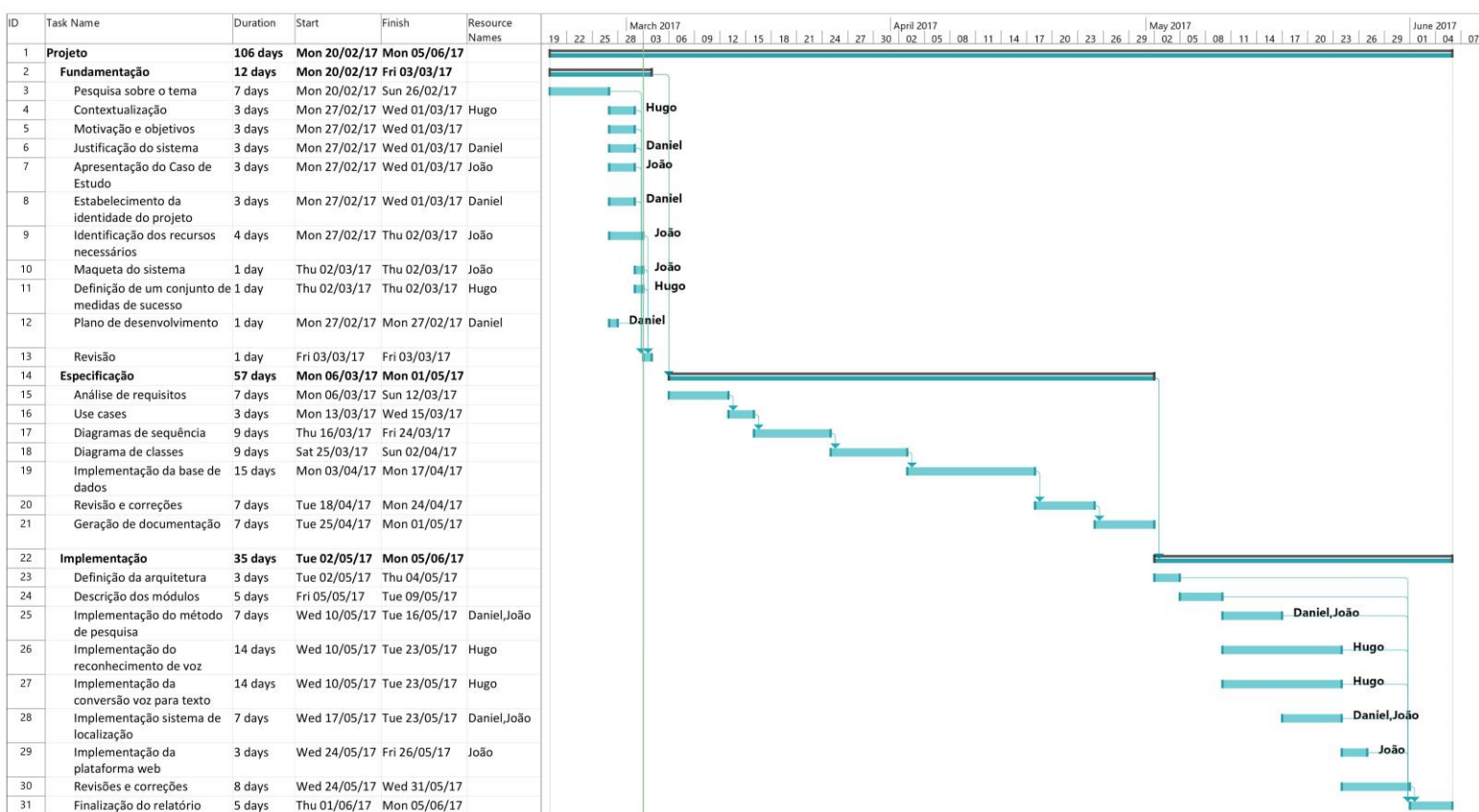


Figura 2 – Diagrama de Gantt

A primeira fase do projeto passou pela fundamentação do tema. Inicialmente, foi efetuada uma intensiva pesquisa acerca do tema com o objetivo de recolher informação que auxiliasse o grupo a entender o atual estado do mercado e aquilo que o consumidor esperaria da nossa aplicação. Para além disso, este estudo também nos permitiu perceber em que medida é que o projeto iria facilitar os consumidores na escolha de restaurante.

Partindo das informações recolhidas, dividimos grande parte do relatório entre os membros do grupo sendo apenas a secção “Motivação e os Objetivos” realizada por todos os elementos. Após a realização das tarefas correspondentes a esta fase do projeto, foi feita uma revisão por todos os membros do grupo e foram planeadas e delineadas as restantes fases que irão compor a realização do projeto.

Relativamente a cada tarefa, foi alocado para cada uma um período de tempo que achamos adequado para a realização da mesma, não sendo possível prever com exatidão o tempo de desenvolvimento de cada uma delas.

De notar que na segunda parte do projeto o grupo prevê que seja necessário um trabalho simultâneo de todos os elementos pois achamos que devemos terminar cada uma das fases antes de prosseguir para a próxima.

Por último, é de realçar que a divisão das tarefas por elementos do grupo foi feita através de uma discussão em equipa tendo em conta os pontos fortes de cada membro. Contudo, esta distribuição inicial de cargos é suscetível a alterações desde que estas sejam devidamente justificadas.

2. Requisitos

A pesquisa e análise de requisitos trata-se de uma importante e essencial fase da especificação do projeto dada a sua função de descrever o funcionamento de um sistema. É fulcral existir uma listagem das pretendidas capacidades do *software* que reflitam ambas as ideias dos desenvolvedores e clientes. Para tal efetuou-se vários estudos na área de restauração e as suas deficiências, bem como entrevistas a proprietários de negócio para saber o que mais se pretende de uma aplicação do género.

Nas seguintes secções fazer-se-á a apresentação dos requisitos de utilizador bem como os do sistema, que se dividem em funcionais e não funcionais.

2.1 Requisitos de utilizador

Aborda-se neste capítulo os requisitos de utilizador, no entanto é importante destacar dois aspetos que permitem uma compreensão básica do sistema:

1. Na aplicação existem dois tipos de utilizador – clientes e proprietários de negócio – ambos com distintas funções.
2. É obrigatória a verificação de restaurantes inseridos, que passa por um processo de validação.

No que diz respeito aos clientes, estes terão de ter as seguintes funções disponíveis:

3. Procurar um estabelecimento que ofereça um determinado produto alimentar, introduzindo essa mesma preferência por teclado.
4. Procurar um estabelecimento que ofereça um determinado produto alimentar, introduzindo essa mesma preferência por voz.
5. Procurar um estabelecimento pelas etiquetas favoritas;
6. Dado um produto consumido num determinado estabelecimento, fazer uma avaliação.
7. Adicionar etiquetas aos favoritos;
8. Partilhar um produto ou avaliação pessoal em redes sociais.

Consequentemente, no que concerne aos proprietários de negócio:

9. Adicionar produto, com os exatos constituintes e preços;
10. Editar produtos previamente criadas;
11. Apagar produtos previamente criadas.

2.2 Requisitos do Sistema

Os requisitos do sistema classificam-se em requisitos funcionais - que corresponde ao que o sistema deve fornecer, responder ou comportar – e requisitos não funcionais – que dizem respeito a limitações nos serviços ou funções oferecidas pelo sistema.

2.2.1 Requisitos funcionais

1.
 - a. De modo a manter a distinção entre cliente e proprietário de negócio implica obrigatoriamente a criação de diferentes perfis para os diferentes tipos de utilizadores de modo a separar as diversas funcionalidades;
 - b. Uma vez que os clientes alvo são pessoas com forte presença em redes sociais, o registo será feito usando *API's* do Facebook e Twitter para tal efeito; De lá serão obtidos dados essenciais para o registo – e-mail e nome – para ser colocado na base de dados.
 - c. Por outro lado os proprietários de negócio registam o seu negócio que requerem obrigatoriamente os seguintes dados:
 - i. Certidão permanente do negócio;
 - ii. Localização exata do negócio, recorrendo ao Bing Mapas para tal efeito;
 - iii. Horários;
 - iv. Temática;
 - v. Contactos (E-mail, telefone);
 - d. Ambos os tipos de utilizadores terão de estar autenticados para aceder a funções;
 - e. Permissões serão geridas pelo sistema gestor de base de dados (SGBD) que será responsável pela atribuição de funções disponíveis.
2.
 - a. Após o registo feito em 1c) a administração terá de comprovar os dados submetidos pelos proprietários de negócio.
 - b. Se os dados tiverem sido validados com sucesso, será enviado com *e-mail* com resposta afirmativa e password de acesso.
 - c. Se os dados forem inválidos, será enviado um *e-mail* com resposta negativa.
3.
 - a. Após o input do utilizador com a preferência em modo textual, p.e. “bitoque, cheesecake”, a aplicação recorre à função GPS (ou localização por IP caso esta esteja indisponível) para obter a localização. Terá também a capacidade de reconhecer padrões de limitações à sua refeição, isto é, que contenha ou não determinado(s) constituinte(s), p.e. francesinha **sem** linguiça ou francesinha **sem** ovo **com** batata.
Desta maneira o padrão de escrita terá de ser:

- i. $\sum_1^n([\text{Etiqueta}] + \sum_0^n \text{com} [\text{Constituinte}] + \sum_0^n \text{sem}[\text{Constituinte}])$
 - b. A partir daí obtém uma lista dos estabelecimentos mais próximos, ordenados por pontuação, com a preferência alimentar do utilizador;
 - c. Se selecionado algum estabelecimentos, este indicará a distancia e o tempo a que se encontra bem como possível navegação;
- 4.
- a. Este requerimento funciona da mesma maneira que 4), com a importante diferença de poder-se usar a voz para o fazer. Para tal, recorre-se a um reconhecedor de voz da *Microsoft* que converterá o ficheiro de voz em texto.
- 5.
- a. Consultando o menu das etiquetas favoritas, seleciona-se a pretendida e obtém-se os resultados analogamente a 4. b) e c).
- 6.
- a. Visitando o histórico de produtos consumidos, escolhe-se o pretendido.
 - b. A avaliação da(s) produto(s) terá de ser numa escala de 0 a 5.
 - c. A avaliação poderá vir acompanhada com uma crítica em texto.
- 7.
- a. Acendendo à secção dos favoritos terá uma caixa de texto que permitirá adicionar preferências, seguindo o mesmo padrão que em 4.a)
- 8.
- a. Visitando o histórico de produtos consumidos ou avaliações é escolhida a opção de partilhar nas redes sociais.
 - b. A partilha será possível fazer em *Facebook* ou *Twitter*;
- 9.
- a. Entende-se por produto qualquer entrada de um menu regular, isto é, um prato, bebida, sobremesa, bebida, enfim, qualquer produto de especialidade vendido na área de restauração.
 - b. Para adicionar uma ementa é necessário preencher obrigatoriamente os seguintes campos:
 - i. Nome
 - ii. Constituintes
 - iii. Porção [Pequena/Média/Grande/(..)]
 - iv. Preço
 - v. Etiqueta

A etiqueta corresponde à categorização de uma ementa. Isto é, uma ementa que se denomine “Bitoque à moda Porto” terá a etiqueta “Bitoque”.
- 10.
- a. De modo a editar uma ementa é necessário existir alguma já criada anteriormente.

11.

- a. De modo a apagar uma ementa é necessário existir alguma já criada anteriormente.

2.2.2 Requisitos não funcionais

12. O dispositivo para utilizadores da aplicação deverá estar equipado com microfone e GPS.

3. Modelação em UML

Após concluída a fase dos requisitos, estes foram apresentados e acordados com o coordenador de projeto. De modo modelar o sistema e funcionamento da aplicação, recorremos aos vários diagramas UML para nos auxiliarem nesta tarefa.

3.1 Diagrama de Use Case

Uma vez que validados dos Requisitos de utilizador, estes servirão de base para a criação de use cases. Neste diagrama é apresentada a interação entre os três tipos de utilizador – cliente, proprietário e administrador com o sistema. No entanto, estes têm diferentes papéis neste sistema o que leva a que este lhe diferentes funções a diferentes utilizadores. De seguida encontra-se ilustrado as diferentes funcionalidades.

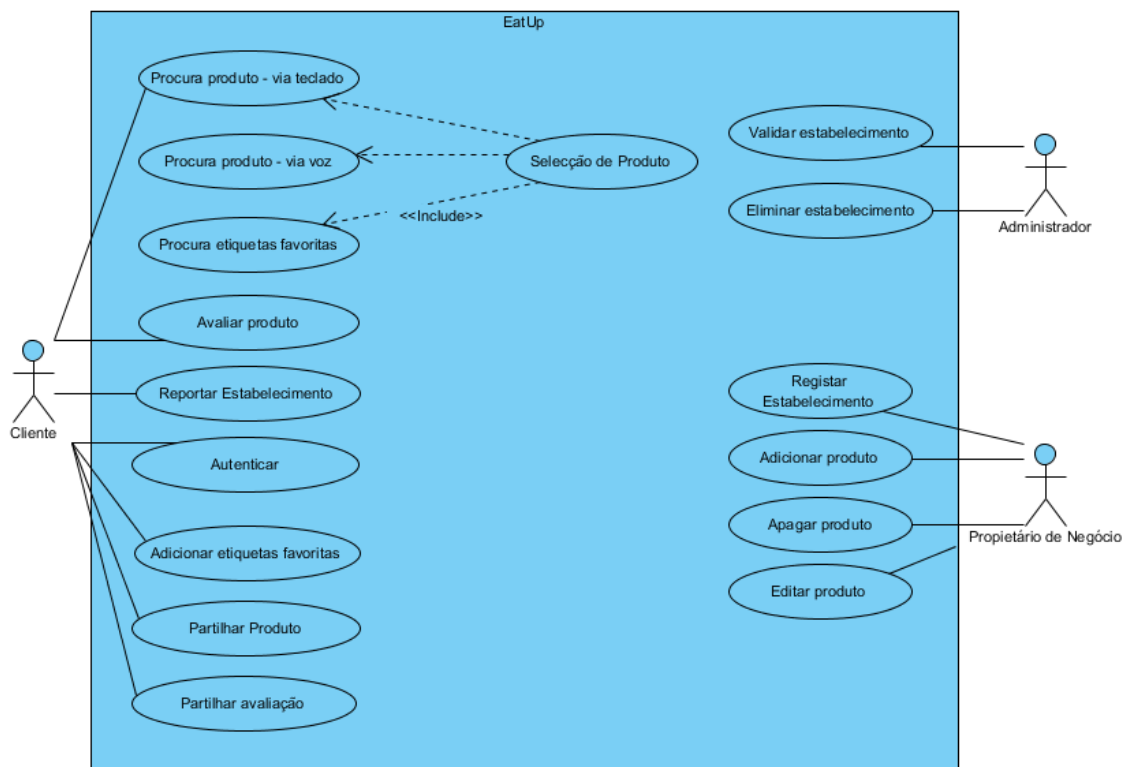


Figura 3 - Diagrama de Use Case

Após análise do diagrama é possível, verifica-se que está representado através de uma seta, uma relação de generalização entre atores – cliente, proprietário de negócio e administrador – e os use cases a eles limitados. Especial atenção ao use-case 'Seleção de Produto' que precisa anteriormente ter sido procurado um produto – seja por voz, texto ou etiquetas.

Cada Use Case/funcionalidade presente no diagrama anterior foi devidamente especificado. Essa especificação foi efetuada em formato tabular, em que se descreve sucintamente o Use Case, as pós e pré-condições, bem como o processo de interação entre o ator e o subsistema. De forma exemplificar o processo, serão de seguida apresentados uns exemplos desta especificação, bem como outros relevantes em anexo.

3.1.1 Especificação de use case – Procurar produto – via texto

Na imagem de seguida representada é possível observar a descrição em formato tabular do Use case “Procura de produto – via teclado”, que tem como ator um Cliente. É apresentada uma breve descrição do funcionamento bem como pré-condições (que nesta situação não existe) assim como pós condições, que se trata de uma lista de produtos.

Especificação de Use Case			
Name	Procurar produto - via teclado		
Brief Description	Permite a um cliente procurar um produto com as restrições desejadas.		
Preconditions	n/a		
Post-conditions	Apresenta lista de estabelecimentos com o produto pretendido.		
Flow of events		Actor Input	System Response
	1	Carrega na opção procurar por teclado.	
	2		Apresenta caixa de texto.
	3	Digita preferencias.	
	4	Seleciona raio de distancia pretendido.	
	5	Confirma.	
	6		Valida dados.
	7		Procura estabelecimentos com preferencias indicadas.
	8		Apresenta resultados obtidos com os produtos, estabelecimentos e as suas distancias.
Exceção 1 (passo 6) [Padrão de pesquisa errado]	1		Indica que input é inválido
	2		Volta a 2.
Exceção 2 (passo 7) [Não existem estabelecimentos no raio indicado]	1		Informa que não existe qualquer estabelecimento perto do utilizador
	2		Volta a 2.

Figura 4 - Especificação de use case – Procurar produto – via texto

De seguida é descrito o comportamento normal que demonstra como desencadeia o processo de forma sequencial, mostrando como agiria um ator e o sistema ao longo do seu funcionamento e iterações. Inicialmente o utilizador carrega na opção por teclado, e o sistema apresenta a caixa de texto. De seguida o utilizador digita as suas preferências e seleciona o raio de distância de preferido e confirma. Após validar os dados, o sistema procurar estabelecimentos que tenham os produtos pretendidos e mostra uma lista com os produtos.

Existem no entanto exceções que podem acontecer, como padrão de pesquisa inválido na caixa de texto no passo 3, ou então que não existem estabelecimentos que contenham o produto desejado, no passo 7.

3.1.2 Especificação de use case – Selecionar produto

Mais uma vez encontram-se na tabela abaixo representada o nome – Selecionar produto – descrição, pré e pós condições necessárias.

Como pré-condição é necessário que o cliente já tivesse procurado um restaurante por voz, texto ou etiquetas, como representado anteriormente.

A pós condição será que um produto será registado no histórico desse mesmo cliente.

Especificação de Use Case			
Name	Selecionar restaurantes.		
Brief Description	Permite ao cliente selecionar restaurantes da pesquisa.		
Preconditions	Procura efetuada.		
Post-conditions	produto registada no histórico.		
Flow of events		Actor Input	System Response
	1	Seleciona um produto	
	2		Calcula rota mais breve.
	3		Apresenta rota em mapa.
	4	Seleciona navegação.	
	5		Apresenta navegação GPS.
	6	Marca como visitado.	
	7		Guarda dados.
	8		Volta para menu principal.
Comportamento Alternativo 1 (4)			
[Dispositivo não tem GPS]	1		Informa que o dispositivo utilizado não tem GPS
	2		Passa para 6.

Figura 5 - Especificação de use case – Selecionar produto

Segue-se é mostrado o desencadeamento do *Use Case*: inicialmente o utilizador seleciona um dos produtos da lista, de seguida calcula a rota mais breve e apresenta-a no mapa. O utilizador seleciona navegação até o local, o sistema apresenta e o cliente marca como visitado. Finalmente, o sistema guarda os dados e volta para o menu principal.

3.2. Diagramas de Sequência

Com o objetivo de representar a informação descrita e detalhada sobre as funcionalidades do sistema, foram desenvolvidos diagramas de sequência, permitindo assim uma melhor compreensão da interação entre o utilizador e o sistema em cada um dos *Use Cases*.

Este tipo de diagrama está mais próximo do código que será desenvolvido e facilita o processo de transição para essa mesma fase de desenvolvimento.

De forma exemplificar o desenvolvimento deste tipo de modelo, serão de seguida apresentados alguns exemplos, bem como outros relevantes em anexo.

3.2.1. Exemplo de Diagrama de Sequência – Procurar Produto via texto

De forma a completar o exemplo fornecido anteriormente sobre a descrição do *Use Case* “Procurar produto via texto”, é apresentado em seguida o seu diagrama de sequência.

Neste diagrama é possível verificar que o ator corresponde ao cliente que vai usar a aplicação. Deste modo, para iniciar a procura, o cliente tem que escolher a opção de procurar um produto via teclado, na qual o sistema responde apresentando uma caixa de texto. Em seguida, este digita as suas preferências e o raio (em quilómetros) máximo que está disposto a procurar. Este processo dá origem à existência de dois ciclos uma vez que enquanto não existir nenhum estabelecimento no raio escolhido pelo cliente ou o padrão de pesquisa estiver errado, o sistema indica a exceção e apresenta de novo a caixa de texto para que o cliente possa voltar a procurar. Por último, quando o sistema consegue procurar de acordo com as escolhas do cliente, este apresenta os resultados obtidos com a lista de produtos, estabelecimentos e suas respectivas distâncias.

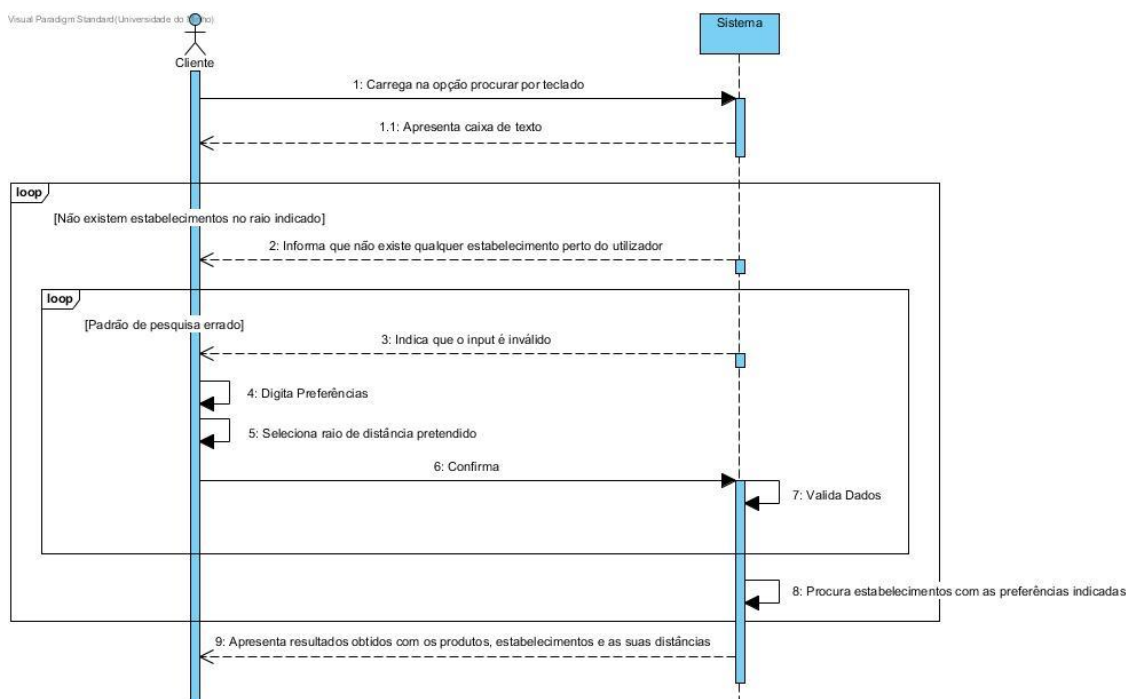


Figura 6 – Diagrama de Sequência – Procurar produto via texto

3.2.2. Exemplo de Diagrama de Sequência – Selecionar Produto

Em sequência com o exemplo de *Use Case* “Selecionar Produto” apresentado anteriormente, apresenta-se em seguida o diagrama de sequência que reflete o comportamento descrito na sua especificação.

Este processo de selecionar um produto surge após o cliente ter efetuado uma pesquisa, seja esta através de voz, texto ou verificando os seus favoritos. O cliente inicia este processo selecionando o produto que deseja encontrar, na qual o sistema efetua uma procura da rota mais breve até ao estabelecimento que tem esse produto, e apresenta o mapa ao cliente. Em seguida temos dois comportamentos que podem acontecer:

1. Se o cliente estiver a utilizar um dispositivo com GPS, este deve selecionar a opção de navegação, na qual o sistema apresenta a respetiva navegação GPS.

2. Se o cliente estiver a utilizar um dispositivo sem essa funcionalidade, o sistema apenas informa o cliente que o dispositivo não tem GPS.

Por ultimo, o cliente marca o produto como visitado, o sistema guarda os dados e volta ao menu principal da aplicação.

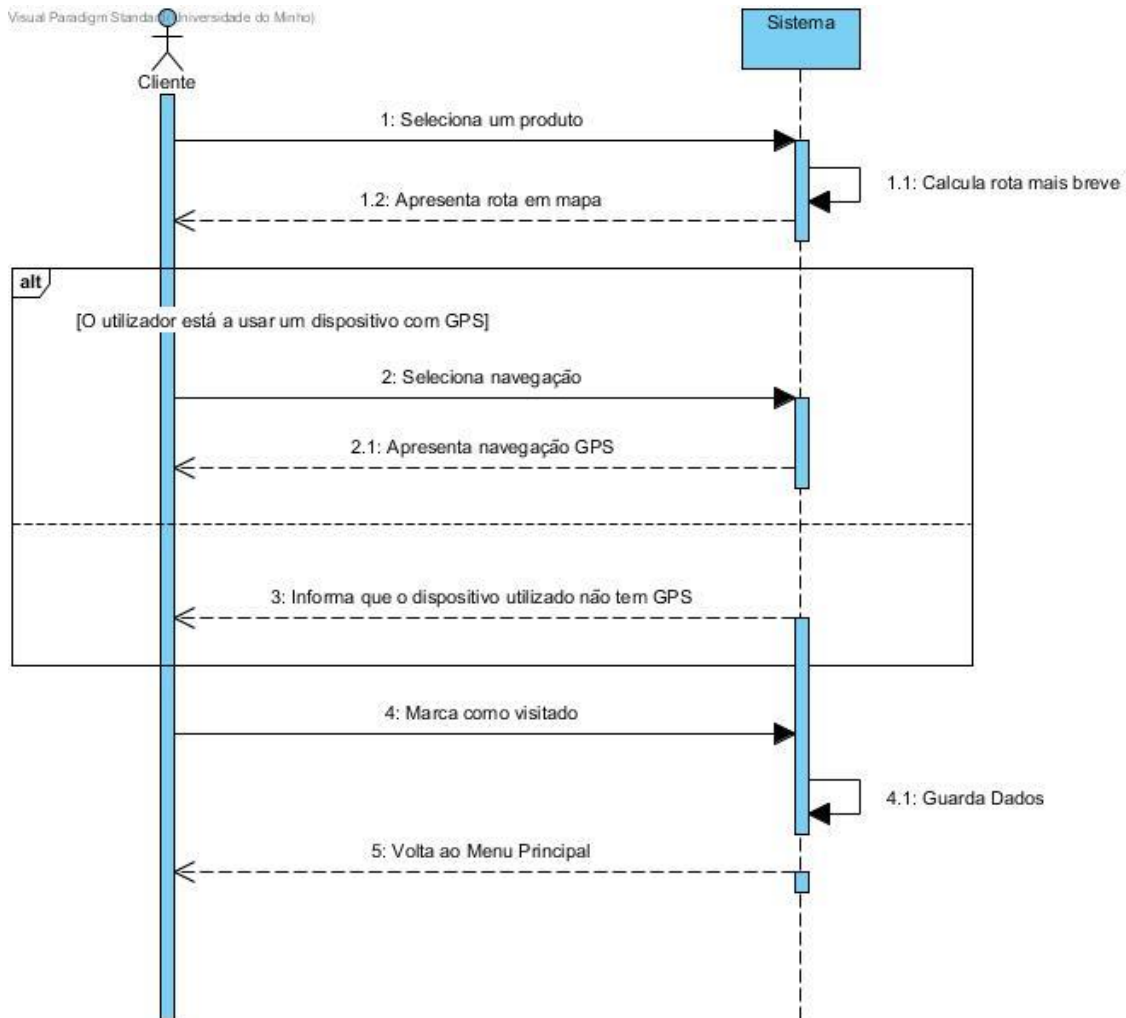


Figura 7 – Diagrama de Sequência – Selecionar Produto

3.3. Diagrama de Classe

É agora apresentado o diagrama de classes. Este diagrama apresenta todas as classes e os seus respetivos atributos e métodos. Também são especificadas as relações que as classes têm umas com as outras. Por fim, é especificado nas relações o tipo de associação (agregação ou composição).

Para a realização deste diagrama foram analisados os requisitos do sistema descritos na secção anterior.

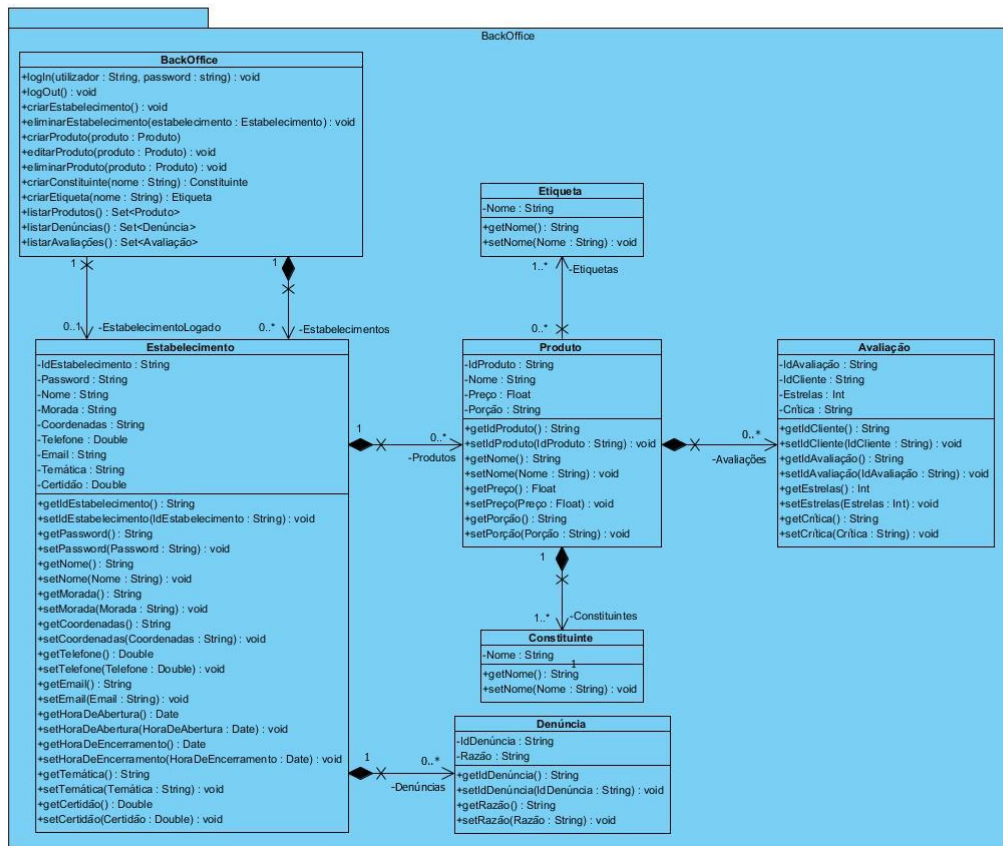
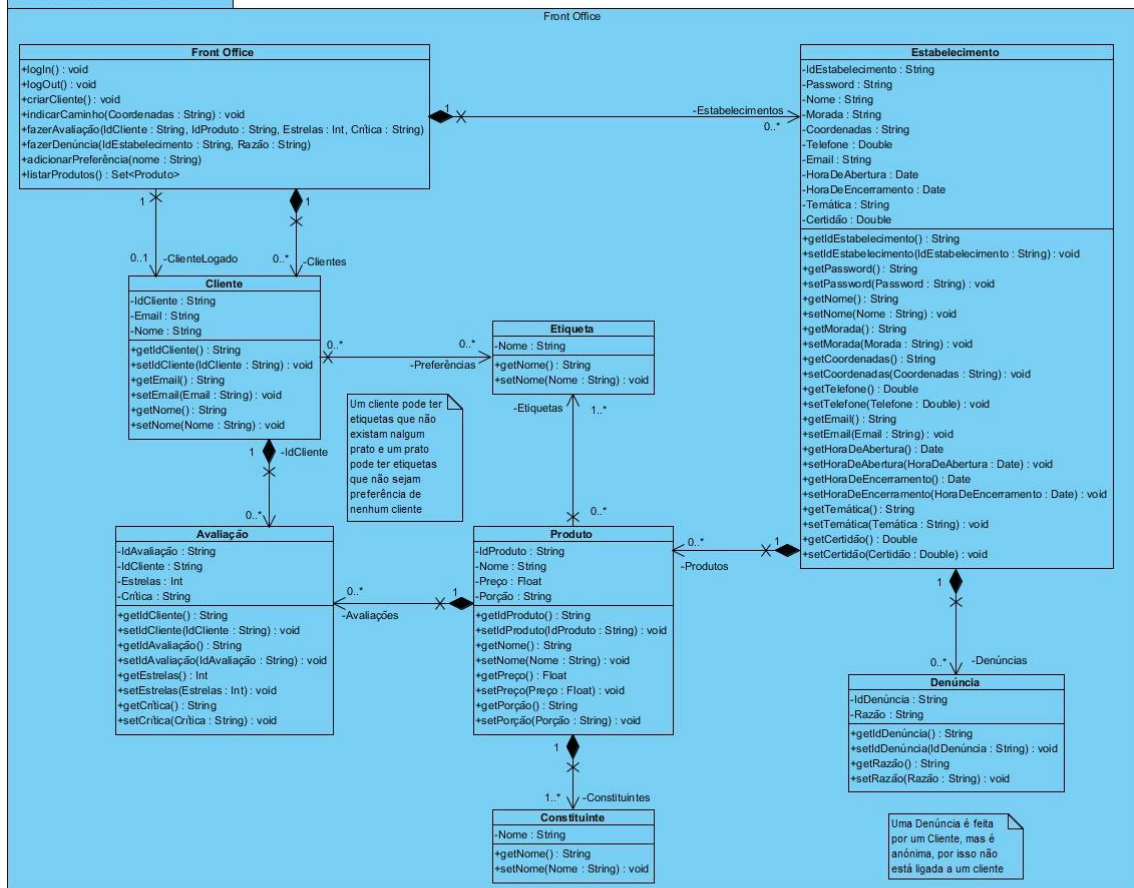


Figura 8 – Diagrama de Classes

4. Base de Dados

4.1. Modelo Conceptual

4.1.1. Identificar os tipos de entidades

De forma a conseguirmos identificar as diferentes entidades que o sistema engloba, foi necessário determinar quais os objetos que se enquadram nesta definição. Para isso, após a leitura e análise de requisitos, foram identificadas três entidades: cliente, avaliação e estabelecimento.

Nome da Entidade	Descrição	Sinónimos	Ocorrências
Cliente	Cliente que pretende usar a aplicação para encontrar um produto desejado	Utilizador, comprador	Um cliente pode procurar um produto através de texto ou voz
Avaliação	Classificação de um produto, realizada pelo cliente	Classificação, opinião	Qualquer cliente registado pode avaliar um produto
Estabelecimento	Estabelecimento que adiciona produtos para estejam visíveis na aplicação.	Negócio, restaurante.	Um estabelecimento pode adicionar o número de produtos que pretender

Tabela 1 – Tabela das Entidades

4.1.2. Identificar tipos de relacionamento

Após identificadas as entidades do nosso SBD, é necessário detetar todos os relacionamentos existentes entre as mesmas. A leitura e análise dos requisitos permite identificar estes relacionamentos que as entidades estabelecem entre si, bem como a sua respetiva cardinalidade.

Desta forma, apresentamos na seguinte tabela todos os relacionamentos entre entidades que foram identificados

Nome da Entidade	Multiplicidade	Relacionamento	Multiplicidade	Nome da Entidade
Cliente	1	Escreve	N	Avaliação

Tabela 2 – Tabela dos Relacionamentos

4.1.3. Identificar e associar atributos com os tipos de entidades e relacionamentos

De seguida, será abordada a associação entre a informação que conseguimos reter sobre as diferentes entidades. Para isso, foi necessário detetar todos os atributos necessários para representar cada entidade, bem como identificar o seu respetivo tipo.

4.1.3.1. Atributos simples/compostos

Após a leitura dos requisitos, chegou-se à conclusão que existem três atributos da entidade estabelecimento com a definição de atributo composto: produto, denuncia e contacto. No caso do produto, esta é composta por um número identificador, preço, porção, constituinte nome e etiquetas. Por outro lado, a denuncia é composta um número identificador e pela razão que levou à denuncia. Por fim, o contacto é composto por e-mail e telefone.

4.1.3.2. Atributos derivados

Após o levantamento de requisitos e identificação dos atributos, chegou-se à conclusão que não existem atributos derivados no nosso sistema.

4.1.3.3. Atributos multivalor

Da mesma análise referida nos pontos anteriores, é possível concluir que os atributos produto e denuncia da entidade estabelecimento, já anteriormente referidos como atributos compostos, são também atributos multivalor, uma vez que podem existir mais que uma ementa ou denuncia associada ao mesmo estabelecimento.

Por outro lado, os atributos etiquetas e constituinte, associados ao atributo composto produto, são também atributos multivalor pois podem ter mais do que uma ocorrência.

Do mesmo raciocínio anterior, os atributos etiquetas favoritas e histórico de produtos da entidade cliente são também um atributo multivalor.

4.1.3.4. Associação entre atributos e entidades

Nome da Entidade	Atributos	Descrição	Tipo e Tamanho	Nulo	M.V	Derivado	Composto
Cliente	ID_Utilizador	Número identificador do cliente	Valor inteiro positivo	Não	Não	Não	Não

	Nome	Nome do cliente	45 Caracteres variáveis	Não	Não	Não	Não
	E-mail	E-mail com que o cliente está registado	45 Caracteres variáveis	Não	Não	Não	Não
	Etiqueta_Favoritas	Etiquetas que o cliente adiciona como suas favoritas	20 Caracteres variáveis	Sim	Sim	Não	Não
	Histórico_produto	Identificadores dos produtos que o cliente pesquisou	Valor inteiro positivo	Sim	Sim	Não	Não
Avaliação	ID_Avaliação	Número que identifica a avaliação	Valor inteiro positivo	Não	Não	Não	Não
	Id_produto	Número que identifica o produto que está a ser avaliado	Valor inteiro positivo	Não	Não	Não	Não
	Estrelas	Número de estrelas com que avalia o produto	Valor inteiro positivo	Não	Não	Não	Não
	Crítica	Observações relativas à avaliação	Texto	Sim	Não	Não	Não
Estabelecimento	Id_estabelecimento	Número que identifica o estabelecimento	Valor inteiro positivo	Não	Não	Não	Não
	Nome	Nome do estabelecimento	45 Caracteres variáveis	Não	Não	Não	Não
	Password	Password de registo do estabelecimento	45 Caracteres variáveis	Não	Não	Não	Não
	Certidão	Identifica unicamente um negócio em Portugal	12 Dígitos	Não	Não	Não	Não
	Coordenadas	Coordenadas GPS para identificar o estabelecimento	45 Caracteres variáveis	Não	Não	Não	Não

	Temática	Temática em que o estabelecimento está inserido	45 Caracteres variáveis	Não	Não	Não	Não
	Hora_de_Abertura	Hora de abertura do estabelecimento	Hora	Não	Não	Não	Não
	Hora_de_Encerramento	Hora de encerramento do estabelecimento	Hora	Não	Não	Não	Não
	Observações	Observações relativas ao comboio	Texto	Sim	Não	Não	Não
Contacto							
	E-mail	E-mail com que o estabelecimento está registado	45 Caracteres variáveis	Não	Não	Não	Não
	Telefone	Contacto telefónico do estabelecimento	Valor inteiro positivo	Não	Não	Não	Não
Produto							
	Número	Número que identifica o produto	Valor inteiro positivo	Não	Não	Não	Não
	Nome	Nome do produto	45 Caracteres variáveis	Não	Não	Não	Não
	Preço	Preço do produto	Valor decimal positivo	Não	Não	Não	Não
	Porção	Quantidade do produto	String	Não	Não	Não	Não
	Etiquetas	Nome do produto	20 Caracteres variáveis	Não	Não	Não	Não
	Constituinte	Nome do constituinte	45 Caracteres variáveis	Não	Não	Não	Não
Denuncia							
	Número	Número que identifica a denúncia	Valor inteiro positivo	Não	Não	Não	Não

		Razão	Observações relativas à denúncia	Texto	Sim	Não	Não	Não

Tabela 3 – Associação de Atributos e Entidades

4.1.4. Determinar domínio dos atributos

Seguidamente, são descritos os domínios dos atributos referentes às diferentes entidades. O domínio consiste num conjunto de valores que pertencem a um determinado tipo, e que pode ser atribuído a cada atributo.

Entidade: Cliente

- ID_Utilizador: Número identificador do cliente. É um número inteiro positivo;
- Nome: Primeiro e último nome do cliente. É uma *string* com 45 caracteres variáveis;
- E-mail: E-mail com que o cliente está registado. É uma *string* com 45 caracteres variáveis.
- Etiqueta_favoritas: Etiqueta que o cliente marca como favorita. É uma *string* com 45 caracteres variáveis;
- Historico_produtos: Produtos que o cliente pesquisou. Cada elemento é um numero inteiro positivo;

Entidade: Avaliação

- ID_Avaliacao: Número que identifica a avaliação. É representado por um número inteiro positivo;
- Id_produto: Número que identifica o produto que está a ser avaliado. É representado por um número inteiro positivo;
- Estrelas: Número de estrelas com que o cliente avalia o produto. É representado por um número inteiro positivo;
- Crítica: Texto correspondente a observações relativas ao produto que está foi avaliado.

Entidade: Estabelecimento

- Id_estabelecimento: Número que identifica o comboio. É representado por um número inteiro positivo;
- Nome: Texto correspondente a observações relativas ao comboio.
- Password: Password corresponde ao registo do estabelecimento. É caracterizado por uma *string* com 45 caracteres variáveis;
- Certidão: Número que identifica um negócio nacionalmente. É caracterizado por doze dígitos.

- Coordenadas: coordenadas GPS onde o estabelecimento está localizado. É caracterizado por uma *string* com 45 caracteres variáveis;
- Temática: Temática em que o estabelecimento está inserido. É caracterizado por uma *string* com 45 caracteres variáveis;
- Hora_de_Abertura: Hora de abertura do estabelecimento. Corresponde à hora, minuto e segundo marcados para a abertura do mesmo;
- Hora_de_Encerramento: Hora de encerramento do estabelecimento. Corresponde à hora, minuto e segundo marcados para o encerramento do mesmo;
- Contacto: Contactos do estabelecimento, composto por:
 - E-mail: É uma *string* com 45 caracteres variáveis.
 - Telefone: Número inteiro positivo que identifica o contacto telefónico do estabelecimento;
- Produto: Identificador dos produtos do estabelecimento, composto por:
 - Número: Número inteiro positivo que identifica o produto
 - Nome: É uma *string* com 45 caracteres variáveis.
 - Preço: Preço de comercialização do produto. É um número decimal positivo.
 - Constituinte: Identificadores dos constituintes do produto. É um número inteiro positivo.
 - Porção: Quantidade do produto. É um inteiro decimal positivo.
 - Etiquetas: Etiquetas em que o produto está inserido. Corresponde a *strings* com 20 caracteres variáveis.
- Denúncia: Identificador das denúncias do estabelecimento, composto por:
 - Número: Número inteiro positivo que identifica a denúncia.
 - Razão: Texto correspondente a observações relativas ao produto que está foi denunciado.

4.1.5. Determinar chaves primárias, candidatas e alternativas

De forma a identificar unicamente cada uma das ocorrências, foi necessário determinar os atributos chave.

Deste modo, uma vez de cada uma das entidades contém um identificador, este atributo foi considerado como chave primária. Assim, é possível identificar cada um dos registos de uma entidade através do seu identificador, uma vez que este é único.

De seguida apresentamos as chaves primárias que consideramos para as entidades identificadas anteriormente:

- Cliente: ID_Utilizador
- Avaliação: ID_Avaliação
- Estabelecimento: ID_Estabelecimento

4.1.6. Desenho do diagrama ER

Apresentamos, de seguida, o desenho do diagrama E-R (Entidade-Relacionamento) de forma a representar conceptualmente as relações entre as entidades da base de dados.

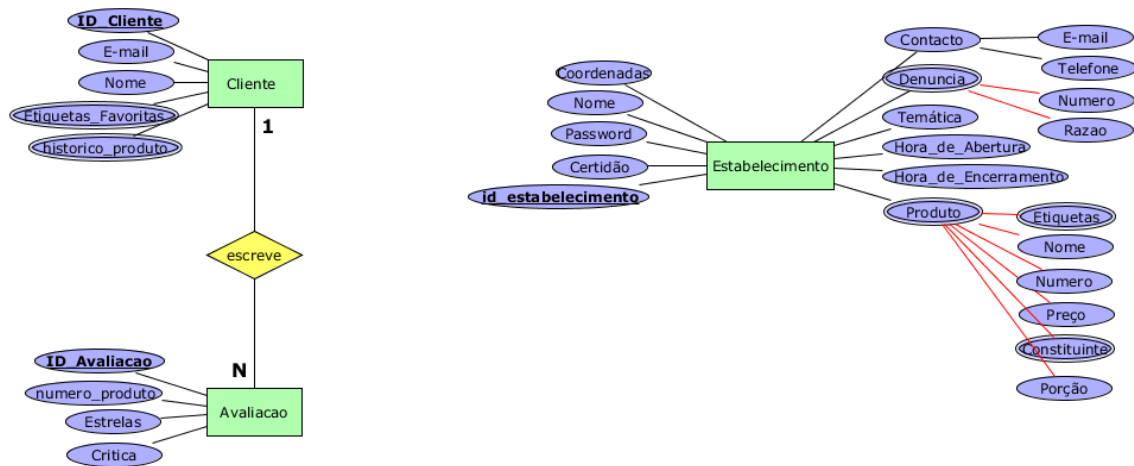


Tabela 4 – Desenho do Diagrama ER

4.1.7. Revisão do modelo de dados com o utilizador

Após a realização do modelo conceptual, este foi revisto pelo utilizador. Nenhum problema foi detetado e por isso o modelo de dados foi aceite.

4.2. Modelo Lógico

Nesta secção irá ser demonstrado como iremos transformar o modelo conceptual construído no modelo lógico, seguindo várias etapas até ao modelo final.

4.2.1 Entidades

Sendo que uma entidade forte trata-se duma entidade em que a sua existência não depende de outra, todas as três entidades presentes no modelo conceptual – Cliente, Avaliação e Restaurante – estão de acordo com esta norma e irão dar origem a tabelas.

4.2.2 Relacionamentos

No modelo conceitual está representada uma relação 1:N. Este relacionamento foi mantido no modelo lógico através da inclusão da chave estrangeira `ID_Utilizador` na tabela de `Avaliacao`, assim como abaixo demonstrado.

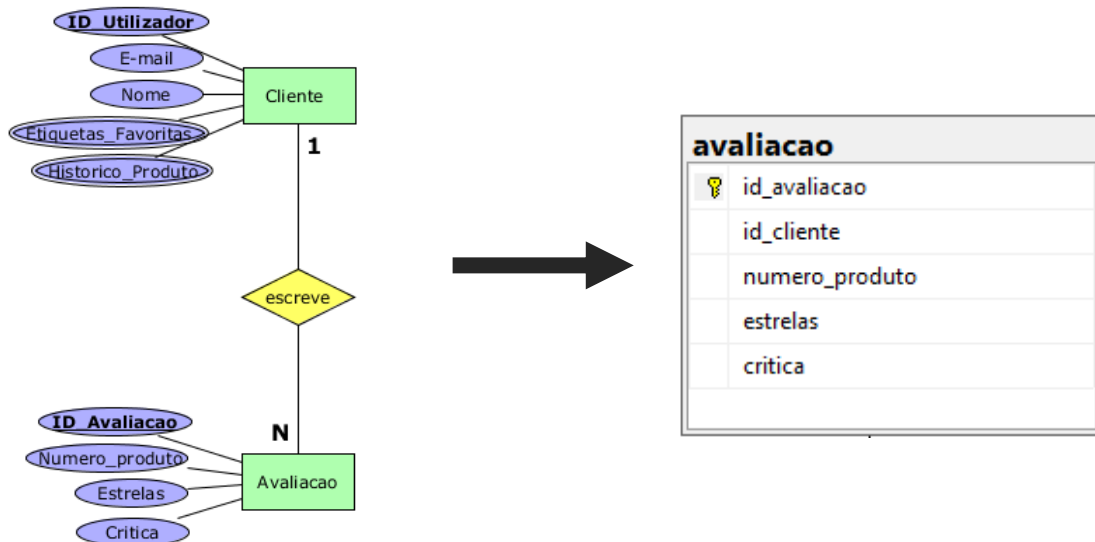


Figura 9 - Transformação de Relacionamento em Tabela

4.2.3 Atributos Multi-valor

No modelo conceitual estão representados vários atributos multi-valor, inclusive atributos compostos multi-valor. De modo a representar no modelo lógico, foram criadas novas tabelas para esses atributos.

Começando pelo atributo 'Estabelecimento', esta contém dois valores multi-valor, sendo eles Denuncia e Produto sendo o último composto. Foram criadas duas novas tabelas, usando o `id_estabelecimento` como chave estrangeira.

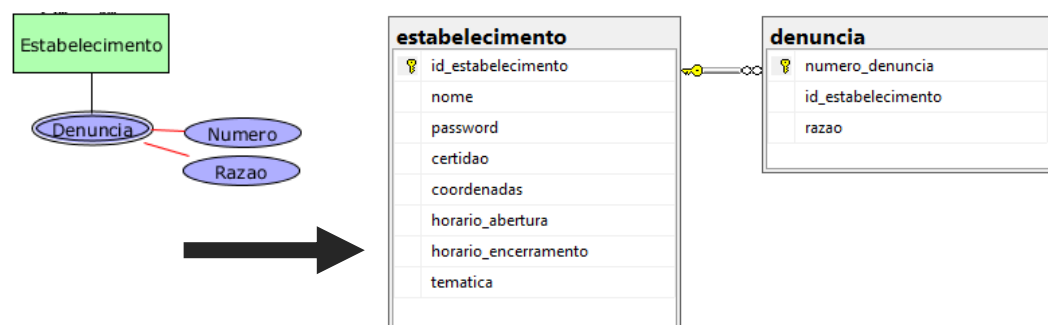


Figura 10 - Denuncia multi-valor em tabela

A atributo multi-valor 'Produto' tem em si mais dois atributos multi-valor 'Constituinte' e 'Etiqueta', resultado em mais duas tabelas, sendo esses dois valores um conjunto de chaves primárias compostas.

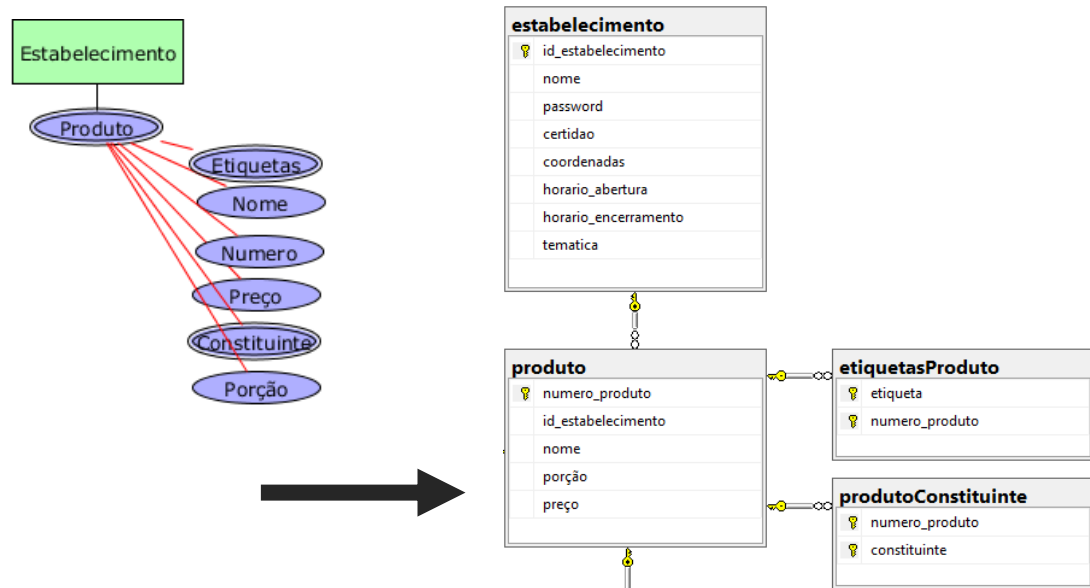


Figura 11 - Transformação atributo multi-valor 'Produto' em tabela

Relativamente à entidade 'Cliente', esta contém dois atributos multi-valor, sendo eles `Etiquetas_Favoritas` e `Historico_Produto`. No que diz respeito a `Etiquetas_Favoritas` irá dar origem a uma nova tabela, tendo como chave primária o conjunto `id_Cliente` (Chave estrangeira) e `etiqueta`.

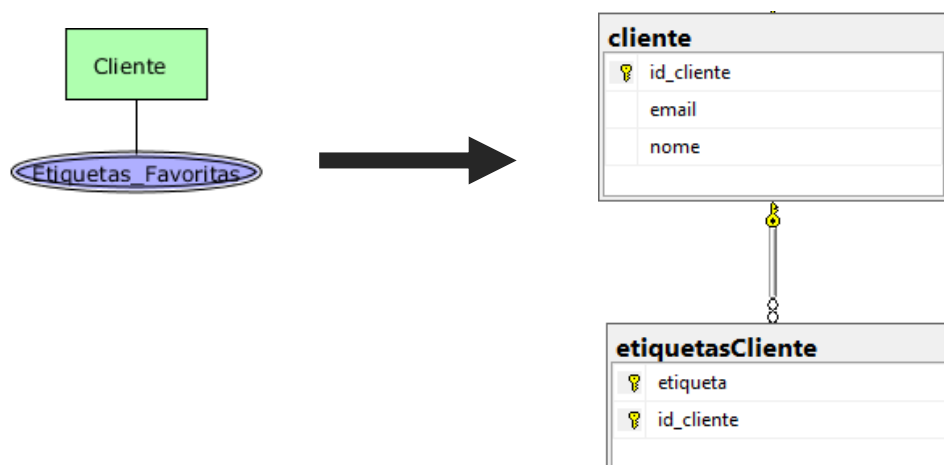


Figura 12 - Transformação atributo multi-valor 'Etiquetas_favoritas' em tabela

Finalmente, relativamente ao atributo `Historico_Produto` este também deu origem a uma nova tabela, tendo `id_Cliente` (Chave Estrangeira) e `numero_produto` (Chave estrangeira) com chave principal composta.

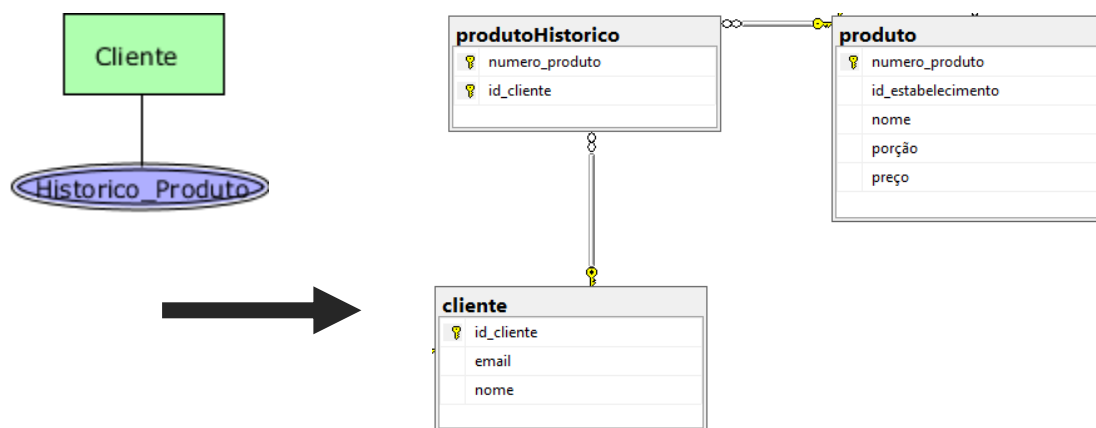


Figura 13 - Transformação do atributo multi-valor 'Historico_produto' em tabela

4.2.4 Normalização

1. 1ª Forma Normal

De modo a uma tabela estar de acordo com a 1ª forma normal, é necessário que uma chave não contenha várias ocorrências do mesmo atributo.

Ora, de acordo com uma análise ao modelo atualmente contruído é possível constatar que em todas as tabelas não existe a ocorrência de grupos repetidos, estando assim, de acordo com a primeira forma normal.

2. 2ª Forma Normal

Como anteriormente foi provado, o modelo encontra-se de acordo com 1ª forma normal, que é o primeiro requisito necessário para estar de acordo com a segunda.

Outra condição necessária a segunda forma normal é que todas as chaves não primárias sejam dependentes da chave primária – algo que, através da observação das tabelas existentes, é também possível de confirmar. Existem, no entanto, chaves que poderiam por si mesmas identificar uma tabela, mas que no entanto foram anteriormente consideradas chaves candidatas, o que as torna também dependentes.

3. 3ª Forma Normal

Finalmente, para uma tabela encontrar-se de acordo com a terceira normal, é necessário que cumpra também a segunda.

De seguida verifica-se se alguma das colunas existentes nessa tabela é dependente de outra(s), algo que não acontece em nenhuma das tabelas existentes no modelo lógico construído.

4.2.5 Elaboração e validação do esquema lógico da base de dados

Após a realização do modelo lógico e da validação feita anteriormente, conclui-se que este se encontra corretamente elaborado e validado. Como resultado final obtemos o seguinte:

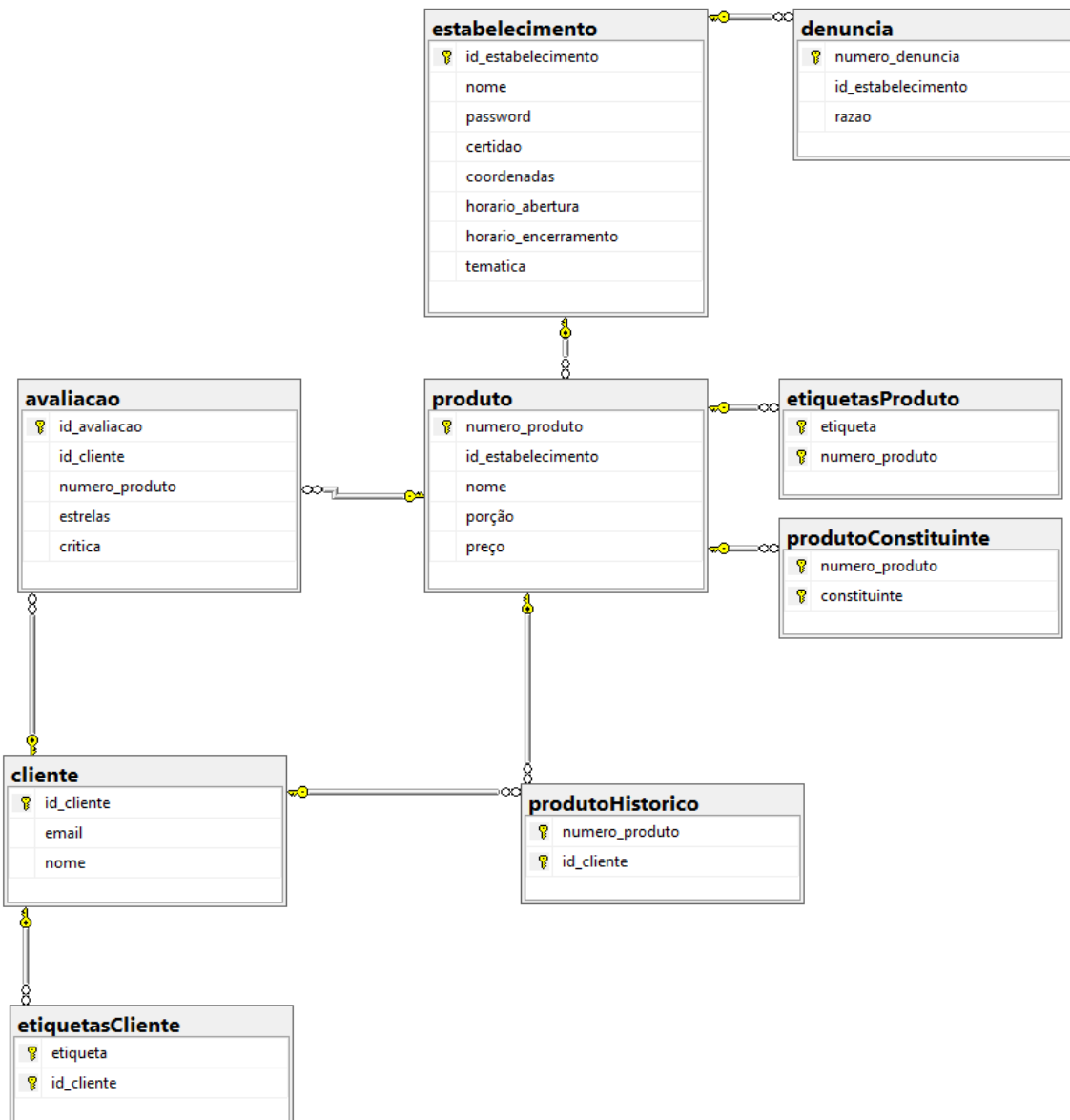


Figura 14 - Modelo Lógico

4.2.4 Análise do crescimento futuro

Embora a atual base de dados foi criada para satisfazer as necessidades anteriormente mencionadas, é bastante flexível e pronta para trabalhar com alterações que sejam pretendidas. Por exemplo, se um proprietário de um estabelecimento quisesse colocar imagens dos seus

produtos e/ou estabelecimento, facilmente seriam criadas tabelas para esse mesmo efeito, não afetando de qualquer maneira os dados atualmente existentes.

4.2.5 Revisão do modelo lógico com o utilizador

Após a realização do modelo lógico, este foi revisto pelo utilizador com o intuito de assegurar que este o considera uma representação fidedigna do que é pretendido. Nenhum problema foi detetado e por isso o modelo de dados foi aceite.

5. Mockups

Neste capítulo serão apresentados os *mockups* realizados para as várias funções da aplicação. As ilustrações pretendem dar uma ideia geral da interface do cliente, ou seja, da camada de apresentação.

Começamos primordialmente com o ecrã inicial, o ecrã onde nos permite nos pesquisar pelas nossas preferências.

Na imagem abaixo representada é possível encontrar uma caixa de texto onde nos permite procurar uma aplicação por teclado, um coração que ilustra o local onde é possível ver os favoritos, o botão de voz, onde é possível pesquisar por voz, e finalmente, no canto superior esquerdo, o menu.

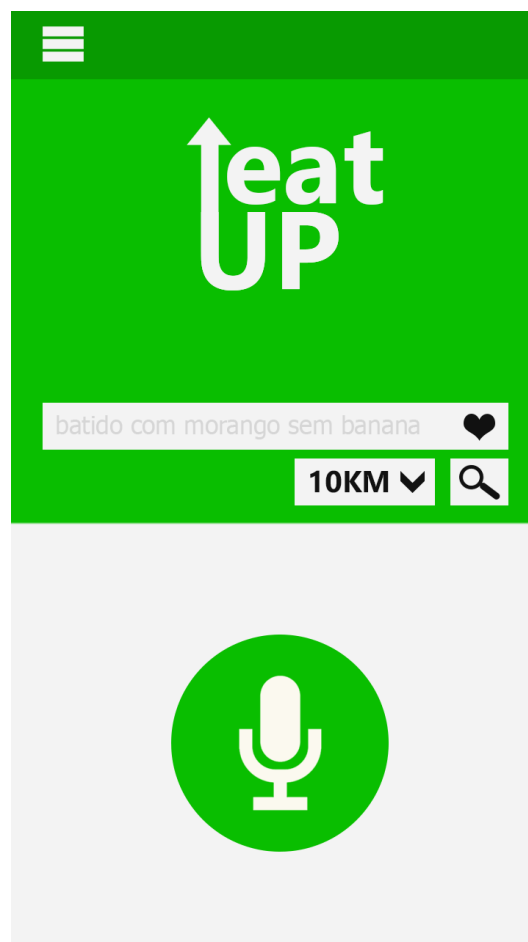


Figura 15 - Mockup Página Principal

Nesse mesmo menu abaixo é possível encontrar as restantes funções criadas para aplicação que não se encontram no menu inicial. São elas 'Adicionar Etiquetas', 'Ver Histórico', 'Histórico de Avaliações' e 'Autenticar' assim como está abaixo representado.

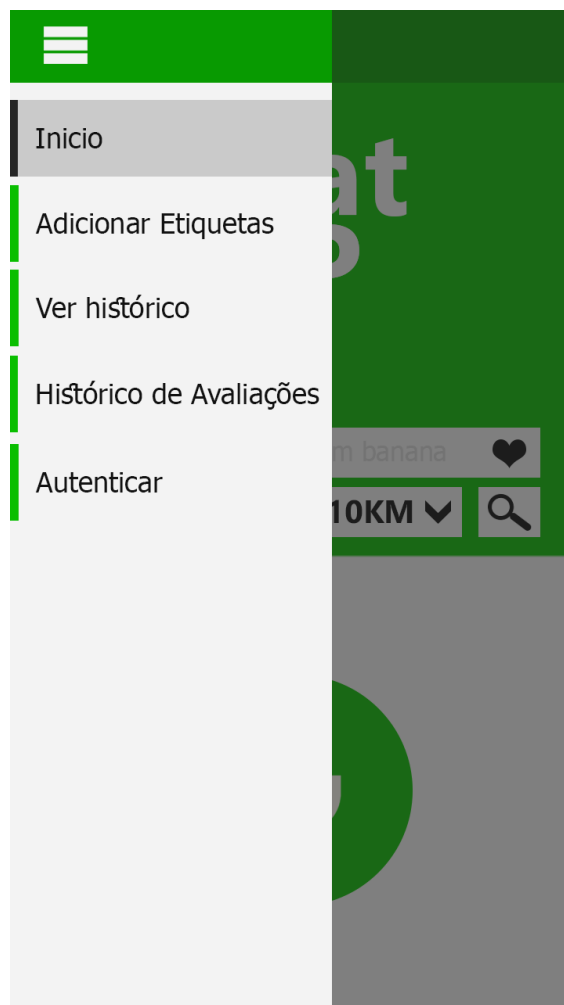


Figura 16 - *Mockup Menu*

No que diz respeito aos resultados após qualquer tipo de pesquisa, serão apresentados com o nome do estabelecimento, produto e os seus constituintes, ordenados pela distância atual, assim como é possível abaixo confirmar:

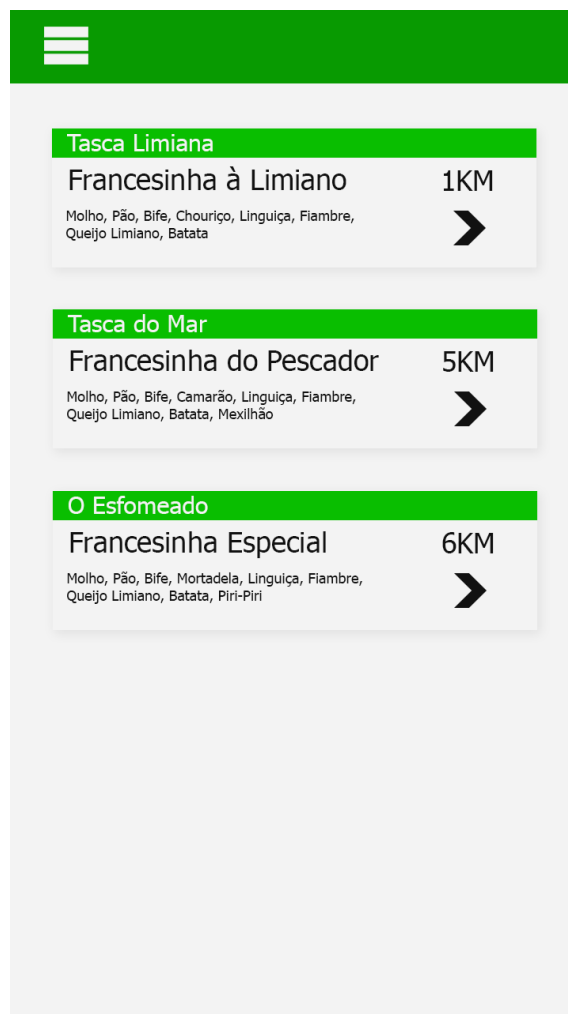


Figura 17 - *Mockup* Resultados de Pesquisa

Finalmente, no que diz respeito às entradas dos menus temos 'Adicionar Etiquetas', 'Ver Histórico', 'Histórico de Avaliações'. Começando pela primeira, é possível verificar uma caixa de texto onde colocar etiquetas e uma caixa abaixo com as etiquetas já existentes.



Figura 18 - *Mockup* Adicionar Favoritos

No que toca ao histórico temos um menu onde se encontram produtos onde adicionamos ao histórico após pesquisa. A cada entrada será possível partilhar nas redes sociais, reportar e escrever uma avaliação.

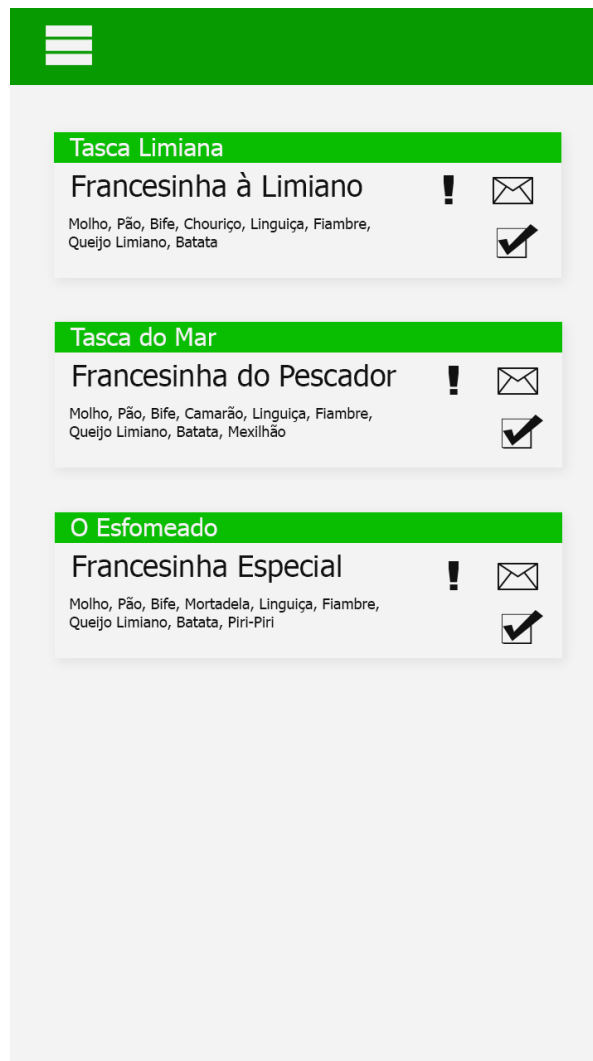


Figura 19 - Mockup Ver Histórico

Por último temos o histórico de avaliações onde estarão as avaliações feitas pelos clientes aos restaurantes visitados. Nesse local será possível partilhar essas mesmas avaliações nas redes sociais.

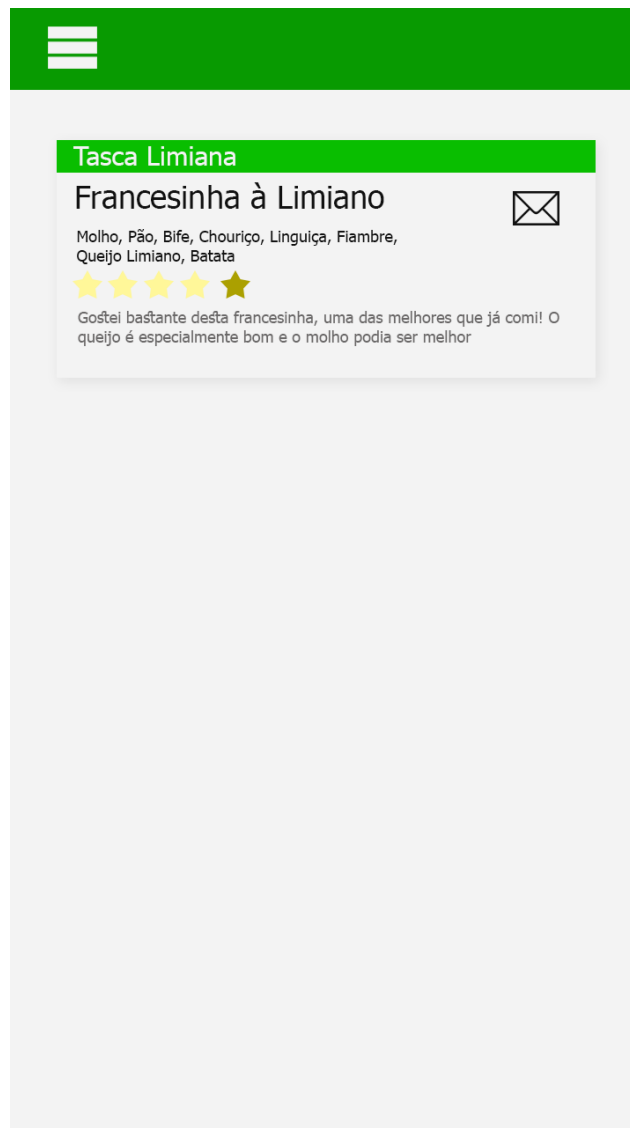


Figura 20 - *Mockup* Ver avaliações

6. Implementação

Após concluídas as diferentes etapas de modelação – quer da arquitetura da aplicação quer da base de dados – chegou o momento de implementar as funções acima descritas. Partindo dos pressupostos e requisitos indicados ao longo do relatório, começou-se por desenvolver a aplicação em si, isto é, o Front-Office que estará disponível aos utilizadores. Tomou-se esta decisão por se tratar da parte mais fulcral e complexa de todo o projeto.

6.1. Ferramentas utilizadas

Previamente à implementação do projeto, foi necessário estudar qual as melhores ferramentas disponíveis para o nosso tipo de necessidade – a plataforma de desenvolvimento, a tecnologia de suporte e o *software* de trabalho.

6.1.1 Plataforma de desenvolvimento

Sendo um dos objetivos o escalamento da aplicação, isto é, a possibilidade de a correr em vários tipos de dispositivos, levou-nos a averiguar a plataforma que melhor nos proporcionará uma maior escala de uso. Para tal, foi escolhido o *software* que qualquer smartphone ou desktop de hoje dia tem – o *browser* – pelo imenso poder que atualmente detém para nos satisfazer as necessidades. É possível também no futuro, construir um simples *wrapper* que utiliza o mesmo código para conceber uma aplicação.

6.1.2 Tecnologia de suporte

Tendo escolhido o *browser* para suportar a aplicação, falta agora saber com a que vamos construir e a ferramenta de eleição será *web-forms ASP.net*. Esta plataforma permitirá implementar todos nossos requisitos e possui uma poderosa documentação e suporte a nível de ferramentas.

6.1.3 Visual Studio 2015

O uso deste poderoso IDE no desenvolvimento deste projeto deve-se ao facto de ser um IDE que possui múltiplas ferramentas e funcionalidades que ajudam no desenvolvimento de aplicações de forma segura e adaptáveis as várias plataformas. Isso traz como acréscimo melhorias na produtividade da equipa e na gestão do ciclo de vida da aplicação e permite a utilização das mais recentes tecnologias.

6.2. Estrutura inicial

Uma das funcionalidades do ASP.net é existência de páginas mestre, isto é, uma página que estrutura todas as páginas a ela conjugadas. Foi então criada uma estrutura que todas as páginas iriam ter, permitindo desta maneira existir consistência em todas, contendo os mesmos estilos e atributos.

A partir de tal criou-se uma página esqueleto que servirá de base para todas as interações no website.

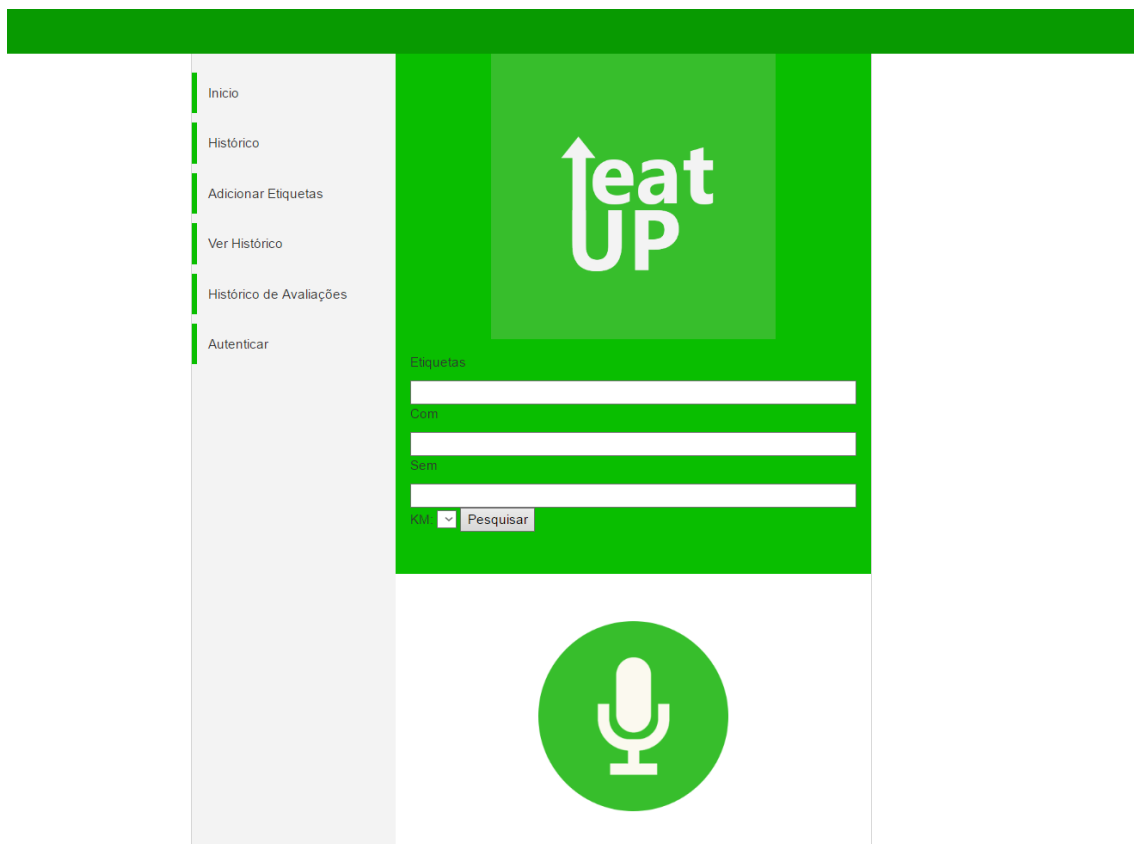


Figura 21 - Estrutura básica

6.3. Preenchimento da Base de Dados

O preenchimento da base de dados passou pela inserção manual de dados de estabelecimentos conhecidos, bem como alguns pratos e as restantes tabelas necessárias de modo a testar o funcionamento inicial da aplicação. Após esta fase seriam usados dados reais de através de APIs como a do Yelp para obtê-los.

6.4. Conexão Base de Dados

Num ponto de vista útil e de boa programação, o correto seria criar DAOs por classe representada no dia Diagrama de classes. No entanto, por falta de tempo foi apenas desenvolvido trabalho no que diz respeito à procura de produtos via texto e consequentemente todas as ligações à base de dados que foram feitas nas classes respetivas a cada página. Não obstante, o processo de ligação será o mesmo. Será necessário configurar o ficheiro `web.config` com o respetivo endereçamento à nossa base de dados e está pronto a receber *queries*, como abaixo está representado.

```
using (SqlConnection cn = new
SqlConnection(ConfigurationManager.ConnectionStrings["data_base"].ToString()))
{
    cn.Open();
    using (SqlCommand cmd = new SqlCommand(command.ToString(), cn))
    {
        SqlDataReader rdr =
cmd.ExecuteReader(CommandBehavior.CloseConnection);

        if (rdr.HasRows) {
            while (rdr.Read())
            {
                e_names.Add(rdr[2].ToString(), rdr[0].ToString());
                p_names.Add(rdr[2].ToString(), rdr[1].ToString());
                e_id.Add(rdr[2].ToString(), rdr[3].ToString());
            }
        } else
        {
            Response.Redirect("Default.aspx?noresults=true");
        }

        cn.Close();
    }
}
```

Figura 22 - Exemplo ligação a base de dados

6.5. Funcionalidades

Como já mencionado, por falta de tempo foi apenas trabalhado em ter produto a obedecer às *mockups* e realizar a pesquisa por texto.

Passando então para o funcionamento desta função, passa por inicialmente validar os campos introduzidos pelo utilizador – caso não existam tags ou não estejam devidamente preenchidas.



Figura 23 - Erro de preenchimento inválido

Os dados relativos a essa pesquisa serão redirecionados para outra página via URL, que os recebe como parâmetro. O *backend* dessa página irá ler os parâmetros e efetuar uma *query* na base de dados de modo a obter resultados correspondentes, apresentando-os na página.



Etiquetas

francesinha,cozido

Com

Sem

ovo

KM: Pesquisar

Figura 24 - Exemplo de pesquisa

(...)/Resultados?tags=francesinha,cozido&positive=&negative=ovo

Figura 25 - Exemplo de redirecionamento

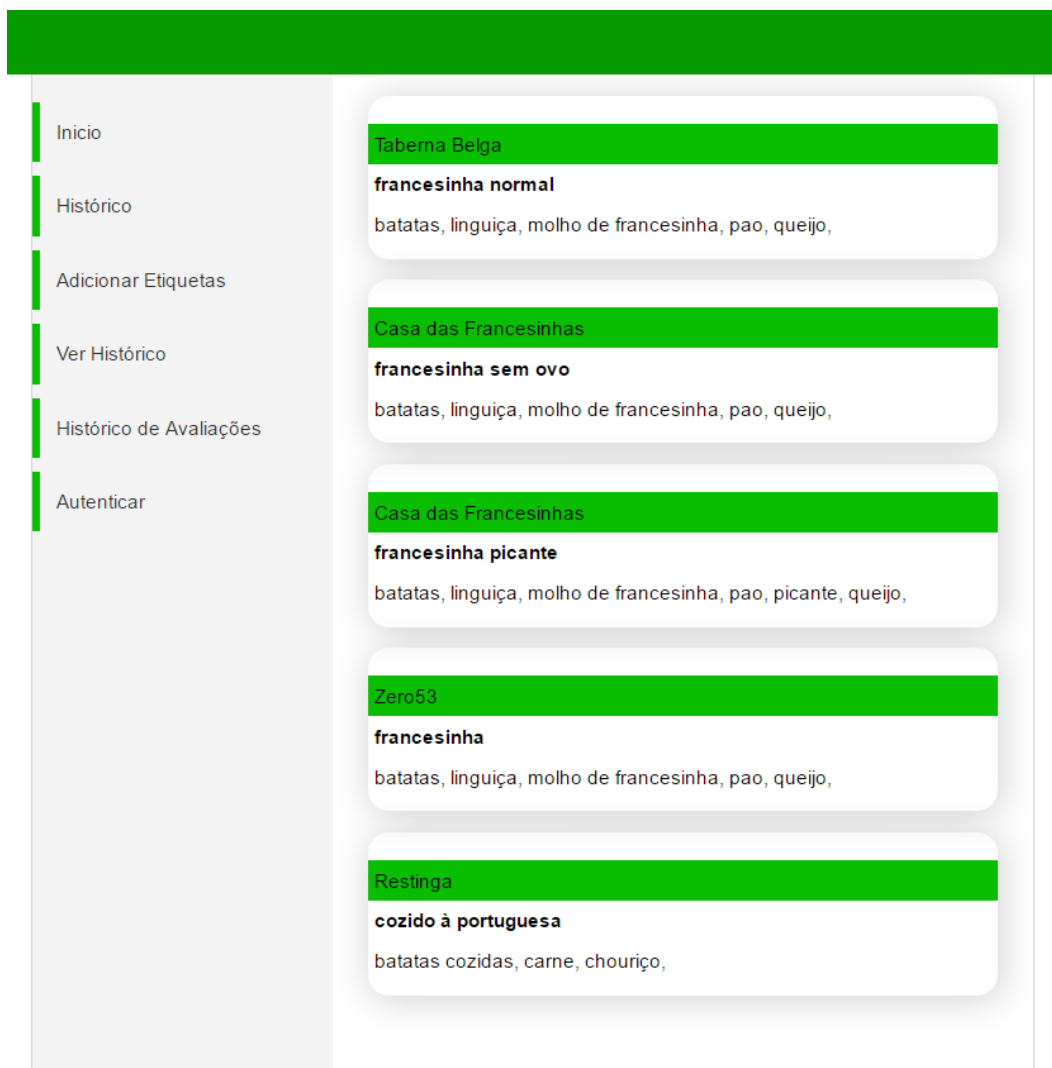


Figura 26 – Resultados

Após clicado num dos resultados, é redirecionado para uma página que conterà como parâmetros a identificação do estabelecimento que será utilizada para efetuar uma *query* à base de dados de modo a obter as coordenadas. Após obtidas as coordenadas, são calculadas as atuais, e com recurso à API do *Bing Maps* desenha-se o percurso no mapa.

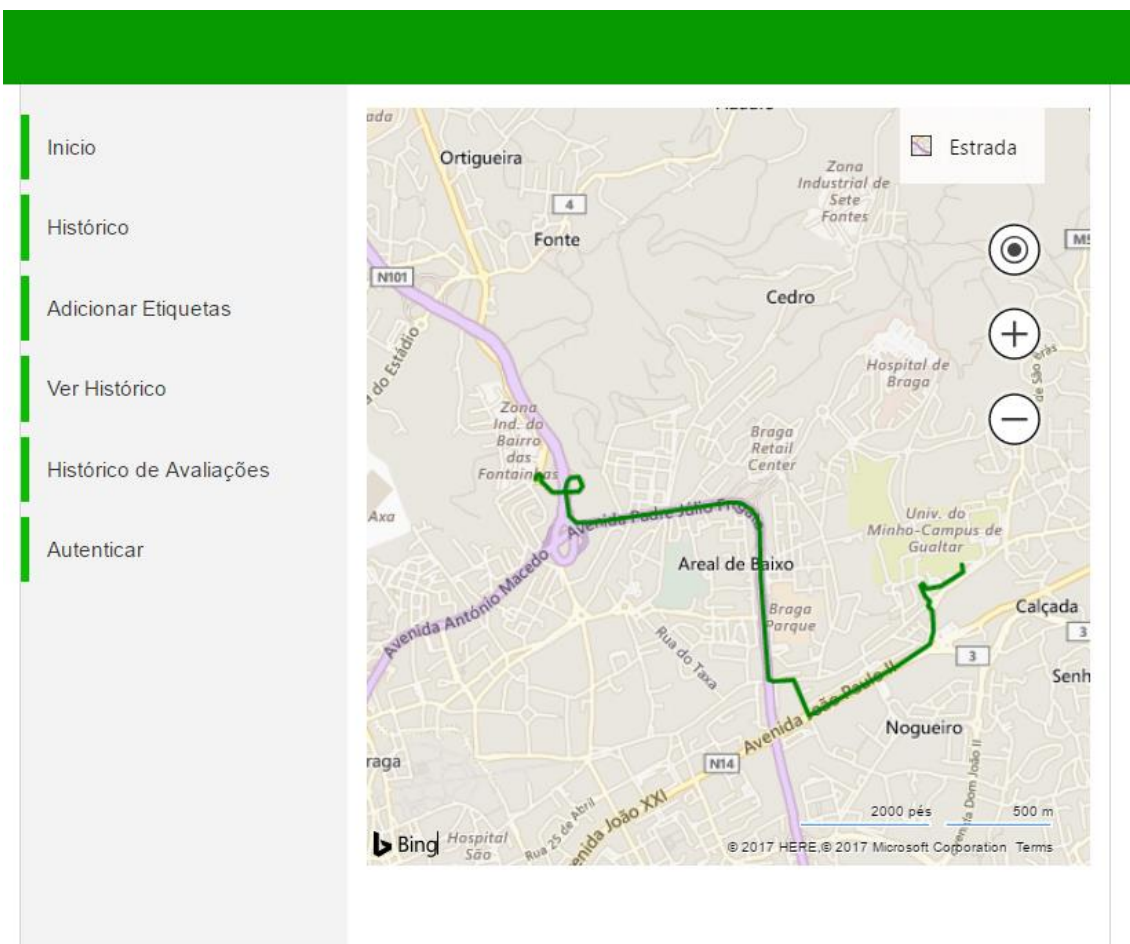


Figura 27 - Percurso a efetuar

7. Conclusões e Trabalho Futuro

Terminada a primeira etapa do projeto, damos por concluída a fase da fundamentação do projeto. Inicialmente, realizamos uma recolha e análise de informações onde identificamos e compreendemos as principais necessidades dos utilizadores. Esta recolha de informação teve por base as opiniões dos utilizadores e também outras aplicações existentes no mercado, onde identificamos as suas respetivas lacunas que o nosso sistema deve conseguir corrigir. Com este conhecimento fundamentamos o caso de estudo, as motivações e objetivos da aplicação a implementar. Esta compreensão detalhada de como se comporta o sistema permite uma idealização e implementação mais eficazes.

Com o planeamento das atividades, foi possível ter uma melhor perceção das tarefas a realizar no futuro. Por outro lado, a distribuição de tarefas pelo grupo de trabalho e a previsão de períodos de tempo para as várias tarefas permitiu uma melhor organização da construção do projeto.

O passo seguinte para a elaboração do projeto consistiu na especificação do sistema, onde foram levantados os vários tipos de requisitos e elaborada a modelação do sistema, através do uso de diagramas em UML.

De modo a uma melhor estruturação dos diagramas, foi realizada um levantamento de requisitos. Assim, procedeu-se à devida separação dos mesmos em requisitos de utilizador e de sistema, sendo estes últimos divididos ainda em requisitos funcionais e não funcionais. Esta organização permitiu compreender melhor as funcionalidades que o sistema deve implementar.

Em seguida, procedeu-se à elaboração dos modelos em UML onde foram desenvolvidos vários diagramas que permitiram interpretar aspetos fundamentais que o sistema deve implementar. Deste modo, foi elaborado um diagrama de *Use Case* que permite visualizar as funcionalidades a implementar e quais os utilizadores que irão usufruir das mesmas. Os diagramas de sequência permitiram ter uma perceção dos comportamentos entre os utilizadores e o sistema, bem como o ordenamento temporal das mensagens trocadas entre estes. Por fim, o diagrama de classes foi útil para obter uma perspetiva sobre a arquitetura do sistema a desenvolver.

Após desenvolvidos os modelos, procedeu-se ao desenvolvimento da base de dados de acordo com o levantamento de requisitos. Deste modo, o primeiro passo foi identificar as entidades, os atributos e os relacionamentos, que permitiram a construção do modelo conceptual. Em seguida, procedeu-se à passagem para o modelo lógico, onde verificamos se o modelo evitava redundâncias.

Foram ainda construídos os *mockups* que permitem ter uma perceção da *interface* que o sistema irá respeitar.

Na terceira e última fase deste projeto foi tratada a implementação do que foi especificado previamente, dado à possibilidade que tivemos – que diferiu totalmente do que

esperávamos quando criamos o diagrama de Gantt. Ainda nesta fase foram testadas as funcionalidades instauradas na aplicação de modo verificar o seu correto funcionamento.

Relativamente a trabalho futuro, passaria por implementar todas as funções e mudar a estrutura da aplicação consoante ao que foi especificado anteriormente. Será também importante implementar a interface para os proprietários de negócio. Foi um importante trabalho para introdução a engenharia de *software* e todas as componentes a ela ligadas, bem como a importância de lidar com imprevistos e tomada de decisões.

Referências

Yelp [online] Disponível em: <https://www.yelp.com/sf> [Acedido a 3 de Março de 2017]

TripAdvisor [online] Disponível: <https://www.tripadvisor.com/> [Acedido a 3 de Março de 2017]

Foursquare [online] Disponível: <https://www.foursquare.com/> [Acedido a 3 de Março de 2017]

Sommerville, I. (2011). *Software engineering*. 9th ed. Boston, Massachusetts: Addison-Wesley.

Connolly, T. and Begg, C. (2005). *Database systems*. 4th ed. Harlow, Essex, England: Addison-Wesley.

Lista de Siglas e Acrónimos

API	<i>Application program interface</i>
GPS	<i>Global Positioning System</i>
IDE	<i>On-Line Analytical Processing</i>
UML	<i>Unified Modeling Language</i>
SBD	<i>Sistema de Base de Dados</i>

Anexos

I. Anexo 1 – Especificação de Use Case

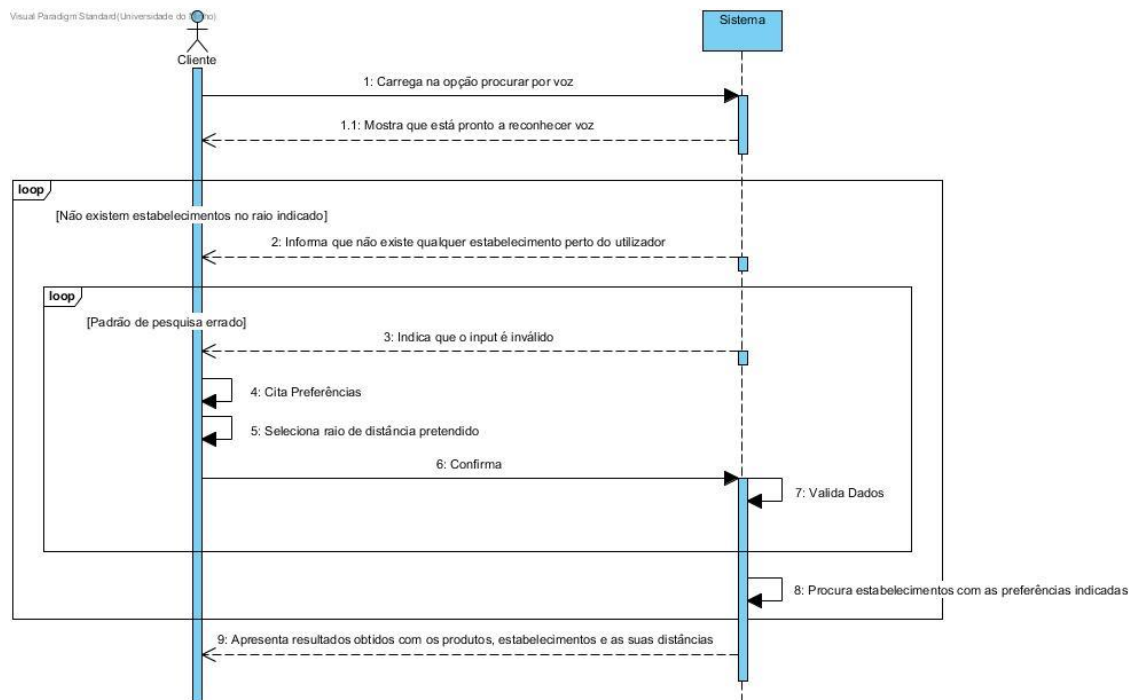
Especificação de Use Case			
Name	Procurar produto - via voz.		
Brief Description	Permite a um cliente procurar um produto com as restrições desejadas.		
Preconditions	n/a		
Post-conditions	Apresenta lista de estabelecimentos com o produto pretendido.		
Flow of events		Actor Input	System Response
	1	Carrega na opção procurar por voz.	
	2		Mostra que está pronto a reconhecer voz.
	3	Cita preferencias.	
	4	Seleciona raio de distancia pretendido.	
	5	Confirma.	
	6		Valida dados.
	7		Procura estabelecimentos com preferencias indicadas.
	8		Apresenta resultados obtidos com as produtos que correspondem bem como os estabelecimentos e as suas distancias.
Exceção 1 (passo 6) [Padrão de pesquisa errado]	1		Indica que input é inválido
	2		Volta a 2.
Exceção 2 (passo 7) [Não existe estabelecimentos no raio indicado]	1		Informa que não existe qualquer estabelecimento perto do utilizador
	2		Volta a 2.

Anexo 1 - Especificação de Use Case Procurar produto - via voz

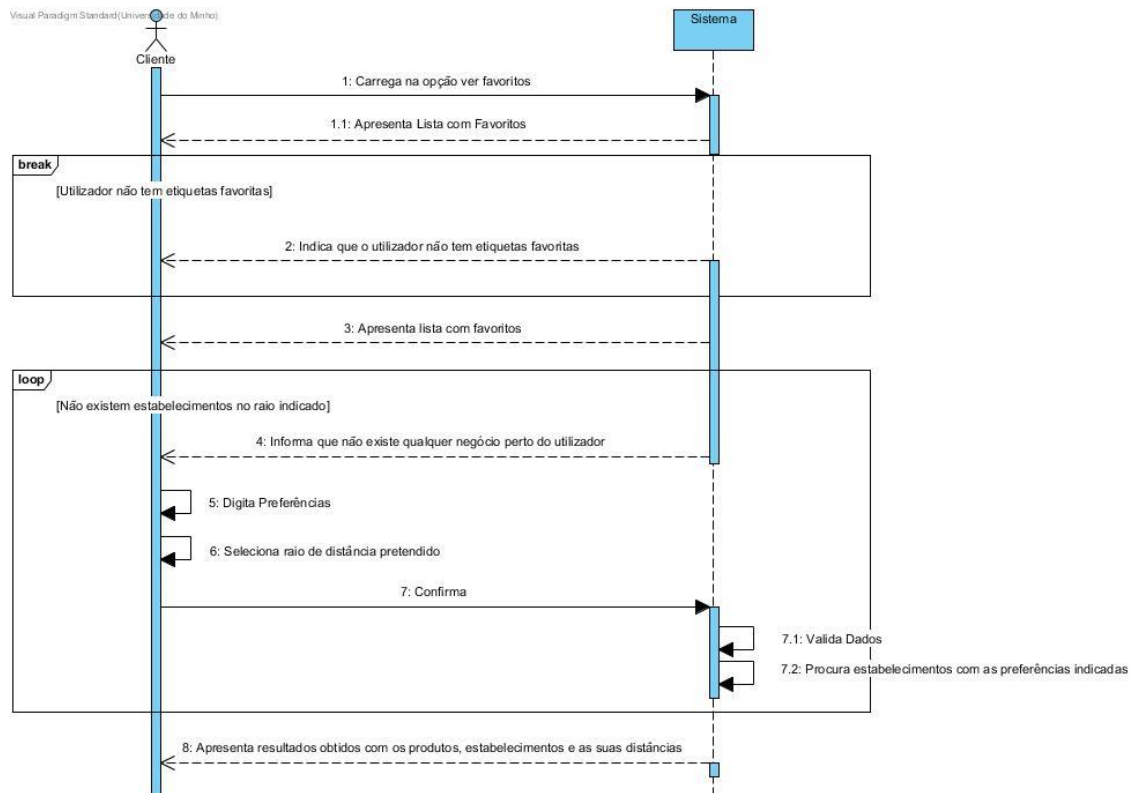
Especificação de Use Case			
Name	Procurar etiquetas favoritas		
Brief Description	Permite a um cliente procurar um produto com as restrições desejadas.		
Preconditions	n/a		
Post-conditions	Apresenta lista de negócios com o produto pretendido.		
Flow of events		Actor Input	System Response
	1	Carrega na opção ver favoritos.	
	2		Apresenta lista com favoritos.
	3	Escolhe etiquetas pretendidas.	
	4	Seleciona raio de distancia pretendido.	
	5	Confirma.	
	6		Valida dados.
	7		Procura negócios com preferencias indicadas.
	8		Apresenta resultados obtidos com as ementas que correspondem bem como os negócios e as suas distancias.
Exceção 1 (passo 3) [Utilizador não tem etiquetas favoritas]	1		Indica que utilizador não tem etiquetas favoritas.
	2		Volta a menu principal.
Exceção 2 (passo 7) [Não existe negócios no raio indicado]	1		Informa que não existe qualquer negócio perto do utilizador
	2		Volta a 2.

Anexo 2 - Especificação de Procurar etiquetas favoritas

II. Anexo 2 – Diagramas de Sequencia



Anexo 3 - Diagrama Sequência Procurar produto - via voz



Anexo 4 - Diagrama de Sequência Procurar etiquetas favoritas