



Universidade do Minho  
Escola de Engenharia  
Mestrado Integrado em Engenharia Informática

## Unidade Curricular de Laboratórios de Informática IV

Ano Lectivo de 2016/2017

***Chewie***

**Carlos Pereira, a61887; Diogo Silva, a76407;  
Maria Ana de Brito, a73580; Rafael Costa, a61799**

Abril, 2017

Data de Recepção	
Responsável	
Avaliação	
Observações	

## ***Chewie***

**Carlos Pereira, a61887; Diogo Silva, a76407;**

**Maria Ana de Brito, a73580; Rafael Costa, a61799**

Abril, 2017

## Resumo

O projeto consiste no desenvolvimento de uma aplicação que compreenda todas as fases de desenvolvimento de uma peça de software. Assim, um projeto viável deve apresentar uma metodologia, processo e técnicas de desenvolvimento assentes num conjunto bem definido de etapas. Assim, este relatório compreende a primeira etapa do desenvolvimento de *Chewie*: fundamentação, incluindo a análise dos requisitos da aplicação, como também a segunda fase relativa à especificação em *UML* da aplicação.

Inicialmente devemos-nos certificar em definir o contexto, expondo o caso de estudo. Após a caracterização do problema deve-se definir a motivação e quais os objetivos pretendidos atingir aquando a finalização do projeto. Para tal ser possível, é necessário justificar o sistema (salientando o contacto com o utilizador), expondo a utilidade da aplicação e identidade do projeto.

De forma a diminuir o nível de abstração, procede-se com uma breve definição da maquete do sistema, permitindo uma primeira visualização do funcionamento da aplicação.

Finalmente, partimos para a especificação em *UML* das várias componentes do sistema, como também o desenho do Sistema de Base de Dados.

**Área de Aplicação:** Engenharia de Software, Base de Dados, Aplicações *Mobile*, Aplicações *Desktop*

**Palavras-Chave:** *Chewie*, utilizador, gastronomia, pesquisa, restaurante, avaliação, comentários, distância, sistema de recomendação

# Índice

1. Introdução	1
1.1. Contextualização	1
1.2. Apresentação do Caso de Estudo	1
1.3. Motivação e Objectivos	1
1.4. Justificação do Sistema	2
1.5. Utilidade do Projeto	3
1.6. Identidade do Projeto	3
1.7. Recursos Necessários	3
1.8. Maqueta do Sistema	4
1.9. Conjunto de Medidas de Sucesso	5
1.10. Planeamento do Desenvolvimento	6
1.11. Levantamento de Requisitos	6
1.11.1 Pesquisa por texto	6
1.11.2 Pesquisa por voz	7
1.11.3 Apresentação dos resultados	7
1.11.4 Filtragem de resultados	7
1.11.5 Informações a exibir	7
1.11.6 Histórico de pesquisa	7
1.11.7 Partilha numa rede social	8
1.11.8 Comentários sobre o prato pesquisado	8
1.11.9 Sugestões de restaurantes a visitar	8
1.11.10 Tempo de resposta da aplicação	8
1.11.11 Memória ocupada pelos dados	8
1.11.12 Facilidade de utilização	8
2. Análise de Requisitos	9
2.1. Interface Layer	9
2.1.1 Input	9
2.1.2 Apresentação dos Resultados	10
2.2. Business Layer	11
2.2.1 Login	11
2.2.2 Opinião	12

2.2.3 Preferências	12
2.2.4 Histórico	13
2.3. Servidor	13
2.3.1 Pesquisa	13
2.3.2 Base de Dados	13
3. Especificação do Software em UML	14
3.1. Modelo de Domínio	14
3.2. Diagrama de Use Cases	15
3.3. Especificação de Use Cases	16
3.3.1 Login	16
3.3.2 Pesquisa	17
3.3.3 Analisa Parâmetros	18
3.3.4 Histórico	20
3.3.5 Sugestões	22
3.3.6 Refinamento da Pesquisa	24
3.3.7 Input por Voz	26
3.3.8 Input por Texto	26
3.3.9 Comentar Estabelecimento	27
3.3.10 Avaliar Estabelecimento	27
3.3.11 Partilha na Rede Social	28
3.4. Diagrama de Classes	28
3.4.1 Cliente	29
3.4.2 Servidor	30
4. Projeto do Sistema de Dados	31
4.1. Modelo Lógico	31
4.1.1 Tabela Utilizador	31
4.1.2 Tabela Preferências	32
4.1.3 Tabela Histórico	32
4.1.4 Tabela Crítica	32
4.1.5 Tabela Prato	32
4.1.6 Tabela Ementa	32
4.1.7 Tabela Estabelecimento	33
4.1.8 Tabela Contacto	33
4.1.9 Tabela Horario_Funcionamento	33
4.2. Dicionário de Dados	34
5. Especificação das Interfaces	39
5.1. Login	39
5.2. Home	40
5.3. Histórico	41
5.4. Resultados	42

5.5. Estabelecimento	43
5.6. Preferências	44
5.7. Comentários	45
5.8. API's	45
6. Conclusões e Trabalho Futuro	47

## **Anexos**

I. Anexo 1	50
II. Anexo 2	53
III. Anexo 3	55

## Índice de Figuras

Figura 1 - Maqueta do Sistema	5
Figura 2 - Diagrama de Gantt	6
Figura 3 - Modelo de Domínio	14
Figura 4 - Diagrama de Use Cases	15
Figura 5 - Diagrama de Sequência do Login	17
Figura 6 - Diagrama de Sequência da Análise de Parâmetros	19
Figura 7 - Diagrama de Sequência do Histórico	21
Figura 8 - Diagrama de Sequência de Sugestões	23
Figura 9 - Diagrama de Sequência do Refinamento da Procura	25
Figura 10 - Diagrama de Sequência do Input por Voz	26
Figura 11 - Diagrama de Sequência de Input por Texto	26
Figura 12 - Diagrama de Sequência de Comentar Estabelecimento	27
Figura 13 - Diagrama de Sequência de Avaliar Estabelecimento	27
Figura 14 - Diagrama de Sequência da Partilha na Rede Social	28
Figura 15 - Diagrama de Classes do Cliente	29
Figura 16 - Diagrama de Classes do Servidor	30
Figura 17 - Modelo Lógico	31
Figura 18 - Mockup do Login	39
Figura 19 - Mockup da janela Home	40
Figura 20 - Mockup do Histórico	41
Figura 21 - Mockup dos Resultados	42
Figura 22 - Mockup da Seleção de um Estabelecimento	43
Figura 23 - Mockup das Preferências	44
Figura 24 - Mockup dos Comentários	45

## Índice de Tabelas

Tabela 1 - Use Case de Login	16
Tabela 2 - Use Case da Análise de Parâmetros	18
Tabela 3 - Use Case do Histórico	20
Tabela 4 - Use Case das Sugestões	22
Tabela 5 - Use Case do Refinamento da Pesquisa	24



# 1. Introdução

## 1.1. Contextualização

*Chewie* é um *software* que visa facilitar a procura de pratos de forma genérica no âmbito da região de Braga. Isto é, o utilizador será convidado a indicar o "algo" que pretende degustar e a procura indicará um conjunto de estabelecimentos num perímetro dentro do território em que o utilizador se insere. Este processo poderá ser efetuado via comunicação de voz ou introdução de texto de forma a tornar o procedimento mais rápido, conciso e eficaz.

Esta aplicação, possível de utilizar no *desktop*, também voltada à plataforma *mobile* poderá ser utilizada num contexto regional tendo os resultados naturalmente influenciados pela mesma.

*Chewie* pretende "compreender" e ajudar o utilizador a encontrar a melhor gastronomia possível, desta forma, os pratos oferecidos ao mesmo terão em conta a avaliação, as preferências de outros utilizadores e a distância ao mesmo. Pretende-se que seja encontrada a melhor solução para o utilizador.

## 1.2. Apresentação do Caso de Estudo

*Chewie* é um sistema de recomendação de restaurantes com base na pesquisa de um prato gastronómico indicado pelo utilizador.

## 1.3. Motivação e Objectivos

Suponhamos que temos um utilizador que desconhece a região onde se encontra, a nível gastronómico. Esse utilizador tem vontade de experimentar um prato local dessa região. Contudo existem certos ingredientes que não aprecia ou que não deve comer, mas não sabe se esses ingredientes são usados nessa região.

Imaginemos também, um outro utilizador faminto e que conhece a região onde se encontra. Contudo, este não quer pensar ou ter que fazer uma pesquisa morosa em pratos na sua zona.

Este projeto surge da necessidade de implementar um motor de busca gastronómico a nível da região de Braga, que tem em consideração as preferências do utilizador, apresentando-lhe os resultados que mais se adequam a ele. Oferecendo também ao mesmo tempo uma interface de fácil uso e funcionalidade, possibilitando a aceitação dos parâmetros de pesquisa por voz. Isto tudo porque não existe no mercado uma aplicação que consiga juntar ao mesmo tempo a pesquisa de pratos gastronómicos com as preferências do próprio utilizador e que seja de uso simples.

Pretendemos assim ter uma aplicação simples, que evite ter o utilizador a perder tempo fazendo pesquisas gastronómicas, pois será a nossa plataforma a fazer por ele. Também queremos que os resultados sejam apresentados de maneira fácil, indicando meios de contacto, avaliações, comentários e até direções ao estabelecimento que o utilizador selecionar de uma lista de resultados. Por fim almejamos em ter uma plataforma que seja rápida no tempo de resposta, visto ser um fator de grande sensibilidade no contexto da hora de refeição.

## **1.4. Justificação do Sistema**

O sistema que nos propomos a desenvolver tem uma aplicação prática no quotidiano dos utilizadores, uma vez que vai ser possível pesquisar os melhores restaurantes para degustação de um prato indicado pelo utilizador, segundo os seus critérios. Desta forma, quando o utilizador pesquisa um certo prato, os restaurantes à escolha ser-lhe-ão mostrados por uma ordem que conjuga vários fatores, que visam garantir a sua preferência. Se, por outro lado, pretender uma pesquisa personalizada, poderá definir quais as suas preferências de procura.

Esta aplicação também promove a mobilidade dos utilizadores, pois estes poderão realizar a pesquisa por comandos de voz. Assim, caso pretendam, poderão apenas interagir por voz com a aplicação, descartando a necessidade de introduzir dados manualmente. Esta funcionalidade é útil em situações em que o utilizador não tem qualquer possibilidade de indicar à mão qual o prato que pretende degustar. Se, por exemplo, alguém estiver a conduzir, não pode quebrar a sua atenção para introduzir os dados manualmente, correndo o risco de causar um acidente rodoviário. No entanto, o utilizador poderá usar esta aplicação, uma vez que lhe é oferecido um serviço de voz, tornando a sua condução segura.

O cliente, além de ler os comentários de um restaurante, também poderá, sempre que entender, deixar os seus próprios comentários nos restaurantes que visitou ou partilhar em direto a sua experiência gastronómica.

Além disso, a aplicação terá uma utilidade real no mercado, dado que não existe nenhuma outra que conjugue estes fatores, pois mais nenhuma permite fazer uma pesquisa de um prato gastronómico, de acordo com os critérios pessoais dos utilizadores, promovendo, também, a mobilidade dos mesmos.

## 1.5. Utilidade do Projeto

O principal intuito da aplicação *Chewie* é facilitar ao cliente a procura de um prato específico. Um utilizador pode então obter várias opções de locais que oferecem o seu prato desejado independentemente do local onde se encontra. A lista de resultados é filtrada de acordo com a qualidade do restaurante e o prato procurado, bem como as preferências do utilizador. Desta maneira, se um utilizador não gostar de um prato servido de uma certa forma (com algum acompanhante, especiaria, molho, etc.), então na lista de resultados não estará incluído o restaurante que serve este prato. Os resultados são acompanhados pela morada do restaurante e respetivo trajeto até este, sendo bastante proveitoso especialmente em casos que o utilizador se encontra em cidades que não conhece.

A aplicação mantém um histórico dos restaurantes visitados por um utilizador possibilitando um rápido acesso aos resultados de uma pesquisa anterior. É possível, também, registar uma opinião de um dado restaurante ou prato numa rede social.

## 1.6. Identidade do Projeto

O projeto a ser desenvolvido trata-se de um sistema de recomendação que, após a indicação de um prato gastronómico, pesquisa quais os melhores locais para realizar a sua degustação. O resultado da procura deverá mostrar uma lista dos melhores restaurantes selecionados de acordo com as preferências do utilizador, bem como as avaliações efetuadas por parte de outros utilizadores e a distância ao local em que se encontra num dado momento.

Além disso, a aplicação também se caracteriza por se enquadrar no mundo moderno, em que a maior parte dos utilizadores está ligada a uma rede social e partilha o que está a fazer ou o que sente em tempo real. Desta forma, é possível partilhar a sua experiência gastronómica num determinado restaurante, numa rede social à sua escolha ou, então, fazer um *tweet* à medida que degusta o prato escolhido.

Em síntese, a aplicação *Chewie* encontra-se direcionada para os utilizadores, de forma a proporcionar-lhes a melhor pesquisa de restaurantes, continuamente atualizada e fiável.

## 1.7. Recursos Necessários

Para o desenvolvimento deste projeto é necessária uma base de dados relacional povoada com dados provenientes de alguns *sítes* de opinião sobre gastronomia. Os *sítes* selecionados devem fornecer as seguintes informações sobre os restaurantes: nome do estabelecimento, morada, contacto, foto, bem como comentários e avaliações feitas por utilizadores.

## 1.8. Maqueta do Sistema

A aplicação tem uma arquitetura Cliente-Servidor, sendo que existem dois tipos de Cliente que o Servidor terá de servir (**Figura 1**). Os Clientes podem ser:

- Cliente *Mobile* - clientes em plataformas móveis, como *Smartphones* e *Tablets*;
- Cliente *Desktop* - clientes em ambiente *Desktop*.

Neste sistema temos um Servidor que responde aos Clientes pelas vias de comunicação apropriadas (através de *sockets*), sendo este capaz de servir mais do que um ao mesmo tempo. Assim, desta forma, será possível ter a aplicação em dispositivos/aparelhos que poderão não ter a capacidade de aguentar os seus requisitos. Sendo assim, o Servidor vai estar ligado a uma base de dados que irá armazenar toda a informação necessária para responder aos pedidos dos vários Clientes. Para além disso, é ainda, da responsabilidade do servidor autenticar um Cliente através do seu nome de utilizador e da sua palavra-passe.

Podemos dividir o Cliente em duas camadas fundamentais: a *Interface Layer* e a *Business Layer*. A *Interface Layer* trata da interação com o utilizador. É responsável pelo tratamento de todos os possíveis *inputs* do utilizador (texto ou voz) e pela apresentação de todos os resultados pretendidos. Já a *Business Layer* trata da extração de todos os campos relevantes de uma qualquer pesquisa, armazenamento de dados que possam ser visualizados por um utilizador (estabelecimentos, histórico de um utilizador, sugestões, ...) e envia pedidos ao Servidor. É importante referir que estas duas camadas mencionadas são ligadas através de um módulo *Facade*.

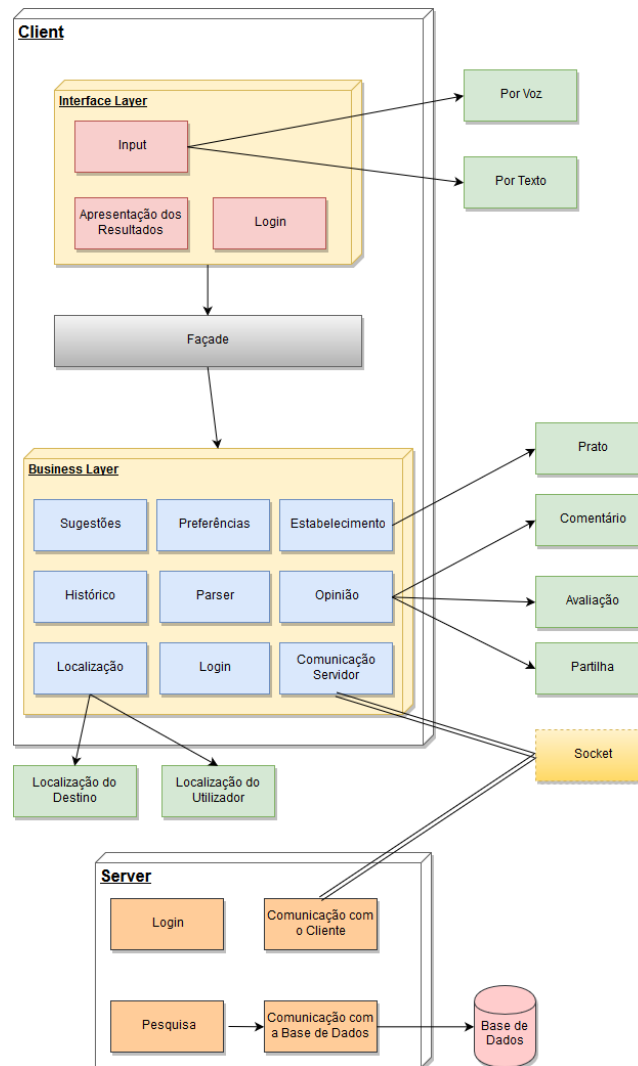


Figura 1 - Maqueta do Sistema

## 1.9. Conjunto de Medidas de Sucesso

Enumera-se, de seguida, um conjunto de medidas necessárias para o sucesso deste projeto. As pesquisas realizadas devem ser efetuadas de uma forma eficiente, em que os resultados obtidos vão de acordo com as preferências do utilizador e os locais indicados apresentam a melhor qualidade possível. A *User Interface* deve ser bastante intuitiva, simples e com um aspeto cativante. O desenvolvimento do projeto deverá ser bem planeado de modo a reduzir o número de horas gastas, bem como o custo de produção deste.

Para um bom funcionamento da aplicação devem-se realizar um conjunto de testes para verificar cada uma das suas funcionalidades em separado. Para a pesquisa, pode-se definir um domínio finito (um local específico) e verificar a validade dos dados obtidos para esse domínio. De modo a garantir a acessibilidade da *User Interface*, a aplicação deverá ser testada por um determinado conjunto de utilizadores que não tenham conhecimento da implementação desta. Para complementar esta informação, podemos realizar um questionário simples acerca da

qualidade da aplicação, facilidade de utilização, rapidez dos resultados e consistência dos mesmos.

Todos os requisitos, apresentados nas secções abaixo, devem estar bem implementados dentro do prazo previsto.

## 1.10. Planeamento do Desenvolvimento

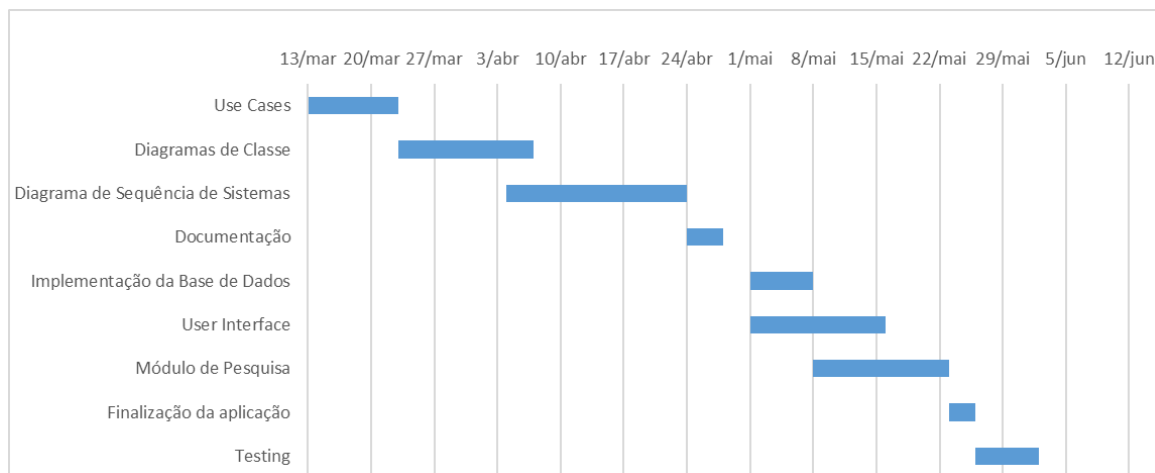


Figura 2 - Diagrama de Gantt

No diagrama acima é apresentado o planeamento previsto para as fases futuras deste projeto. Assim, a fase 2 será desenvolvida desde do dia 13 de março até ao dia 28 de abril. Por outro lado, a fase 3 do projeto terá início no dia 1 de maio e será concluída entre o dia 29 de maio e 5 de junho.

## 1.11. Levantamento de Requisitos

### 1.11.1 Pesquisa por texto

O utilizador tem a possibilidade de introduzir os parâmetros da pesquisa, manualmente, através de um teclado que lhe será apresentado (ou que esteja incorporado na máquina). Aqui, o utilizador apenas introduzirá o prato gastronómico que pretende, podendo também adicionar elementos que ache necessários. Por exemplo, pesquisar por “pastéis de nata sem sabor a limão” ou “pão de ló de ovos moles”.

### **1.11.2 Pesquisa por voz**

Com esta funcionalidade queremos promover a mobilidade do cliente da aplicação, permitindo-lhe realizar uma pesquisa através da voz. Para isso, o utilizador dispõe de um botão reservado para iniciar a pesquisa. Ou seja, ao clicar neste botão, a aplicação sabe que vai procurar pelos parâmetros que vai receber por voz.

### **1.11.3 Apresentação dos resultados**

Feita a pesquisa, a ordem de apresentação dos restaurantes conjuga vários fatores, sendo eles: o número de estrelas (avaliação), o número de comentários feitos por utilizadores e a distância entre o estabelecimento e o cliente. A relevância dos resultados surge da combinação destes três critérios, dando-se maior importância à avaliação dos mesmos. Considera-se então que a avaliação e o número de comentários terão um peso positivo na ordem da apresentação. Por outro lado, quanto maior for a distância ao local a partir da posição do cliente, maior será o impacto negativo na apresentação, logo mais baixo será o seu lugar na lista.

### **1.11.4 Filtragem de resultados**

Enquanto que os fatores descritos no ponto anterior são por defeito, o utilizador poderá refinar a apresentação de resultados, indicando novas opções de filtragem. Por exemplo, número de estrelas mínimo e/ou distância máxima até ao local.

### **1.11.5 Informações a exibir**

Para cada restaurante que se encontre na lista de resultados, serão apresentadas as seguintes informações ao cliente: nome, foto e avaliação. Desta forma, o utilizador poderá seleccionar um estabelecimento com base na sua preferência. Feita a seleção, as informações serão expandidas, passando a incluir: descrição, contactos, comentários, horário de funcionamento e o trajeto ao destino.

### **1.11.6 Histórico de pesquisa**

Cada pesquisa efetuada pelo utilizador será guardada pela aplicação, de forma a que este possa reconsultar pesquisas anteriores e utilizá-las novamente

### **1.11.7 Partilha numa rede social**

Acompanhando as tendências tecnológicas e sociais, o cliente poderá partilhar numa rede social o restaurante e o prato que vai consumir. Assim, irá manter os seus seguidores atualizados acerca das suas escolhas de degustação.

### **1.11.8 Comentários sobre o prato pesquisado**

De forma a contribuir para a avaliação de um restaurante que visitou, o utilizador poderá comentar sobre a sua experiência gastronómica naquele estabelecimento. Assim, outros clientes poderão ter uma melhor compreensão acerca desse local.

### **1.11.9 Sugestões de restaurantes a visitar**

Com o intuito de valorizar e distinguir esta aplicação em relação a outras plataformas, o sistema fornece uma lista de restaurantes atualmente mais populares na região onde o utilizador se encontra. Assim, caso ainda não saiba o que pretende degustar, o cliente poderá selecionar a opção Sugestões que lhe irá sugerir os restaurantes com melhor avaliação e que ainda não teve a oportunidade de visitar.

### **1.11.10 Tempo de resposta da aplicação**

É esperado que o tempo de resposta seja impercetível ao utilizador, uma vez que um sistema deste tipo não pode permitir esperas demoradas.

### **1.11.11 Memória ocupada pelos dados**

Importa que os dados armazenados localmente ao utilizador (*Mobile* ou *Desktop*) ocupem o mínimo possível. Isto porque não pretendemos que o dispositivo do cliente fique sobrecarregado pela aplicação.

### **1.11.12 Facilidade de utilização**

A *User Interface* deve estar preparada para qualquer tipo de utilizador, independentemente do seu nível de experiência com a plataforma que estão a utilizar.



## 2. Análise de Requisitos

### 2.1. Interface Layer

#### 2.1.1 Input

##### 2.1.1.1 Requisitos Funcionais

- **Pesquisa por texto** - Quando o utilizador pretender realizar a sua pesquisa textualmente, ser-lhe-á apresentada uma barra branca na janela da aplicação. Aqui o cliente deverá introduzir os parâmetros da sua procura. Dependendo do tipo de cliente, a introdução é feita de maneira diferente. Se a pesquisa for feita num dispositivo desktop, o teclado está naturalmente presente na máquina. Por outro lado, caso seja feita num dispositivo mobile, será utilizado o teclado que se tem por defeito nesses sistemas. Introduzidos os parâmetros, o cliente deve seleccionar o botão apropriado para que a pesquisa seja iniciada.
- **Pesquisa por voz** - Caso o utilizador não queira ou não consiga fazer a pesquisa por texto, será dada a possibilidade de indicar o que pretende através da sua voz. Para dar início à pesquisa, deverá premir o botão apropriado. Ao seleccionar este botão, a aplicação entra em modo de captura de voz, permanecendo neste estado por um período de tempo curto. Aqui, tudo que for dito ao dispositivo será convertido para texto que por sua vez é utilizado para efetuar a pesquisa. A captura de voz termina automaticamente após o utilizador não indicar nada por alguns instantes de tempo.

## 2.1.2 Apresentação dos Resultados

### 2.1.2.1 Requisitos Funcionais

- **Apresentação dos resultados** - Efetuada uma pesquisa, é apresentado ao utilizador uma lista de restaurantes que satisfazem os parâmetros introduzidos. A ordem dos resultados é influenciada pelos seguintes critérios: avaliação do estabelecimento, número de comentários feitos e distância ao local de onde foi efetuada a pesquisa.

Destes fatores, distinguem-se os que exercem uma influência positiva na ordem de apresentação: avaliação e número de comentários. Entre estes, a avaliação do estabelecimento é a que tem mais influência. Quanto maior for o número de comentários, pressupõe-se que a avaliação do estabelecimento é mais fidedigna. Assume-se que entre um restaurante com uma avaliação de 4/5 e 100 comentários e um restaurante com a mesma avaliação, mas apenas com 10 comentários, a avaliação do primeiro será a mais fiável. Por esta razão conjuga-se a avaliação com o número de comentários.

Por outro lado, quanto maior for a distância ao local pior será a sua posição na lista de resultados. Assim, evita-se que os restaurantes mais longínquos ocupem os primeiros lugares da lista.

- **Histórico de pesquisa** - Sempre que um utilizador selecionar a aba histórico, ser-lhe-á apresentada uma lista de pesquisas ordenadas da mais recente para a mais antiga. Ao selecionar uma pesquisa já feita, o cliente é encaminhado para o respetivo estabelecimento. Devido ao número elevado de procuras que poderão ocorrer, o utilizador terá de fazer *scroll down* de forma a visualizar o histórico mais antigo. Do histórico podem ser eliminados estabelecimentos que foram visitados pelo utilizador.
- **Sugestões de restaurantes a visitar** - É dado ao utilizador a hipótese de ser a aplicação a sugerir-lhe o próximo estabelecimento que deverá visitar. A opção *sugestões* apresenta os cinco melhores estabelecimentos que este ainda não tenha visitado. A ordem de apresentação é por avaliação.
- **Filtragem de resultados** - Apresentados os resultados da pesquisa, o utilizador tem a possibilidade de refiná-los, selecionando as opções de interesse. Para tal, o cliente tem de selecionar um botão que vai apresentar-lhe um menu para definir as opções do refinamento.

O aperfeiçoamento dos resultados pode ser feito na distância máxima ao local, na avaliação mínima do estabelecimento e/ou se encontra em horário de funcionamento no momento. Introduzidos os campos de personalização da pesquisa, o utilizador

seleciona o botão designado para registá-los. Logo de seguida, são apresentados os resultados que cumprem os fatores pretendidos.

- **Informações a exibir** - A lista dos resultados contém informações mais genéricas para cada um dos restaurantes. Aqui apresenta-se: nome, foto e avaliação. Consideram-se estas as informações mais relevantes para que o utilizador possa formar uma primeira impressão sobre o estabelecimento.

Selecionado o restaurante, acrescentam-se às informações iniciais: descrição do estabelecimento, contactos, *website*, horário de funcionamento, comentários e trajeto ao destino.

Quanto aos comentários, o utilizador poderá navegar sobre a lista apresentada. O local é apresentado numa janela da *Windows Maps*, em que ao selecioná-la é calculado o percurso ao destino.

### 2.1.2.2 Requisitos Não Funcionais

- **Facilidade de utilização** - A *User Interface* é intuitiva ao ponto de ser possível a um utilizador, sem experiência na área da informática, conseguir efetuar uma pesquisa, navegar e refinar os resultados, selecionar um estabelecimento e inclusive partilhar numa rede social.

## 2.2. Business Layer

### 2.2.1 Login

#### 2.2.1.1 Requisitos Funcionais

- **Login no sistema** - O Cliente pode registar-se na aplicação ou, caso já o tenha feito, efetuar o *login*. Isto serve para que o Cliente possa trocar entre dispositivos (*mobile* ou *desktop*), sem perder as suas preferências de pesquisa ou histórico, como também não permitir que avalie/comente o mesmo estabelecimento mais que uma vez. Para que uma pesquisa possa ser feita, o Cliente tem que estar autenticado.

## 2.2.2 Opinião

### 2.2.2.1 Requisitos Funcionais

- **Partilha numa rede social** - No *Chewie*, selecionado um estabelecimento, o utilizador pode escolher a opção de partilha, sendo-lhe fornecido um conjunto de redes sociais disponíveis. Depois, este é redirecionado para a aplicação respetiva da rede social ou para o *browser*.
- **Comentários sobre o estabelecimento visitado** - Sempre que um utilizador selecionar um estabelecimento, é-lhe dada a opção de efetuar um comentário sobre o prato gastronómico e/ou estabelecimento. É apresentada uma janela ao utilizador para que possa inserir o seu comentário. Um utilizador apenas pode comentar um estabelecimento uma única vez.
- **Avaliação sobre o estabelecimento visitado** - Sempre que um utilizador selecionar um estabelecimento, é-lhe dada a opção de avaliar o estabelecimento visitado. A avaliação é um número inteiro compreendido entre 1 e 5, que é guardada diretamente na Base de Dados da aplicação. Uma janela é apresentada para que o utilizador faça a sua avaliação, podendo apenas dar uma única avaliação a um estabelecimento.

## 2.2.3 Preferências

### 2.2.3.1 Requisitos Funcionais

- **Informação para as preferências do utilizador** - Um utilizador indica numa pesquisa o prato pretendido e, caso queira, um conjunto de parâmetros adicionais que representam o seu gosto pessoal. Ao indicar que visitou um estabelecimento, é guardada na Base de Dados, para aquele prato, os parâmetros adicionais que inseriu anteriormente. Estes representam as suas preferências, no que diz respeito ao prato pesquisado. Quanto maior o número de vezes que pesquisar o mesmo prato com esses parâmetros, maior será a sua relevância numa pesquisa futura, para aquele prato.

## 2.2.4 Histórico

### 2.2.4.1 Requisitos Funcionais

- **Histórico de Pesquisa** - Na aplicação existe uma aba chamada histórico que armazena todas as pesquisas efetuadas. Para cada utilizador são guardados os estabelecimentos que este indica como visitados, como também a data.

## 2.3. Servidor

### 2.3.1 Pesquisa

#### 2.3.1.1 Requisitos Não Funcionais

- **Tempo de resposta da aplicação** - Pretende-se que o tempo de resposta da aplicação seja o mais curto possível, uma vez que é um fator de grande importância na hora da refeição. Assim, a aplicação deverá apresentar uma lista de resultados num período de tempo inferior a cinco segundos. Desta forma, o tempo de resposta não interfere de maneira negativa na experiência que o utilizador tem com a aplicação.
- **Limitação da região** - A pesquisa de pratos gastronómicos é limitada à região de Braga.

### 2.3.2 Base de Dados

#### 2.3.2.1 Requisitos Não Funcionais

- **Memória ocupada pelos dados** - A aplicação tem uma arquitetura Cliente-Servidor. O espaço ocupado pelos dados no Cliente é mínimo, isto porque o volume de informação a ser acedido pela aplicação é muito elevado, ou seja, o Servidor é quem guarda e tem acesso direto aos dados. O Cliente terá guardado o seu *username* e a respetiva *password*, de forma a poder ser identificado no momento de *login*.



*Preferência*. Quando for feita a escolha de um *Estabelecimento*, é necessário determinar a trajetória entre a sua *Morada* e a *Localização* do *Utilizador*. Contudo, é possível refinar os resultados da *Pesquisa*, aplicando-lhe um *Filtro* que vai desde a *Distância Máxima*, à *Avaliação Mínima* ou até ao *Horário de Funcionamento*. O *Utilizador* também pode efetuar um *Comentário* e uma *Avaliação* sobre o *Estabelecimento* que visitou.

Relativamente ao *Estabelecimento*, têm-se as seguintes informações: *Nome*, *Foto*, *Morada*, *Descrição*, *Horário*, *Ementa* e *Contacto*. Este, por sua vez, pode ser um *E-Mail*, *Website* ou *Telefone*.

A *Ementa* de um *Estabelecimento* é um conjunto de *Prato*, onde para cada um destes existe um *Detalhe*. A *Pesquisa* efetuada pelo *Utilizador* devolve um *Resultado* formado por todos aqueles *Estabelecimento* que têm na sua *Ementa* pelo menos um *Prato* cujo *Detalhe* está de acordo com a *Preferência* do *Utilizador*.

## 3.2. Diagrama de Use Cases

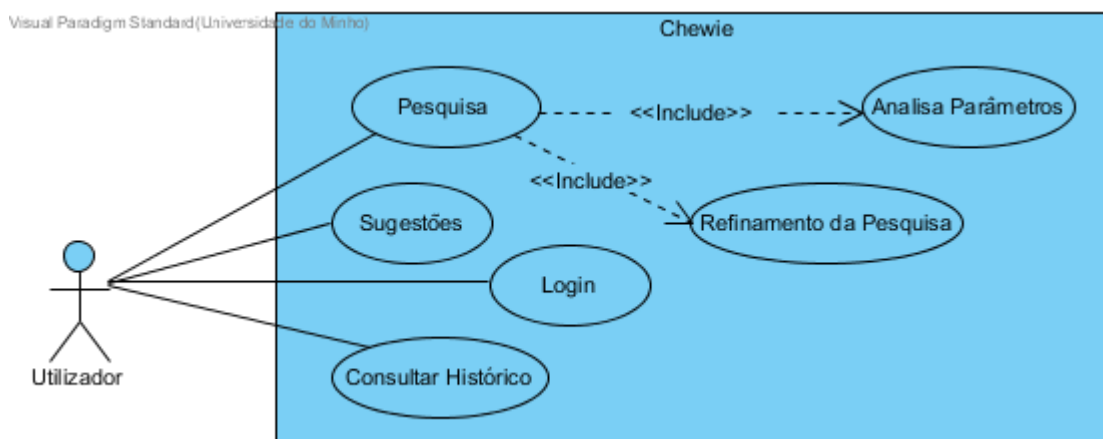


Figura 4 - Diagrama de Use Cases

O utilizador pode interagir de quatro maneiras diferentes com o sistema. Isto é, tem a opção de efetuar uma pesquisa sendo a funcionalidade principal do programa. Tem ainda a oportunidade de consultar o histórico das pesquisas efetuadas sempre que lhe for conveniente. Não apenas consultar mas tem a possibilidade de também o eliminar. Além disso, o utilizador pode visualizar sugestões automáticas, determinadas pelo programa, de estabelecimentos que ainda não tenha visitado e cumpram os requisitos mínimos estabelecidos pelo grupo. Finalizando, para poder efetuar todas estas ações com o programa este tem de iniciar sessão através de um sistema de *login*.

Denote-se que não existe a necessidade de um administrador ao nível da aplicação cliente tendo em conta que esta é local ao seu dispositivo. E, além disso, o único nível de administração é ao nível da aplicação do lado do servidor, contabilizando com a base de

dados. Por fim, o ato de atualizar a aplicação é algo externo ao próprio programa. Poder-se-ia ainda referir que o resultado dessa atualização poderá ser um programa muito diferente com novas *features* e, portanto, novos *Use Cases*.

### 3.3. Especificação de Use Cases

#### 3.3.1 Login

Tabela 1 - Use Case de Login

Name		Login	
Brief Description		O utilizador efetua a autenticação no sistema	
Preconditions			
Post-conditions		Login efetuado	
		Actor Input	System Response
Flow of Events	1	Indica que quer iniciar sessão	
	2	Introduz username e password	
	3		Valida username e password
	4		Indica que o utilizador está autenticado
		Actor Input	System Response
Comportamento Alternativo (passo 1) [utilizador pretende registar-se]			
	2.1	Indica que pretende registar-se	
	2.2	Introduz username e password	
	2.3		Valida username e password
	2.4		Regista username e password
	2.5		Indica que o utilizador foi registado
	2.6		Regressa a 4
Exceção 1 (passo 2) [dados de autenticação inválidos]	1		Indica que os dados de autenticação são inválidos
Exceção 2 (passo 2.3) [formato de username ou password inválido]	1		Indica que os dados de username ou password são inválidos



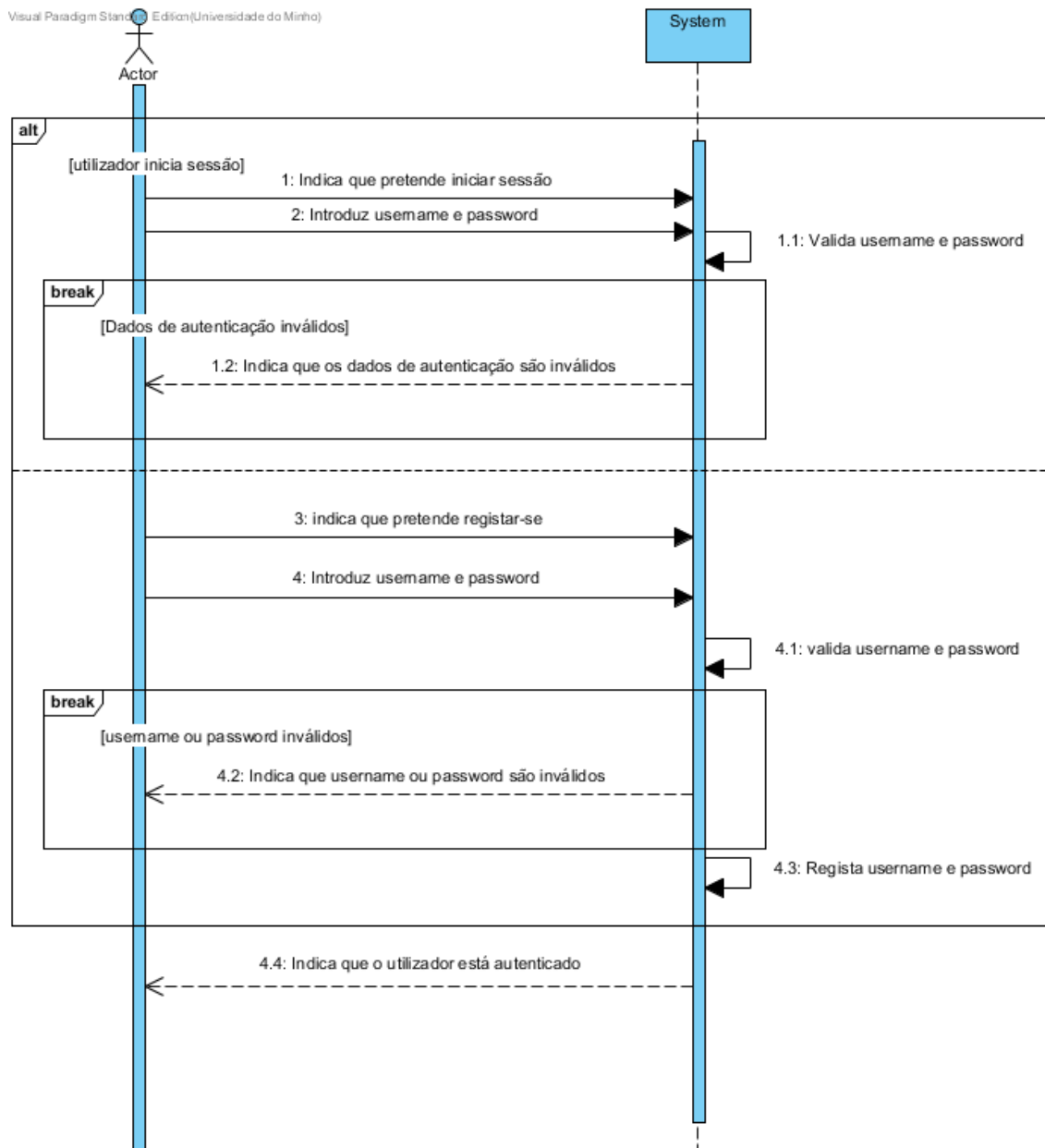


Figura 5 - Diagrama de Sequência do Login

O utilizador é convidado a efetuar um novo registo sempre que lhe for conveniente, tendo em conta o formato admissível para o username e password. Assim, ao abrir a aplicação, este pode iniciar sessão ou registar uma nova conta. Feito isto, o utilizador fica automaticamente autenticado e é redirecionado para a janela inicial.

### 3.3.2 Pesquisa

Este *Use Case* (**Anexo 1** e **Anexo 2**) especifica o processo entre o utilizador e o sistema quando se efetua uma pesquisa de um prato gastronómico. Deste modo, o cliente

pode introduzir os parâmetros por voz ou por texto, dependendo da sua preferência. O sistema tratará de fazer as verificações e análise necessárias. De seguida, com os termos-chave retirados, fará uma pesquisa na Base de Dados. A procura é feita na tabela Prato que retornará os nomes dos pratos cuja primeira palavra coincida com a primeira palavra-chave. De seguida, são filtrados os pratos que contenham os restantes termos-chave. Ainda é feita uma filtragem de acordo com os detalhes do prato e as preferências do utilizador. Por fim, devolvem-se todos os estabelecimentos que possuam esses pratos.

O cliente pode navegar pela lista de resultados, selecionando um estabelecimento que tenha interesse em visitar, assim como as informações acerca do restaurante e o trajeto até lá, se pretender. Por outro lado, ainda pode refinar os resultados, de modo a modificar a distância máxima de procura, o número mínimo de avaliações e o horário de funcionamento desejado.

Caso o utilizador indique que visitou o estabelecimento, tanto o prato pesquisado como o estabelecimento visitado serão guardados no seu histórico. Se os parâmetros da pesquisa contiverem alguma preferência do utilizador, como, por exemplo, "sem molho de tomate", esta também será guardada nas suas preferências.

De modo a aumentar a interatividade dos utilizadores, estes podem comentar o estabelecimento que visitaram, bem como avaliá-lo e partilhar a sua experiência numa rede social à sua escolha.

### 3.3.3 Analisa Parâmetros

Tabela 2 - Use Case da Análise de Parâmetros

Name		Analisa Parâmetros	
<b>Brief Description</b>		Retira os termos-chave de uma pesquisa	
<b>Preconditions</b>		Parâmetros introduzidos e validados	
<b>Post-conditions</b>		Termos-chave retirados	
		Actor Input	System Response
<b>Flow of Events</b>	1		Recebe parâmetros da pesquisa validados
	2		Guarda os nomes comuns
	3		Guarda os adjetivos
	4		Guarda as preposições "com", "sem", "exceto"
	5		Guarda os advérbios de modo "bem", "mal"
	6		Guarda os advérbios de quantidade "muito", "pouco", "mais", "menos"
	7		Guarda os advérbios de negação "não", "nem"
	8		Guarda o advérbio de exclusão "só"
	9		Devolve os termos-chave guardados

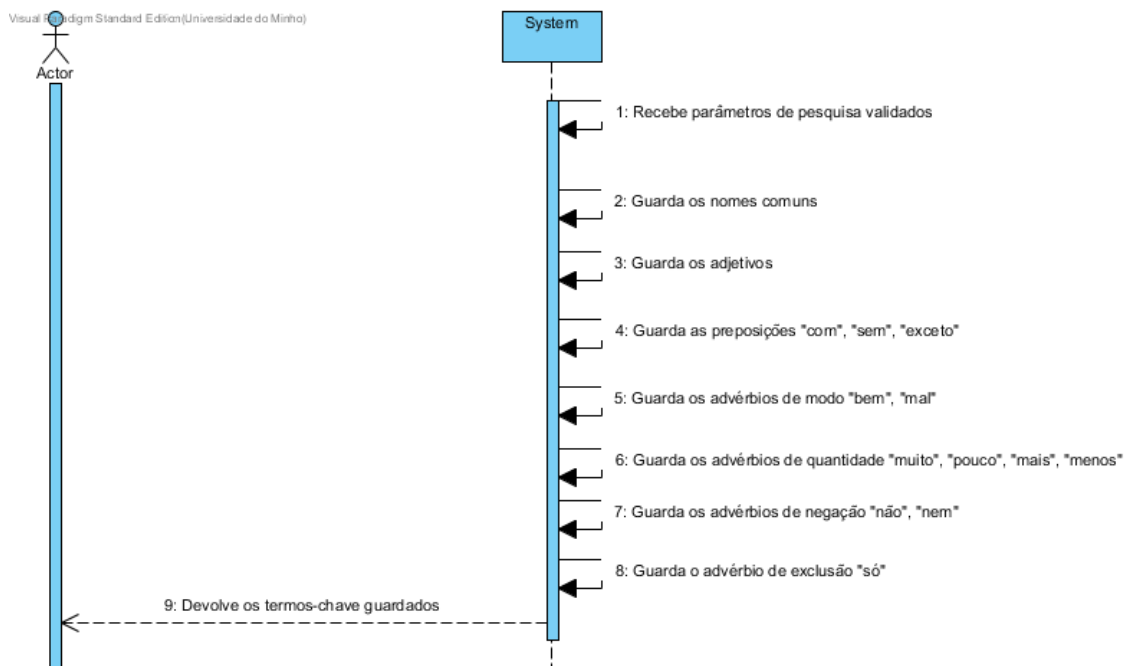


Figura 6 - Diagrama de Sequência da Análise de Parâmetros

Este *Use Case* é incluído na pesquisa de um prato gastronómico quando o utilizador introduz a procura. Desta forma, o sistema recebe estes parâmetros e analisa-os. Durante este processo, os nomes comuns (como, por exemplo, “massa”, “bife”, “queijo”) são guardados, pois fazem parte do conjunto de termos-chave que serão importantes para a pesquisa na Base de Dados. De seguida, guardam-se igualmente os adjetivos, assim como alguns advérbios que se encontram em muitas das pesquisas feitas que incluem também preferências do utilizador, por exemplo, em “arroz de pato com alecrim” temos a preferência “com alecrim”, logo o advérbio “com” é guardado e considerado um termo-chave (ao contrário de “de” que é descartado). Deste modo, os termos-chave finais seriam “arroz pato com alecrim”.

### 3.3.4 Histórico

Tabela 3 - Use Case do Histórico

<b>Name</b>		Histórico	
<b>Brief Description</b>		O utilizador consulta pesquisas anteriores	
<b>Preconditions</b>		Utilizador autenticado	
<b>Post-conditions</b>		Pesquisas anteriores são apresentadas	
		<b>Actor Input</b>	<b>System Response</b>
<b>Flow of Events</b>	1	Utilizador indica que pretende ver o histórico	
	2		Procura histórico do utilizador
	3		Ordena resultados por data de pesquisa mais recente
	4		Apresentação dos resultados
	5	Selecciona uma pesquisa anterior	
	6		Apresenta as informações do estabelecimento
	7		Verifica a localização do utilizador
	8		Verifica a localização do destino
	9		Apresenta o trajeto até ao local
	10	Volta ao menu inicial	
<b>Comportamento Alternativo (passo 5) [utilizador pretende eliminar uma pesquisa]</b>			
	5.1	Selecciona pesquisa	
	5.2	Indica que quer eliminar pesquisa seleccionada	
	5.3		Elimina pesquisa do histórico
	5.4		Indica que a pesquisa foi eliminada
	5.5	Regressa a 5	
<b>Comportamento Alternativo (passo 5) [utilizador não selecciona uma pesquisa]</b>			
	5.1	Regressa a 10	
<b>Excepção 1 (passo 2) [utilizador não tem pesquisas anteriores]</b>		<b>Actor Input</b>	<b>System Response</b>
	2.1		Indica que o utilizador não tem pesquisas anteriores

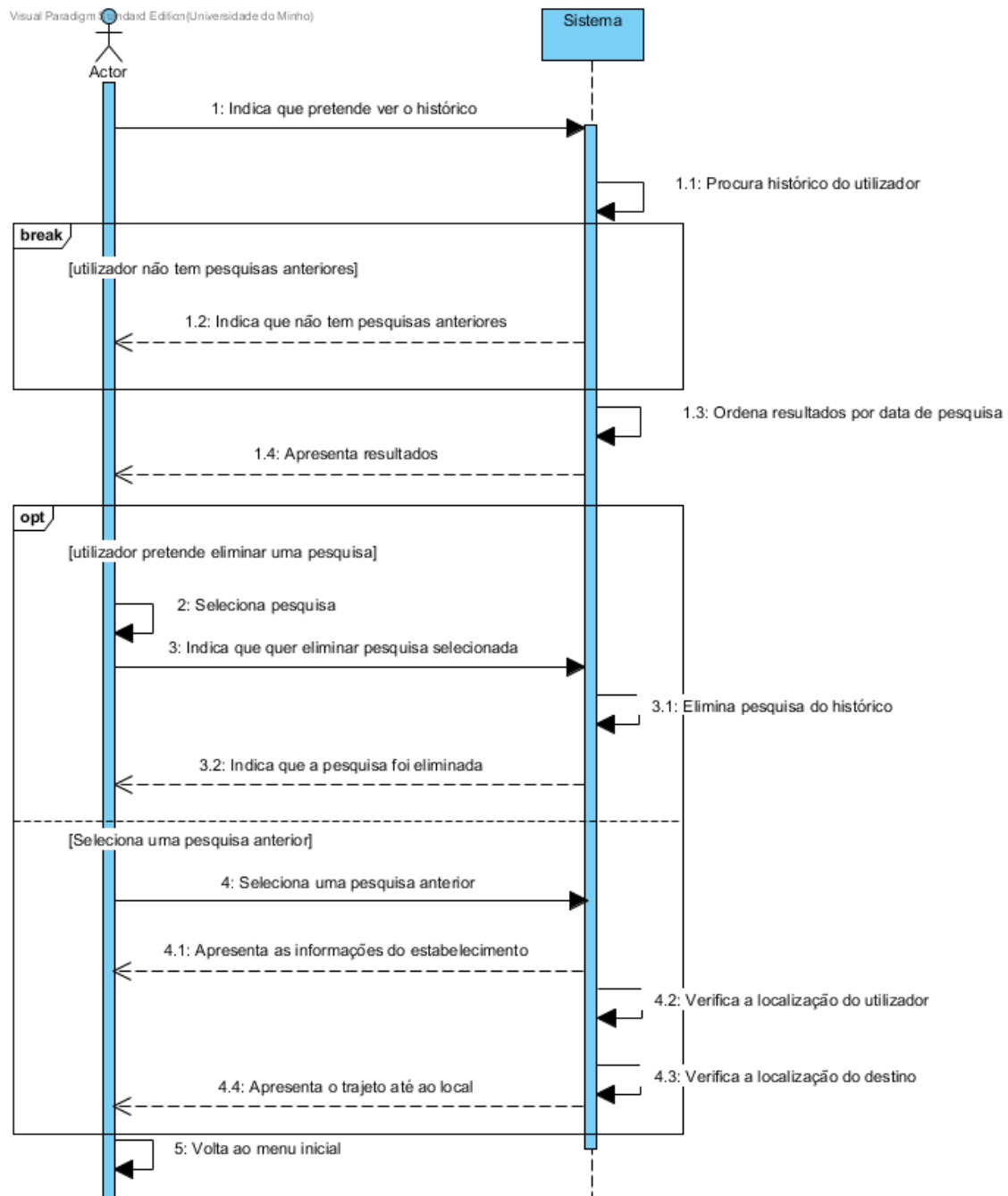


Figura 7 - Diagrama de Sequência do Histórico

O utilizador tem a opção de consultar o histórico. Como observamos, o histórico estará sempre organizado pela data do mais recente para o mais antigo. Sempre que este seleccionar uma entrada no histórico, toda a informação do estabelecimento é-lhe imediatamente fornecida. Da mesma forma, opcionalmente, o utilizador poderá pedir para ter a entrada no histórico apagada. Por fim, caso o utilizador ainda não tenha feito pesquisas ou tenha eliminado o histórico e o pretenda visualizar, ser-lhe-á indicado que não existem pesquisas.

### 3.3.5 Sugestões

Tabela 4 - Use Case das Sugestões

Name		Sugestões	
Brief Description		São apresentadas sugestões de estabelecimentos a serem visitados	
Preconditions		Utilizador autenticado	
Post-conditions		Sugestões são apresentadas ao utilizador	
		<b>Actor Input</b>	<b>System Response</b>
Flow of Events	1	Utilizador indica que pretende ver a lista de sugestões	
	2		Verifica todos os locais que o utilizador ainda não visitou
	3		Verifica os locais que têm uma valiação mínima de 3
	4		Ordena os locais por avaliação
	5		Selecciona os 5 melhores
	6		Apresenta os resultados
	7	Volta ao menu inicial	
Comportamento Alternativo (passo 5) [o número de resultados é inferior a 5]			
	5.1		Selecciona os resultados disponíveis
	5.2		Regressa a 6
Comportamento Alternativo (passo 7) [utilizador selecciona um dos estabelecimentos sugeridos]			
	7.1	Selecciona um estabelecimento	
	7.2		Apresenta as informações do estabelecimento
	7.3		Verifica a localização do utilizador
	7.4		Verifica a localização do destino
	7.5		Apresenta o trajeto até ao local
	7.6		Regressa a 7
Excepção 1 (passo 3) [não existem estabelecimentos que cumprem o requisito mínimo]		<b>Actor Input</b>	<b>System Response</b>
	3.1		Indica que não existem estabelecimentos a serem sugeridos

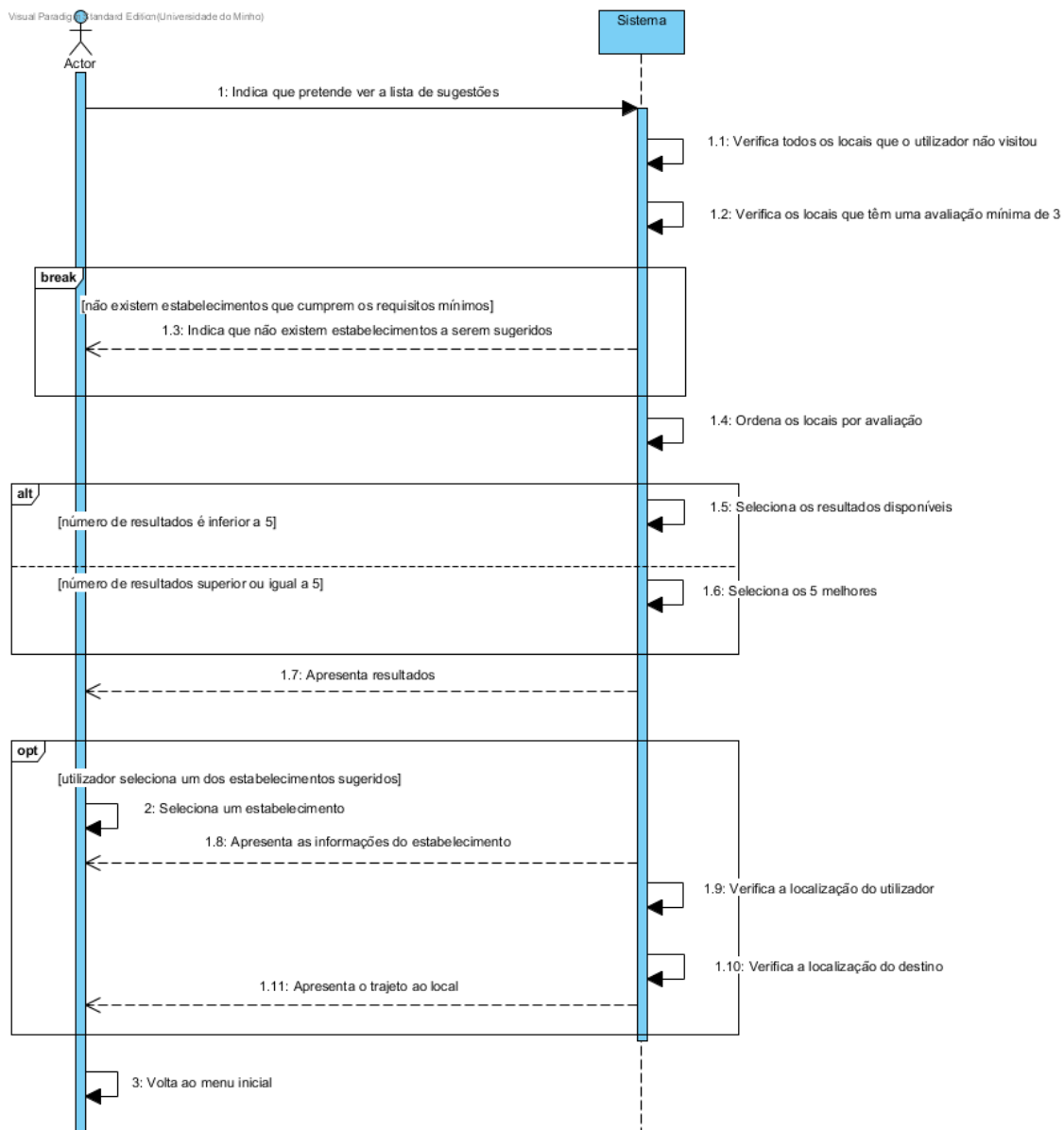


Figura 8 - Diagrama de Sequência de Sugestões

O *Use Case* “Sugestões” aplica-se quando o utilizador pretende ver a lista de estabelecimentos que lhe são sugeridos. Primeiramente, o sistema trata de verificar todos os restaurantes que o utilizador ainda não visitou, de seguida filtra os estabelecimentos que cumprem o requisito mínimo imposto: ter uma avaliação mínima de três, uma vez que o objetivo é sugerir ao utilizador restaurantes que tenham uma boa reputação. Desta maneira, são apresentados os cinco melhores estabelecimentos que resultaram deste filtro.

Caso o utilizador tenha interesse em visitar um dos restaurantes sugeridos, são apresentadas as informações e o trajeto até ao local.

### 3.3.6 Refinamento da Pesquisa

Tabela 5 - Use Case do Refinamento da Pesquisa

<b>Name</b>		Refinamento da Pesquisa	
<b>Brief Description</b>		Utilizador redefine condições da pesquisa	
<b>Preconditions</b>		Pesquisa efetuada	
<b>Post-conditions</b>		Resultados refinados	
		<b>Actor Input</b>	<b>System Response</b>
<b>Flow of Events</b>	1	Indica que pretende refinar a pesquisa	
	2		Apresenta opções de refinamento
	3	Redefine a distância máxima ao local	
	4	Redefine a avaliação mínima do local	
	5	Indica o horário de funcionamento pretendido	
	6		Valida a distância máxima
	7		Valida a avaliação mínima
	8		Refina os resultados com os novos parâmetros
	9		Apresenta os novos resultados
<b>Exceção 1</b> (passo 6) [distância máxima inválida]		<b>Actor Input</b>	<b>System Response</b>
	6.1		Indica que a distância máxima pretendida não é válida
<b>Exceção 2</b> (passo 7) [avaliação mínima inválida]		<b>Actor Input</b>	<b>System Response</b>
	7.1		Indica que a avaliação mínima pretendida não é válida
<b>Exceção 3</b> (passo 9) [não há resultados a apresentar]		<b>Actor Input</b>	<b>System Response</b>
	10.1		Indica que não existem resultados a apresentar



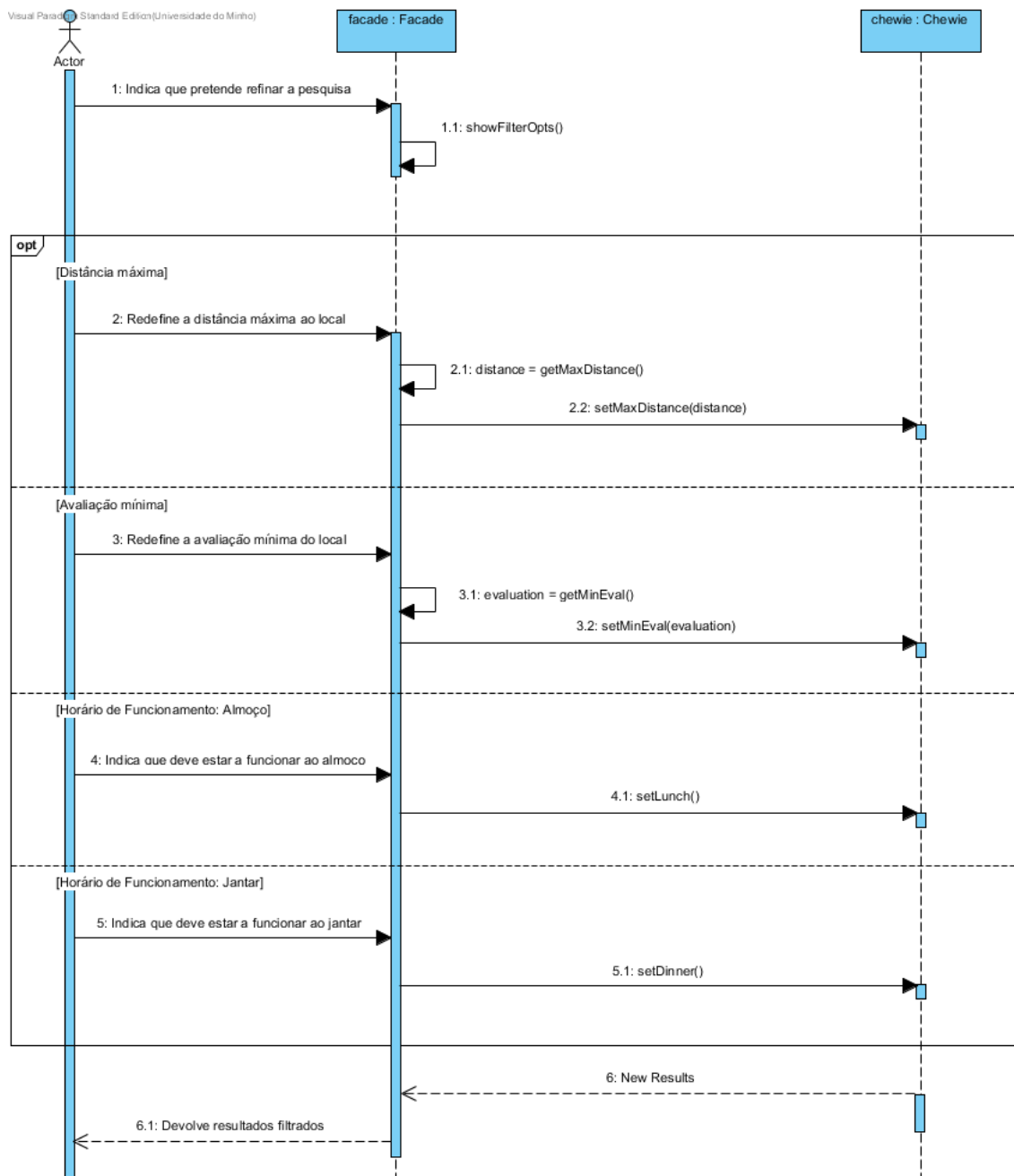


Figura 9 - Diagrama de Sequência do Refinamento da Procura

Como já foi mencionado, quando a lista de restaurantes é apresentada ao utilizador, este pode refiná-los. Para isso, basta indicar que pretende refinar a pesquisa, que lhe serão, de imediato, mostradas as opções de filtragem. Assim, poderá redefinir a distância máxima ao local, assim como a avaliação mínima e o horário de funcionamento pretendido.

O sistema trata de filtrar a lista de resultados com os novos parâmetros inseridos. Qualquer estabelecimento que não cumpra uma destas opções será eliminado da lista e não é apresentado ao cliente.

### 3.3.7 Input por Voz

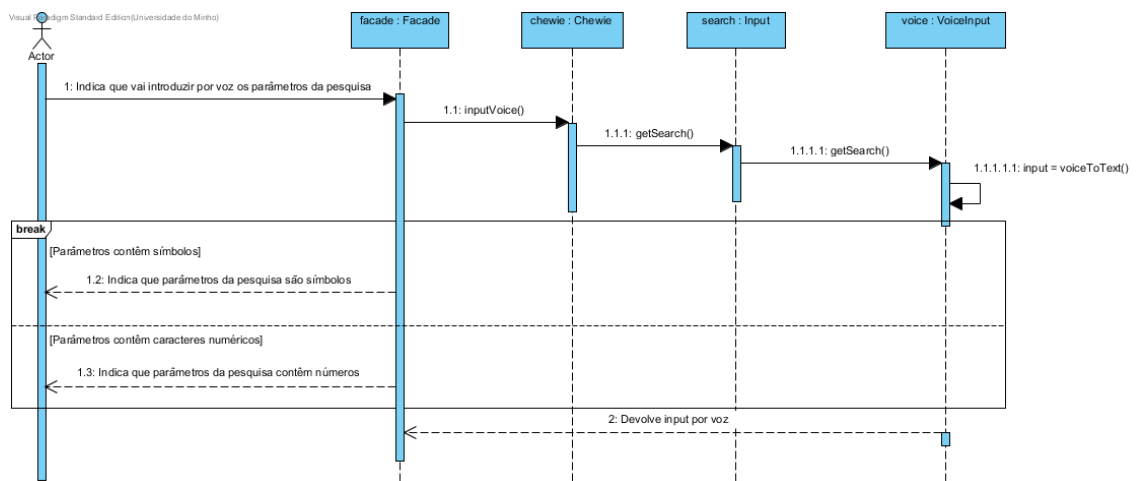


Figura 10 - Diagrama de Sequência do Input por Voz

### 3.3.8 Input por Texto

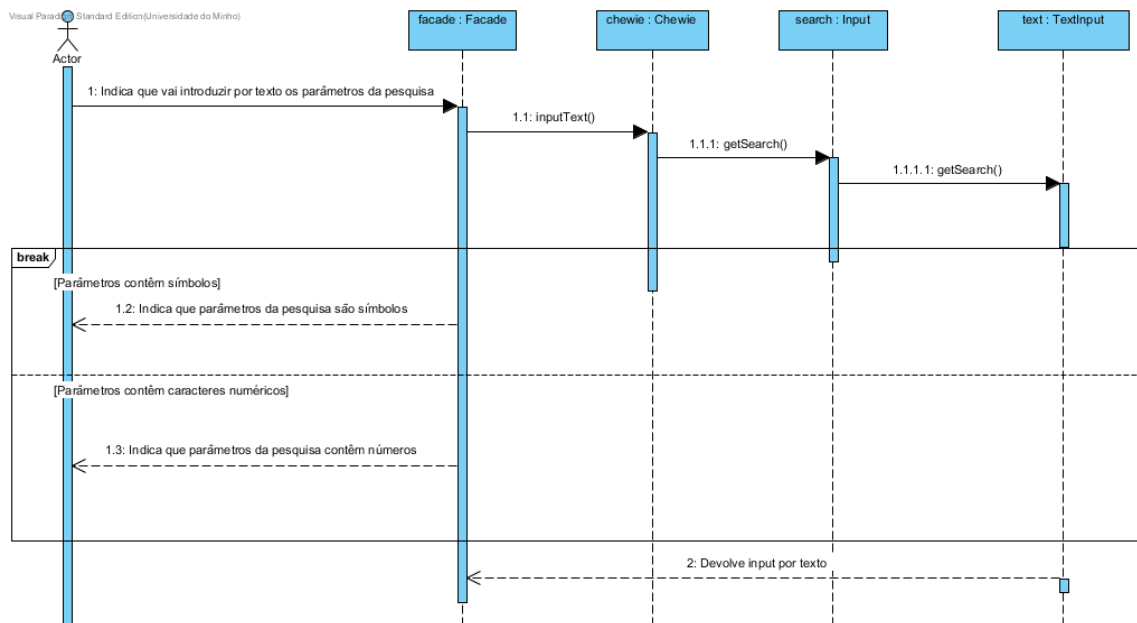


Figura 11 - Diagrama de Sequência de Input por Texto

### 3.3.9 Comentar Estabelecimento

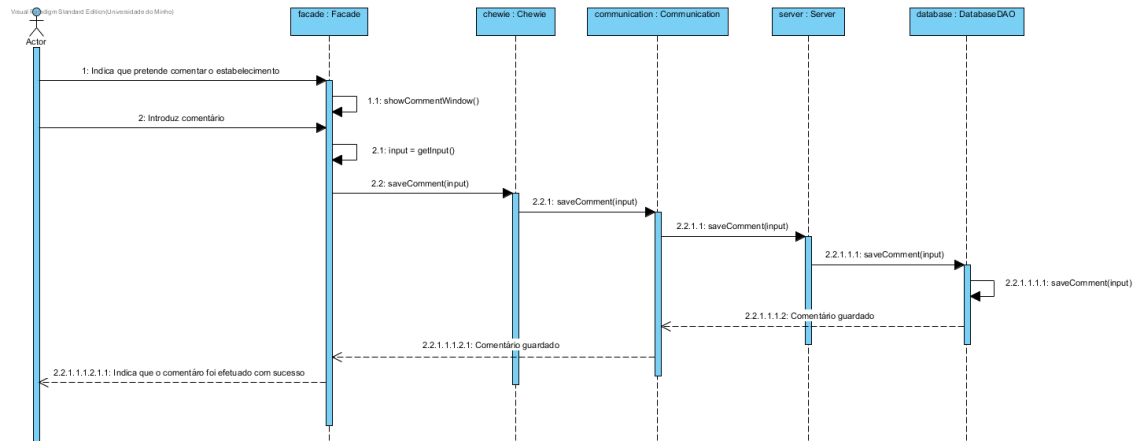


Figura 12 - Diagrama de Sequência de Comentar Estabelecimento

### 3.3.10 Avaliar Estabelecimento

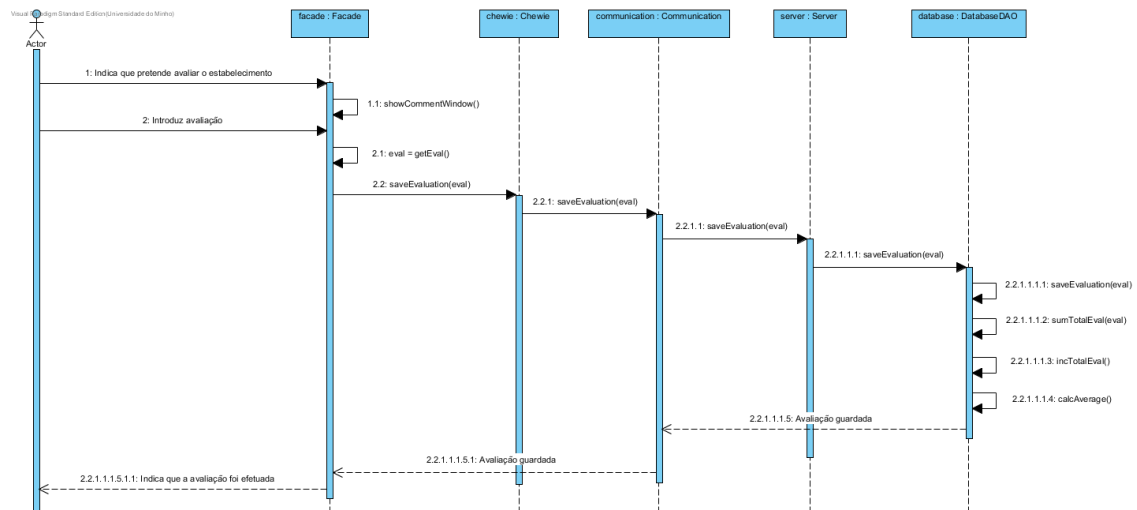


Figura 13 - Diagrama de Sequência de Avaliar Estabelecimento

### 3.3.11 Partilha na Rede Social

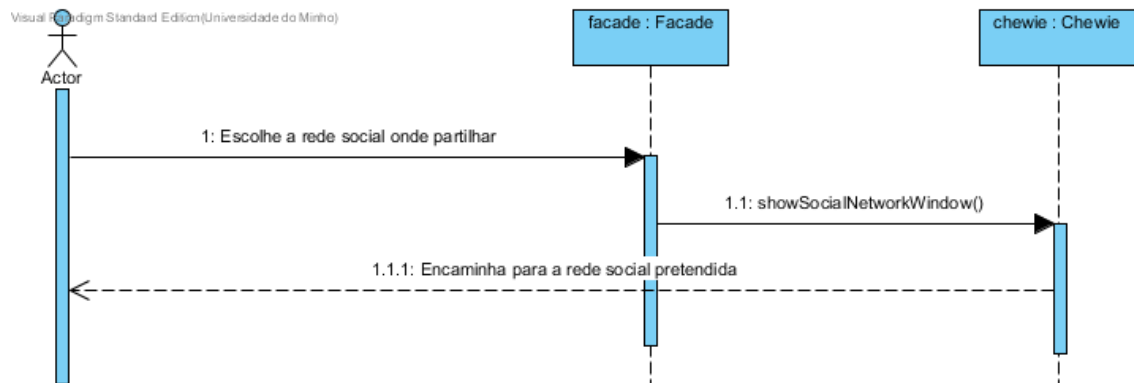


Figura 14 - Diagrama de Sequência da Partilha na Rede Social

## 3.4. Diagrama de Classes

Numa tentativa de aproximar o diagrama de classes o mais próximo ao *software* final, o grupo tentou definir todas as variáveis de instância necessárias ao correto funcionamento do *Chewie*, assim como todos os métodos necessários. Por outro lado, num cenário realista, também acordamos ter de fazer eventuais intervenções na terceira fase do projecto.

### 3.4.1 Cliente

Visual Paradigm Standard Edition (Universidade do Minho)

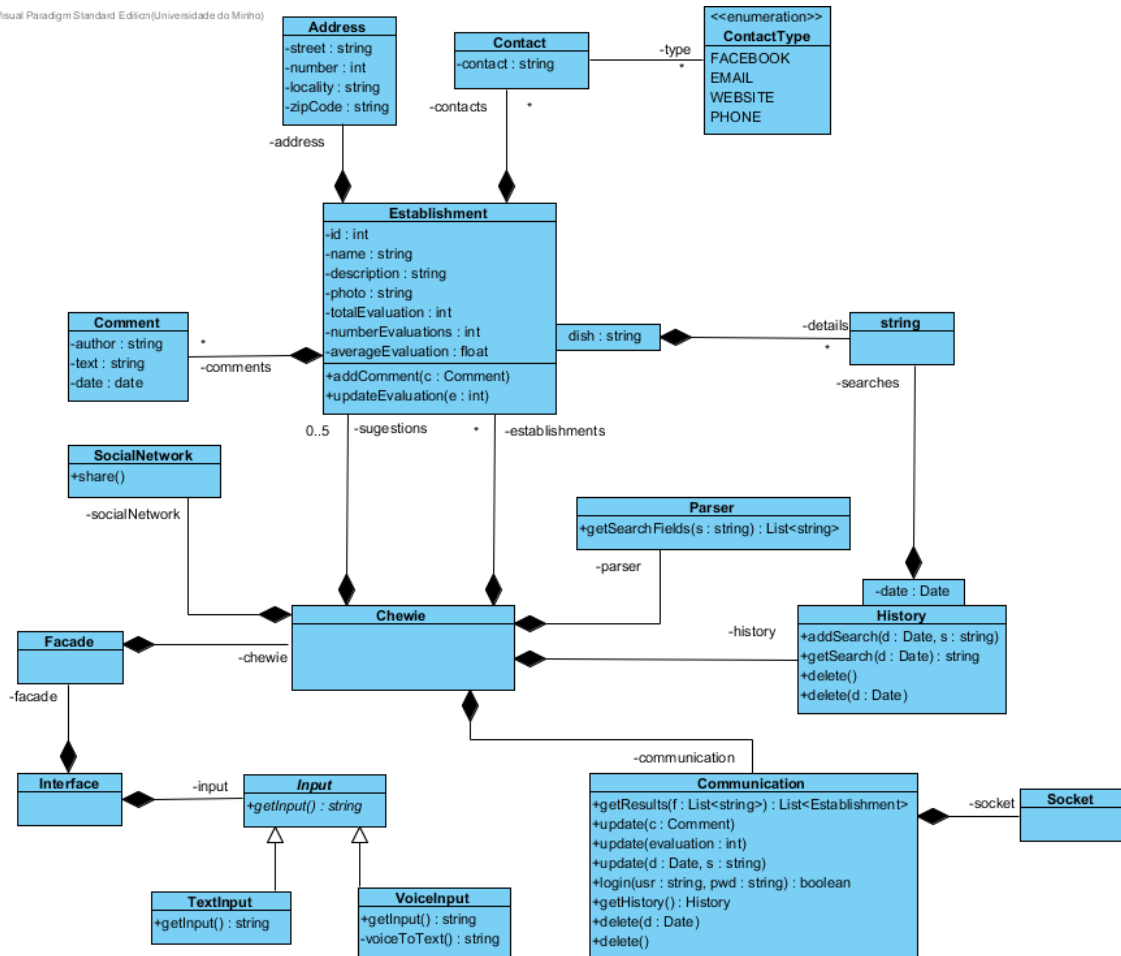


Figura 15 - Diagrama de Classes do Cliente

O estabelecimento é representado por uma classe *Establishment* que aglomera todos os objetos essenciais para a distinguir e representar toda a sua informação. Assim, esta apresenta uma *List<Comment>* definindo cada comentário efetuado para esse estabelecimento. Também, uma classe *Address*, para definir o endereço do estabelecimento. Da mesma forma, *Establishment* apresenta uma *List<Contact>* pois esta pode ter um determinado conjunto de contactos disponíveis. Por fim, *Establishment* deve conter um *Map<String, List<String>>*. A *Key* representa o prato e o *Value* todos os seus detalhes.

A classe que aglomera todos os objetos essenciais é a *Chewie*. Esta além de uma *List<Establishment>* para identificar todos os estabelecimentos resultados da pesquisa. Conta ainda com outra variável para identificar os melhores estabelecimentos que ainda não tenham sido visitados pelo utilizador que se encontra com a sessão iniciada no momento. Para tal efeito é necessária outra *List<Establishment>*. Outro fator importante reside no facto de *Chewie* também conter um objeto do tipo *History*. Este, por sua vez, representa o histórico do utilizador que deve ser atualizado após enviar a alteração à base de dados. O histórico é carregado apenas uma vez da base de dados e é atualizado conforme forem efetuadas pesquisas. Assim, *History* necessita ter um *Map<Date, String>* tal que *Date* representa a data da pesquisa efetuada (sendo esta do tipo MM:HH:DD:MM:AA). Além disso, *Chewie* apresenta um objeto do tipo *SocialNetwork* que se encarrega de partilhar conteúdo numa rede social. Ainda, *Chewie* conta com um objeto do tipo *Parser* que tal como sugere, encarrega-se de efetuar o *parsing* a uma string utilizando o método disponível. Por fim, *Chewie* necessita da classe *Communication*

responsável por fazer toda a comunicação com o servidor, para tal, apresenta todos os métodos relevantes para poder abranger todas as situações. Ou seja, apresenta métodos para obter resultados da parte do servidor assim como atualizar a informação ou mesmo remover entradas no histórico de um utilizador. Esta classe conta ainda com um objeto *Socket* de forma a poder comunicar-se com o servidor.

Por fim, a classe *Facade* é responsável por abstrair a implementação de *Chewie* tendo portanto, um objeto desta. Por sua vez, a interface poderá aceder livremente à *Facade* e conta ainda com a classe *Input*. Esta, por sua vez, encarrega-se de obter o input do utilizador quer por texto quer por voz. Assim, justifica-se estar presente na *Interface*.

### 3.4.2 Servidor

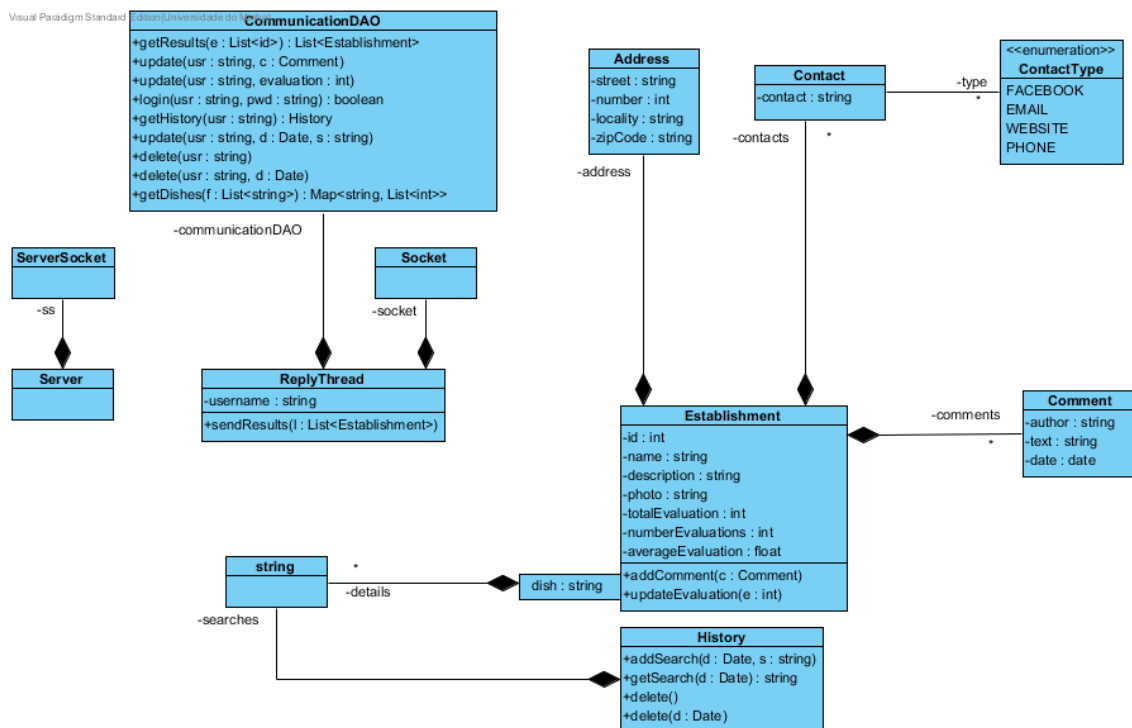


Figura 16 - Diagrama de Classes do Servidor

O Servidor tem como propósito atender a pedidos de vários Clientes concorrentemente. Assim, sempre que a classe *Server* recebe uma nova conexão cria uma nova *ReplyThread* que receberá no construtor o *socket* adequado. Esta, por sua vez, é responsável por tratar de toda a comunicação com o cliente. Tem ainda presente um objeto do tipo *CommunicationDAO* que trata de toda a comunicação com a base de dados, assim como instanciar todos os objetos necessários ao correto funcionamento do programa.

## 4. Projeto do Sistema de Dados

### 4.1. Modelo Lógico

A aplicação *Chewie* utiliza uma Base de Dados relacional, implementada em *SQL Server*. Com os diagramas realizados em *UML* e levantamento de requisitos, chegámos a um Modelo Conceptual de Base de Dados que resultou no Modelo Lógico presente na **Figura 17**.

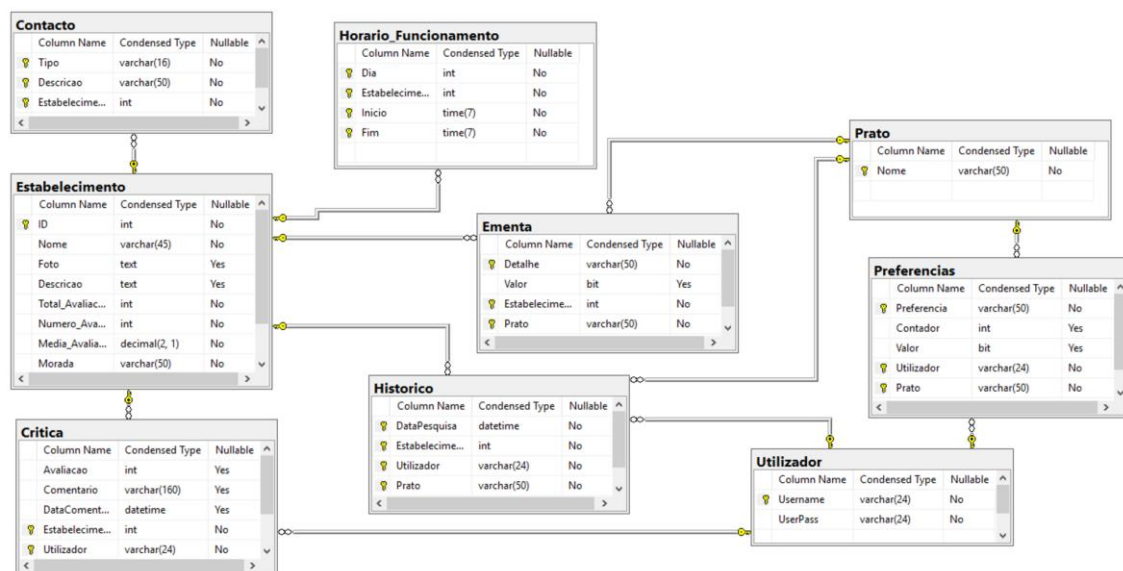


Figura 17 - Modelo Lógico

#### 4.1.1 Tabela Utilizador

Aqui temos as informações básicas de um utilizador, de forma a que este possa efetuar o *login* no sistema através de um *username* e de uma *password*. Assim, o *username* identifica-o unicamente na tabela, e, quando necessário, noutras tabelas.

### 4.1.2 Tabela Preferências

Onde as preferências que os utilizadores têm para os pratos são guardadas. Esta tabela relaciona o utilizador ao prato, quando efetua uma pesquisa. Quanto maior o número de vezes que o mesmo prato for pesquisado com a mesma preferência, maior será o valor do atributo *Contador* (sendo incrementada em um valor por cada ocorrência). Desta forma podemos ter diferentes prioridades nas preferências do mesmo prato. Da mesma forma, na tabela das preferências está presente um valor *booleano* que representa a valorização que esse detalhe tem, ou seja, se é algo positivo ou negativo (por exemplo, se tem ou não tem um ingrediente, respetivamente).

### 4.1.3 Tabela Histórico

Quando um utilizador indica que visitou um estabelecimento, é guardada na mesma linha desta tabela, o utilizador, o estabelecimento, o prato em que a pesquisa foi baseada, como também a data em que foi efetuada.

### 4.1.4 Tabela Crítica

É onde todos os comentários e avaliações realizados pelos utilizadores sobre um estabelecimento são guardados. Cada comentário e avaliação relaciona-se com o **Estabelecimento** e **Utilizador** através do *ID* e *username* destes, respetivamente, que são elementos identificadores nas tabelas. Também é guardada a data em que o comentário foi efetuado. Pelo facto de um utilizador não poder fazer mais do que um comentário e avaliação a um estabelecimento, a chave primária da crítica é a chave composta do estabelecimento e utilizador.

### 4.1.5 Tabela Prato

O nome de cada prato é guardado nesta tabela. O seu nome serve para identificá-lo nas tabelas em que o prato for referenciado. Como não pode existir mais do que um prato na mesma tabela, com o mesmo nome, a chave primária é o próprio nome do prato.

### 4.1.6 Tabela Ementa

Resulta da junção entre elementos da tabela **Estabelecimento** e **Prato**. Ou seja, contém as informações sobre os pratos que os estabelecimentos servem. Contudo, o mesmo prato em



estabelecimentos diferentes pode não ser igual, podendo existir entre eles detalhes que não são comuns. Sendo por isso também guardado, para o prato de cada estabelecimento, um detalhe que possa diferenciá-lo desse prato noutro estabelecimento. Além do detalhe, existe ainda um *booleano* que identifica a existência ou inexistência desse mesmo detalhe. Por exemplo, o “Bife à Cortador” no Restaurante A é servido com batata frita e arroz, mas no Restaurante B só é servido com batata frita.

#### 4.1.7 Tabela Estabelecimento

Contém todas as informações relativas a um estabelecimento, nomeadamente: nome, foto, descrição, morada, total das avaliações, número total de avaliações e a média das avaliações. Quando for necessário obter informações sobre formas de contactar o estabelecimento (*website*, telefone, etc.), existe a tabela **Contacto** que guarda tudo isso. O horário de funcionamento está guardado na tabela **Horario\_Funcionamento**. Os comentários e as avaliações que os utilizadores fazem sobre o estabelecimento encontram-se na tabela **Critica**. Os pratos que são servidos no estabelecimento encontram-se na tabela **Ementa**.

#### 4.1.8 Tabela Contacto

Aqui são guardados as várias formas de comunicação que existem para entrar em contacto com um estabelecimento. Todas as linhas desta tabela têm um contacto diferente (*website*, telefone, etc.) que podem ser, se for esse o caso, pertencentes ao mesmo estabelecimento. Existe uma coluna com o *ID* do estabelecimento a que se refere o contacto, para quando for necessário relacionar linhas das tabelas **Contacto** e **Estabelecimento**.

#### 4.1.9 Tabela Horario\_Funcionamento

Nesta tabela encontram-se os horários de funcionamento de todos os estabelecimentos. Cada linha tem a hora de início e a hora de fim, assim como o dia de funcionamento. Caso o mesmo estabelecimento tenha para o mesmo dia da semana vários horários de funcionamento, serão guardados em linhas diferentes. Isto porque, apesar do dia e o estabelecimento serem os mesmos, as horas de início e de fim são diferentes. Para distinguir estas linhas unicamente, a chave primária desta será a junção de todos os atributos existentes.

## 4.2. Dicionário de Dados

Nome da Entidade	Descrição	Sinónimos	Contexto
Estabelecimento	Termo geral que representa todos os estabelecimentos	Restaurante	Os estabelecimentos fazem parte dos resultados das pesquisas efetuadas e possuem uma ementa (conjunto de pratos gastronómicos com diversos detalhes)
Prato	Termo geral que representa todos os pratos gastronómicos	Refeição	Um prato faz parte da ementa de um restaurante e, dependendo do estabelecimento, pode ter detalhes distintos
Utilizador	Termo geral que representa todos os clientes da aplicação	Cliente	Cada cliente efetua pesquisas e tem a si associado um histórico de pesquisas anteriores
Crítica	Termo geral que representa uma opinião de um utilizador, seja sob o formato de um comentário e/ou de uma avaliação	Comentário, Avaliação, Opinião	Cada crítica é efetuada por um utilizador em relação a um determinado estabelecimento

A Base de Dados é constituída por quatro entidades distintas: Estabelecimento, que representa os restaurantes da zona de Braga; Prato, que representa os pratos gastronómicos presentes nas ementas dos vários estabelecimentos; Utilizador, que representa todos os clientes da aplicação; Crítica, que representa a opinião de um utilizador, seja ela um comentário ou uma avaliação.

Entidade	Multiplicidade	Relação	Multiplicidade	Entidade	Atributos	Descrição	Tipos de Dados	Nulo/Composto/Multivalor
Estabelecimento	N	Tem	N	Prato	detalhe	Detalhe de um prato gastronómico num determinado estabelecimento. Pode-se tratar de ingredientes ou algo mais específico (por exemplo: sem glúten)	50 Caracteres	Não/ Não/ Não
Estabelecimento	N	Histórico	N	Utilizador	data	Data em que a pesquisa foi efetuada	DATETIME	Não/ Não/ Não
Estabelecimento	N	Histórico	N	Prato	data	Data em que a pesquisa foi efetuada	DATETIME	Não/ Não/ Não
Utilizador	N	Tem	N	Prato	Preferência	Preferência de um utilizador em relação a um prato gastronómico	50 Caracteres	Não/ Não/ Não
					Contador	Número de vezes que o utilizador fez uma pesquisa com a mesma preferência para o mesmo prato. Quanto maior for o seu número, maior relevância terá	INT	Não/ Não/ Não
					Valor	Detalhe tem valor positivo ou negativo	BIT	Não/ Não/ Não
Crítica	N	Sobre	1	Estabelecimento	----	----	----	----
Utilizador	1	Efetua	N	Crítica	----	----	----	----

Existem vários relacionamentos distintos entre as entidades, que representam a associação que cada entidade possui com outra. O estabelecimento é relacionado com o prato

num relacionamento com multiplicidade N:N, que indica que um prato possui N pratos (uma ementa é um conjunto de pratos gastronômicos) e um prato encontra-se presente em N restaurantes (o mesmo prato pode estar presente na ementa de estabelecimentos diferentes). Este relacionamento possui ainda o atributo detalhe, que tanto pode ser um dos ingredientes do prato, como também pode representar algo mais específico, como, por exemplo o prato ser “gluten-free”. Assim, um detalhe é um termo-chave que caracteriza o prato gastronômico.

Por outro lado, também verificamos que o estabelecimento se encontra associado num único relacionamento com as entidades Utilizador e Prato, que representa o histórico de cada cliente da aplicação. Deste modo, sempre que um utilizador visitar um determinado estabelecimento, serão guardadas as informações desse restaurante, bem como o prato pesquisado. Por esta razão, justifica-se a existência de um relacionamento que envolve três entidades diferentes: Estabelecimento, Prato, Utilizador, uma vez que é necessário associar o prato pesquisado ao local onde decorreu essa degustação e o cliente envolvido nessa visita. De modo a ser possível ordenar o histórico da pesquisa mais recente para a mais antiga, também é necessário armazenar a sua data, que é representada por um atributo do relacionamento.

O utilizador está relacionado com o prato, tendo em conta que este pode possuir preferências em relação à sua confeção. O relacionamento possui, assim, dois atributos: preferência, indicação da preferência do cliente em relação àquele prato, contador, que sempre que o utilizador pesquise pelo prato com aquela preferência, é incrementado e valor, que indica se a preferência tem valor positivo ou negativo. Desta maneira, teremos preferências com diferentes níveis de relevância. Uma preferência que tenha sido pesquisada um maior número de vezes terá mais importância que uma preferência que tenha sido pesquisada apenas uma vez, por exemplo.

Por fim, a crítica está associada com o utilizador num relacionamento de multiplicidade 1:N, uma vez que um utilizador pode efetuar várias críticas (ou seja, uma crítica é efetuada por um utilizador). Desta forma, sabemos quem é o autor de um comentário e avaliação. Como um utilizador irá comentar um restaurante que frequentou, a crítica encontra-se ainda relacionada com o estabelecimento num relacionamento de 1:N, pois uma crítica é sobre um restaurante (um restaurante possui várias críticas).

Entidades	Atributos	Descrição	Tipo de Dados	Nulo/Composto/ Multivalor
Estabelecimento	ID	Identifica de forma única o estabelecimento	INT	Não/ Não/ Não
	Nome	Nome do estabelecimento	45 Caracteres	Não/ Não/ Não
	Descricao	Descrição do estabelecimento	TEXT	Sim/ Não/ Não
	Morada:	Morada do estabelecimento	50 Caracteres	Não/ Não/ Não
	Contacto:	Contactos do estabelecimento	----	Não/ Sim/ Sim
	- Tipo	Tipo do contacto	16 Caracteres	Não/ Não/ Não
	- Descricao	Descrição do contacto	50 Caracteres	Não/ Não/ Não
	Horario_Funcionamento:	Dias e horas de funcionamento do estabelecimento	----	Não/ Sim/ Sim
	- Dia	Dia da semana em que o estabelecimento se encontra em funcionamento	INT	Não/ Não/ Não
	- Inicio	Horário de abertura do estabelecimento	TIME	Não/ Não/ Não
	- Fim	Horário de encerramento do estabelecimento	TIME	Não/ Não/ Não
	Foto	Foto do estabelecimento	TEXT	Sim/ Não/ Não
	Total_avaliacao	Soma de todas as avaliações efetuadas	INT	Não/ Não/ Não
		Número total de		

	Numero_avaliacao	avaliações	INT	Não/ Não/ Não
	Média_avaliacao	Média das avaliações	DECIMAL(2,1)	Não/ Não/ Não
Prato	Nome	Nome do prato gastronómico	50 Caracteres	Não/ Não/ Não
Utilizador	Username	Username do utilizador, que será visível para os outros utilizadores	24 Caracteres	Não/ Não/ Não
	Userpass	Password do utilizador, que não é visível pelos outros utilizadores	24 Caracteres	Não/ Não/ Não
Crítica	Avaliacao	Avaliação dada por um utilizador a um determinado estabelecimento	INT	Sim/ Não/ Não
	Comentario	Comentário efetuado por um utilizador em relação a um estabelecimento	160 Caracteres	Sim/ Não/ Não
	DataComentario	Data e hora em que foi efetuada uma opinião de um utilizador	DATETIME	Sim/ Não/ Não

O Estabelecimento tem a si associado um *ID* que o identifica unicamente na *BD*, um nome, uma descrição, uma foto e dois parâmetros (número e soma total das avaliações) que permitirão calcular a média das avaliações. Todos estes atributos mencionados são simples.

Por outro lado, a morada é um atributo composto, uma vez que tem campos distintos, como a rua, o número, o código-postal e a localidade. Por sua vez, o contacto é um atributo composto e multivalorado, dado que um restaurante pode possuir vários tipos de contacto (telemóvel, *e-mail*, *Facebook* e *website*).

O Prato possui o nome. O Utilizador, por sua vez, contém o username, que será a sua identificação única e será visível para os outros utilizadores (por exemplo, para verificar quem foi o autor de um comentário) e uma password, que é privada, ou seja, os outros utilizadores não a podem ver.

Por fim, a Crítica também uma avaliação e um comentário (efetuados por um utilizador), assim como a data e a hora em que foi submetida.

## 5. Especificação das Interfaces

Nesta fase optou-se por contruir a *mockup* o mais perto possível do produto final priorizando a localização de menus e das janelas mais importantes. A *mockup* foi desenhada com o intuito de tornar a aplicação *Chewie* fácil de utilizar e o mais apelativa possível. Cada janela representa uma ação bastante específica, independente e intuitiva que, para além disso, é acompanhada de um menu bastante simples localizado na parte superior do ecrã. Este menu permite a navegação entre as diferentes janelas da aplicação. Nas secções seguintes descrevem-se, com detalhe, as janelas da aplicação *Chewie*.

### 5.1. Login



Figura 18 - Mockup do Login

Esta janela tem o intuito de permitir o início de sessão (ou registo) de um utilizador e é apenas executada uma vez. Na janela são apenas apresentados o logotipo da aplicação, duas *textBoxes* (que representam o nome de utilizador e uma palavra-passe) e um botão que efetua o *login* propriamente dito de um utilizador, enviando as suas credenciais.

## 5.2. Home



Figura 19 - Mockup da janela Home

Esta janela representa a página inicial e é carregada sempre que se inicia a aplicação ou sempre que o utilizador pressiona no botão *home* do menu. Esta apresenta apenas o logotipo da aplicação e a barra da pesquisa (composta por uma *textBox* para pesquisa textual e de um botão com o símbolo de um microfone para a pesquisa por voz).



### 5.3. Histórico



Figura 20 - Mockup do Histórico

A apresentação do histórico é feita por secções de tal forma que cada uma delas represente uma hora de um dia específico, sendo as mais recentes carregadas primeiro. Dentro de cada secção, da esquerda para a direita localizam-se o restaurante, o prato e a hora de uma pesquisa. Para o utilizador voltar a efetuar a mesma pesquisa, basta pressionar na secção relativa a essa pesquisa. No fundo do ecrã, distinguem-se dois botões. À esquerda, “selecionar” tem como propósito selecionar uma ou mais pesquisas para apagá-las posteriormente. À direita, “Apagar Histórico”, tem como propósito apagar todas as pesquisas efetuadas.

## 5.4. Resultados

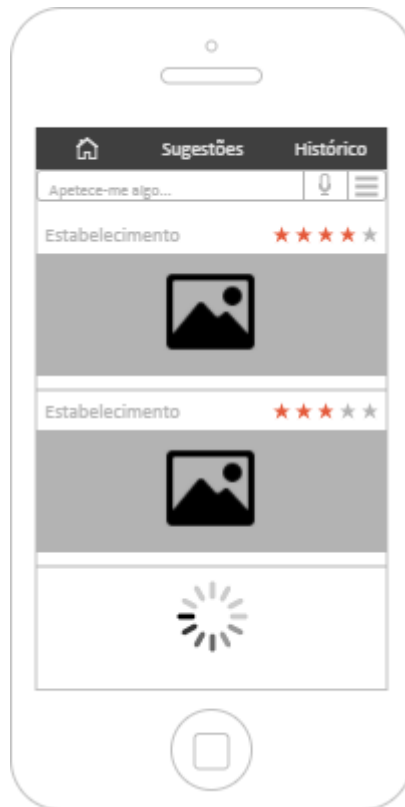


Figura 21 - Mockup dos Resultados

Após ser efetuada uma pesquisa, é apresentado um conjunto de estabelecimentos que satisfazem os critérios dessa pesquisa. Estes são representados sequencialmente de cima para baixo sendo permitido ao utilizador efetuar *scroll*. A ordem destes deve seguir as preferências do utilizador tendo em conta o número de estrelas. Os estabelecimentos que aparecerão inicialmente terão, portanto, a melhor pontuação. Por cada estabelecimento, por cima da imagem respetiva e à esquerda destaca-se o nome do mesmo. Logo à direita, o número de estrelas e por fim, abaixo, a imagem do estabelecimento respetivo. A **Figura 21** ilustra um exemplo em que o utilizador fez *scrolling* até ao final de uma página de resultados, ou até ao fim destes, e se tenta carregar mais resultados da pesquisa efetuada.

Tal como esta janela existe a janela de Sugestões que apresenta exatamente o mesmo esquema para a apresentação de resultados. A única diferença consiste na apresentação dos cinco melhores estabelecimentos que um utilizador ainda não tenha visitado.

## 5.5. Estabelecimento

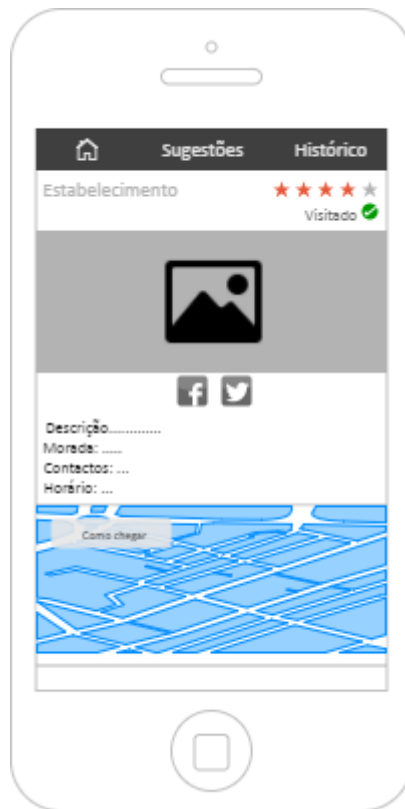


Figura 22 - Mockup da Seleção de um Estabelecimento

Ao seleccionar-se um estabelecimento, é apresentada uma janela com detalhes do mesmo. Aqui distinguem-se alguns detalhes importantes. O número de estrelas mantém-se no canto superior direito, agora acompanhado com uma indicação se o estabelecimento já foi visitado pelo utilizador. Abaixo, encontramos a imagem do estabelecimento que na parte inferior tem botões para partilhar a sua experiência numa das redes sociais disponíveis (podendo posteriormente voltar à página e partilhar novamente). Mais abaixo, encontramos quatro blocos de texto. O primeiro representa a descrição do estabelecimento mencionando aspetos importantes, o segundo a morada respetiva, seguindo-se dos contactos disponíveis e por fim, o horário de funcionamento. Por fim, no fundo do ecrã distingue-se o mapa visual da localização do estabelecimento com as indicações de como lá chegar tendo em conta a localização atual do dispositivo.

## 5.6. Preferências



Figura 23 - Mockup das Preferências

Após uma pesquisa, o utilizador tem a opção de a refinar. Assim, ao fundo da barra horizontal de pesquisa, encontra-se um botão de preferências. Ao ser clicado, uma pequena janela é suspensa com um conjunto de opções de preferências. No topo dessa janela, a distância máxima ao local. Abaixo, a avaliação mínima. O utilizador deve clicar no número de estrelas mínimo que pretende. Por fim, no fundo da janela de preferências, é dada a opção de filtrar dos resultados da pesquisa apenas estabelecimentos que se encontrem abertos no momento.

## 5.7. Comentários

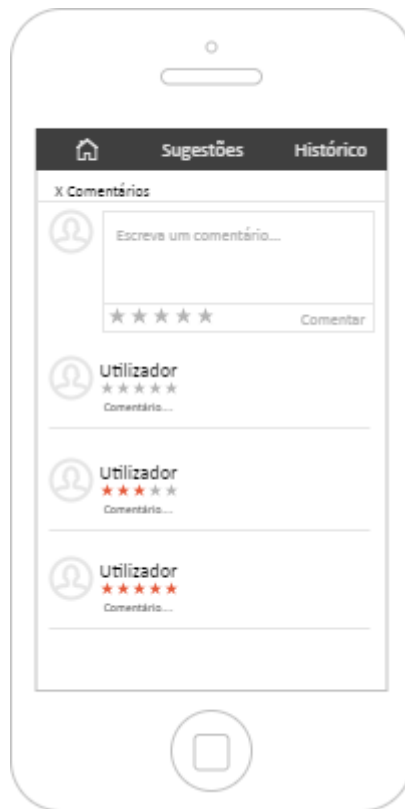


Figura 24 - Mockup dos Comentários

Os comentários são um componente da janela de um estabelecimento e dizem respeito apenas a um estabelecimento em específico. O utilizador é convidado a comentar a sua experiência num dado estabelecimento. Este pode escrever um comentário e uma avaliação ao mesmo tempo através da *textBox* ilustrada na **Figura 24**. É importante referir que o utilizador pode apenas avaliar, deixando o comentário em branco, ou apenas comentar, não clicando nas estrelas da *textBox*. Os comentários efetuados são ilustrados do mais recente para o mais antigo e em cada um destes visualizam-se a foto de um utilizador, o nome de utilizador, o comentário propriamente dito e a avaliação. Um utilizador pode facilmente verificar os diferentes comentários efetuados a um estabelecimento fazendo *scroll*.

## 5.8. API's

No planeamento do nosso projeto verificamos que existe um número de *API's* necessário para o bom funcionamento de determinadas tarefas da aplicação. Enumeraremos, de seguida, algumas *API's* que consideremos serem adequadas para o bom funcionamento dessas tarefas:

- *System.Speech* – Necessária para a pesquisa por voz. Necessita-se sobretudo de um método que inicia o reconhecimento por voz e o traduza para uma representação textual.

- *System.Web* – Para a ligação à *Web*. Esperam-se métodos que efetuem uma conexão à *Internet*, bem como uma verificação a essa mesma conexão. Esta *API* deve também conter um método que permita a obtenção da localização do utilizador através do seu endereço *IP*.
- *Windows Maps* – Esta *API* é compatível com a plataforma de desenvolvimento da nossa aplicação e deve permitir uma ilustração de um mapa, bem como a criação de um trajeto até um dado estabelecimento.
- *API's* de gramática em *C#* - A linguagem *C#* oferece métodos que usam como recurso as linguagens *LEX* e *YACC* para a definição de gramáticas e deteção de padrões. Isto é bastante útil para a extração de informações relevantes do texto de uma pesquisa (nome do prato, ingredientes, ...). São necessários métodos de *parsing* e definição de gramáticas, bem como de palavras chave e padrões.
- *API* do *Facebook* – Espera-se que esta *API* forneça métodos que permitam partilhar um conteúdo (comentário, foto, avaliação) nesta rede social.
- Classe *document* de *Javascript* – Através da linguagem *Javascript* e da classe *document*, pode-se facilmente extrair todos os dados necessários para a população da base de dados. Esta classe possui inúmeros métodos que permitem a obtenção de elementos, atributos e fácil navegação por todos os nodos de uma hierarquia em *HTML*. Torna-se apenas necessário conhecer as estruturas dos *WebSites* das fontes de informação.

Além das *API's* necessárias, é importante referir que esta aplicação será desenvolvida com recurso ao *template Windows Universal (Blank App)*. Esta opção é vantajosa pois aumenta a portabilidade da aplicação, tornando-a suportável para várias plataformas em *Windows (Windows Desktop, Mobile, Surface, ...)*.

## 6. Conclusões e Trabalho Futuro

O intuito da etapa de especificação do projeto consiste na análise (com bastante detalhe) de cada componente do projeto. Este processo é bastante útil para aumentar a robustez, segurança e versatilidade de um projeto. Para além disso permite uma resolução antecipada de inúmeros problemas ou *bugs* que poderiam vir a decorrer no processo de construção. No fundo, este método serve para aumentar a taxa de sucesso de um projeto.

Baseando-nos (o máximo possível), na nossa especificação, a próxima etapa consiste na construção do produto final. Tendo em conta o prazo previsto de entrega do mesmo, dar-se-á prioridade às funcionalidades básicas para o bom funcionamento da aplicação *Chewie*, sendo estas, a povoação da base de dados, implementação da pesquisa e obtenção dos resultados e construção da *User Interface*. Depois de todas estas funcionalidades básicas estarem completamente operacionais serão acrescentas à aplicação o maior número de funcionalidades extras que se conseguir.

## Referências

Connolly, T., Begg, C., Database Systems: a Practical Approach to Design Implementation and Management

Sommerville, I., Software Engineering.



## **Anexos**

Em anexo encontram-se a especificação do Use Case da Pesquisa, assim como o seu Diagrama de Sequência de *Use Case* e Diagrama de Sequência.

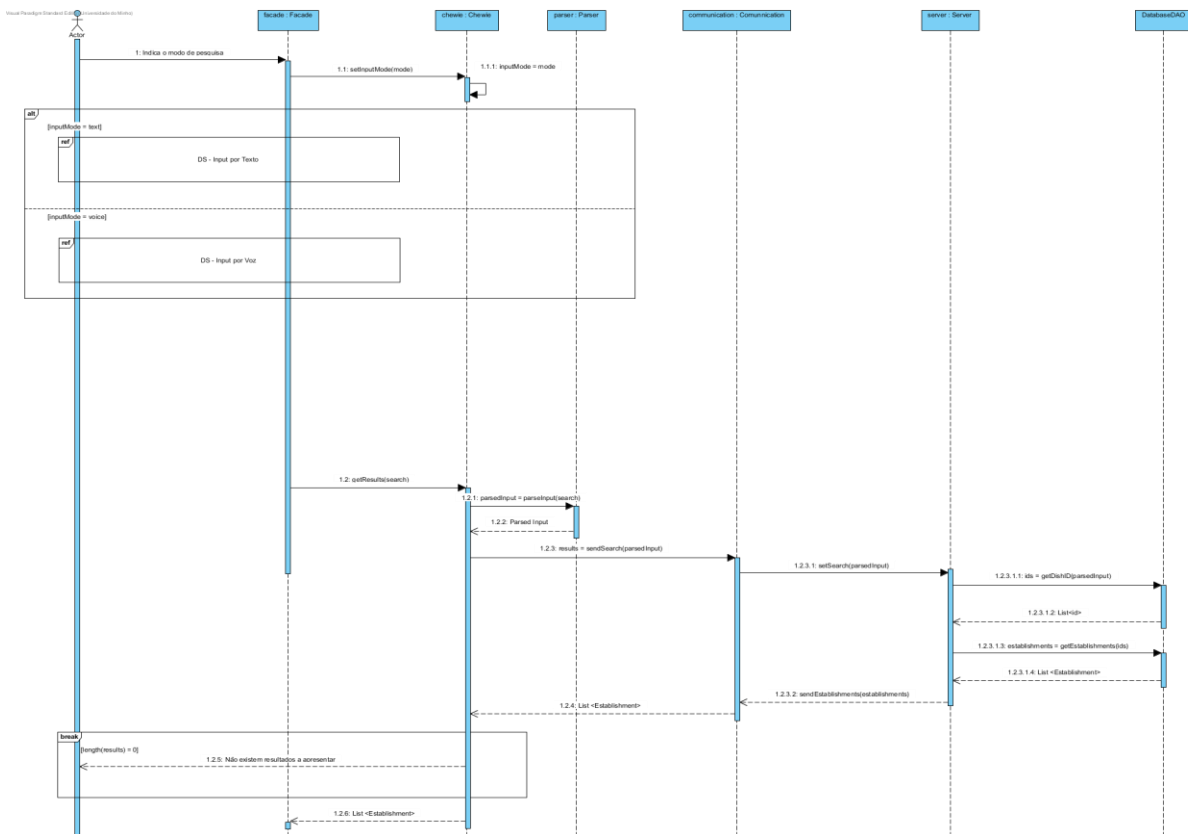
## I. Anexo 1

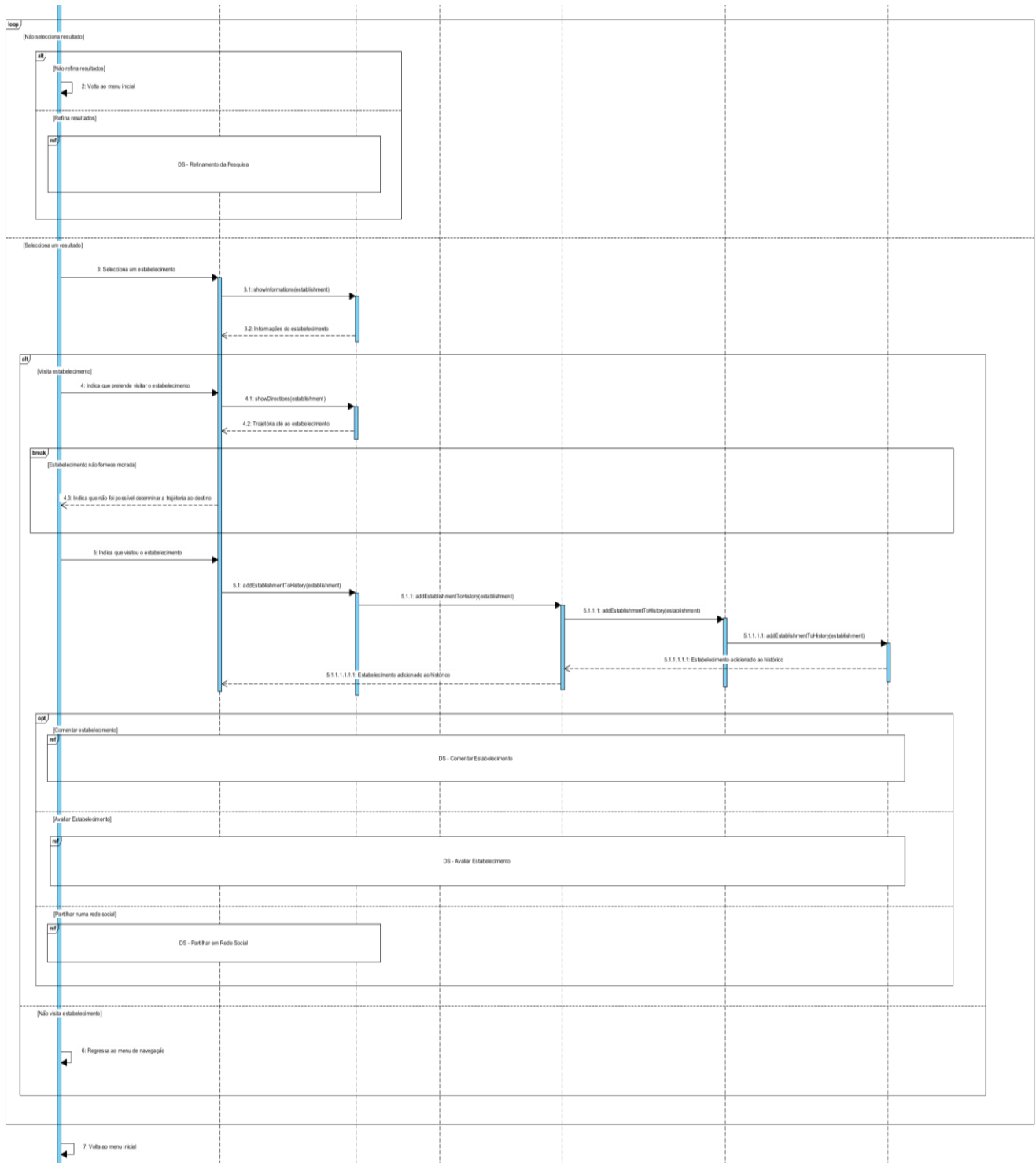
<b>Name</b>		Pesquisa	
<b>Brief Description</b>		O utilizador efetua uma pesquisa acerca de um prato gastronómico	
<b>Preconditions</b>		Utilizador autenticado	
<b>Post-conditions</b>		Resultados da pesquisa apresentados	
		<b>Actor Input</b>	<b>System Response</b>
<b>Flow of Events</b>	1	Indica que vai introduzir por texto os parâmetros da pesquisa	
	2	Introduz parâmetros da pesquisa	
	3		Verifica se os parâmetros não são numéricos
	4		Verifica se os parâmetros não são símbolos
	5		<<include>> Analisa Parâmetros
	6		Efetua uma pesquisa na Base de Dados pelos termos-chave
	7		Apresenta os resultados
	8	Navega pela lista de resultados	
	9	Seleciona um estabelecimento	
	10		Apresenta as informações completas do estabelecimento
	11	Indica que pretende visitar o estabelecimento	
	12		Fornece direções ao destino
	13	Indica que visitou o estabelecimento	
	14		Guarda a nova visita ao estabelecimento
	15		Adiciona o estabelecimento e o prato ao histórico do utilizador
	16		Adiciona parâmetros de pesquisa às preferências do utilizador
	17	Indica que pretende comentar o estabelecimento	
	18		Apresenta janela para comentar
	19	Introduz comentário	
	20		Valida comentário
	21		Guarda comentário na Base de Dados
	22		Informa que o comentário foi submetido
	23	Volta ao menu inicial	
<b>Comportamento Alternativo (passo 1)</b> [indica por voz os parâmetros da pesquisa]			
	1.1	Indica que vai introduzir por voz os parâmetros da pesquisa	
	1.2		Regressa a 2
<b>Comportamento Alternativo (passo 8)</b> [refinamento dos resultados]		<b>Actor Input</b>	<b>System Response</b>
	8.1	<<include>> Refinamento da Pesquisa	
	8.2		Regressa a 8

Comportamento Alternativo (passo 13) [utilizador não visita estabelecimento]		Actor Input	System Response
	13.1	Indica que não visitou o estabelecimento	
	13.2	Regressa a 8	
Comportamento Alternativo (passo 16) [não existem preferências para adicionar]		Actor Input	System Response
	16.1		Regressa a 17
Comportamento Alternativo (passo 17) [utilizador avalia o estabelecimento]		Actor Input	System Response
	17.1	Indica que pretende avaliar o estabelecimento	
	17.2		Apresenta janela para avaliar
	17.3	Introduz avaliação	
	17.4		Guarda avaliação na Base de Dados
	17.5		Informa que a avaliação foi submetida
	17.6		Regressa a 17
Comportamento Alternativo (passo 17) [utilizador partilha numa rede social]		Actor Input	System Response
	17.1	Indica que pretende partilhar numa rede social	
	17.2		Apresenta janela para escolher a rede social
	17.3	Seleciona rede social pretendida	
	17.4		Apresenta janela para partilhar na rede social escolhida
	17.5	Efetua a partilha	
	17.6		Confirma que a partilha foi efetuada
	17.7		Regressa a 17
Comportamento Alternativo (passo 17) [utilizador não pretende partilhar a sua opinião]		Actor Input	System Response
	17.1	Indica que não pretende dar a sua opinião	
	17.2	Regressa a 23	
Exceção 1 (passo 3) [parâmetros contêm caracteres numéricos]		Actor Input	System Response
	3.1		Indica que parâmetros da pesquisa são numéricos

<b>Excepção 1 (passo 4) [parâmetros contêm símbolos]</b>		<b>Actor Input</b>	<b>System Response</b>
	4.1		Indica que parâmetros da pesquisa são símbolos
<b>Excepção 2 (passo 7) [não existem resultados a apresentar]</b>		<b>Actor Input</b>	<b>System Response</b>
	7.1		Indica que não existem resultados para apresentar
<b>Excepção 3 (passo 12) [estabelecimento não fornece morada]</b>		<b>Actor Input</b>	<b>System Response</b>
	7.1		Indica que não foi possível determinar a trajetória ao destino

## II. Anexo 2





### III. Anexo 3

