



Universidade do Minho

Escola de Engenharia

Mestrado Integrado em Engenharia Informática

## Unidade Curricular de Laboratórios de Informática IV

# Taste Advisor

**Ana Rita Marques, A74218**

**Célia Figueiredo, A67637**

**Humberto Vaz, A73236**

**Ricardo Lopes, A72062**

Fevereiro, 2016

LI4

|                  |  |
|------------------|--|
| Data de Recepção |  |
| Responsável      |  |
| Avaliação        |  |
| Observações      |  |

## Taste Advisor

**Ana Rita Marques, A74218**

**Célia Figueiredo, A67637**

**Humberto Vaz, A73236**

**Ricardo Lopes, A72062**

Fevereiro, 2016

<</opcional Dedicatória>>

# Resumo

Este projeto visa o desenvolvimento de *software*, implementado numa plataforma *Web-based*, utilizando ferramentas Microsoft, capaz de recomendar e localizar locais nos quais se possa comer algo que verdadeiramente apeteça ao utilizador. Caberá ao utilizador escolher se o sistema será de aplicação genérica ou se será especificamente relacionado a uma dada área gastronómica, nacional ou internacional. O sistema deverá ser capaz de ter em conta a localização atual do utilizador, assim como as suas preferências.

Este projeto será desenvolvido em três fases: Fundamentação, Especificação e Implementação.

Terminada a fase de fundamentação, é apresentada a fase de especificação em que são especificados e analisados os requisitos do sistema em questão. Nesta mesma fase, além do levantamento e da análise de requisitos, foram também criados os modelos de sistema (em UML). Estes modelos consistem no diagrama de *Use Cases*, diagramas de Sequência, diagramas de Atividades, modelo de Domínio e diagrama de Classes.

Com os requisitos levantados e analisados e os modelos construídos procedeu-se à idealização e elaboração da base de dados a partir da construção dos respetivos modelos conceptual e lógico, baseados nos requisitos.

Para concluir esta segunda fase, são apresentados os *mockups* que constituem o planeamento da interface com o utilizador.

## Área de Aplicação: Restauração

**Palavras-Chave:** Desenvolvimento de Software, Engenharia de Software, Aplicação, Análise de Requisitos, Metodologias de Software, Assistente de Restauração, Investigação, Gestão de Projeto, Maquete, Diagrama de Gantt, Microsoft Project, Use Cases, Modelo Domínio, Mockups.

# Índice

|   |    |
|---|----|
| 1. Introdução   | 1  |
| 1.1. Contextualização                                     | 1  |
| 1.2. Apresentação do Caso de Estudo                       | 2  |
| 1.3. Motivação e Objetivos                                | 2  |
| 1.4. Justificação do sistema                              | 3  |
| 1.5. Utilidade do Sistema                                 | 3  |
| 1.6. Estabelecimento da identidade do projeto             | 3  |
| 1.7. Identificação dos recursos necessários               | 3  |
| 1.8. Maqueta do sistema                                   | 6  |
| 1.9. Definição de um conjunto de medidas de sucesso       | 6  |
| 1.10. Plano de desenvolvimento                            | 7  |
| 1.11. Organização do documento                            | 9  |
| 2. Análise de requisitos                                  | 10 |
| 2.1. Levantamento de requisitos do Sistema e Utilizador   | 10 |
| 2.1.1 Módulo Registo                                      | 10 |
| 2.2. Módulo Utilização/Pesquisa                           | 11 |
| 2.2.1 Sem login efetuado                                  | 11 |
| 2.2.2 Com login efetuado                                  | 12 |
| 2.3. Módulo Preferências do Utilizador                    | 13 |
| 2.4. Levantamento de requisitos do Sistema e Proprietário | 14 |
| 3. Modelos do sistema                                     | 15 |
| 3.1. Diagramas UML: Diagramas de <i>Use Case</i>          | 15 |
| 3.1.1 Diagrama de <i>Use Case</i> : BackOffice            | 15 |
| 3.2. Diagramas UML: Diagramas de Atividade                | 25 |
| 3.2.1 Diagrama de Atividade: Efetuar Pesquisa             | 26 |
| 3.2.2 Diagrama de Atividade: Editar Preferências          | 26 |
| 3.3. Diagramas UML: Modelo de Domínio                     | 28 |
| 3.4. Diagramas UML: Diagrama de Classes                   | 29 |
| 4. Bases de Dados   | 30 |
| 4.1. Modelo Conceitual                                    | 30 |
| 4.1.1 Identificar os tipos de entidades                   | 30 |

|   |    |
|---|----|
| 4.1.2 Identificar tipos de relacionamento   | 31 |
| 4.1.3 Identificar e associar atributos com os tipos de entidades e relacionamentos      | 31 |
| 4.1.4 Desenho do diagrama ER  | 33 |
| 4.1.5 Revisão do modelo de dados com o utilizador                                       | 33 |
| 4.2. Modelo Lógico  | 34 |
| 4.2.1 Validação do modelo lógico através de normalização                                | 34 |
| 4.2.2 Elaboração e validação do esquema lógico da base de dados                         | 35 |
| 4.2.3 Definição do tamanho inicial da base de dados e análise do seu crescimento futuro | 36 |
| 4.2.4 Revisão do modelo lógico final com os futuros utilizadores                        | 36 |
| 5. Mockups  | 38 |
| 6. Conclusões e Trabalho Futuro   | 43 |

### **Anexos**

|            |    |
|------------|----|
| I. Anexo 1 | 48 |
|------------|----|

# Índice de Figuras

|   |    |
|---|----|
| Figura 1 - Maqueta do sistema   | 6  |
| Figura 2 - Diagrama de Gantt  | 9  |
| Figura 3 - Use Case: BackOffice   | 15 |
| Figura 4 - Use Case: <i>FrontOffice</i> , ações permitidas aos utilizadores (proprietário e utilizador) | 16 |
| Figura 5 - Especificação de use case: Efetuar Registo   | 16 |
| Figura 6 - Especificação do caso de uso: Pesquisar por termos   | 17 |
| Figura 7 - Diagrama de sequência: Pesquisa por termos   | 17 |
| Figura 8 - Especificação de Use Case: Avaliar Restaurante   | 18 |
| Figura 9 - Especificação Use Case: Editar Preferências  | 19 |
| Figura 10 - Diagrama de Sequencia: Editar Preferências  | 20 |
| Figura 11 - Subdiagrama Use Case: Pesquisa Por Termos   | 21 |
| Figura 12 - Subdiagrama: Adicionar Componente   | 22 |
| Figura 13 - Especificação de Use Case: Adicionar Restaurante à lista de Favoritos                       | 23 |
| Figura 14 - Subdiagrama: Remover Componentes  | 23 |
| Figura 15 - Especificação de Use Case: Remover Restaurante dos Favoritos                                | 24 |
| Figura 16 - Subdiagrama: Editar Preferências  | 25 |
| Figura 17 - Diagrama de Atividade: Efetuar pesquisa   | 26 |
| Figura 18 - Diagrama Atividades: Editar Preferências  | 27 |
| Figura 19 - Modelo de Dominio   | 28 |
| Figura 20 - Diagrama de Classes   | 29 |
| Figura 21 - Modelo conceitual   | 33 |
| Figura 22 - Modelo Lógico   | 35 |
| Figura 23 - Mockup: Página Inicial  | 38 |
| Figura 24 - mockup: Separador "Geral", separador "Menu", separador "Opiniões" e "Fotografias"           | 39 |
| Figura 25 - Mockup: Inicio de Sessão  | 40 |
| Figura 26 - Mockup: Criação de conta  | 40 |
| Figura 27 - Mockup: Ver Restaurantes Favoritos e remover  | 41 |

Figura 28 - Mockup: Ver pratos não Favoritos adicionados à lista de preferências

41

# **Índice de Tabelas**

|   |    |
|---|----|
| Tabela 1 – Tabela relativa às Entidades                     | 30 |
| Tabela 2 - Tabela de relacionamentos                        | 31 |
| Tabela 3 - Tabela de Associação entre atributos e entidades | 32 |

# **1. Introdução**

No âmbito da Unidade Curricular de Laboratórios de Informática IV, foi proposto o desenvolvimento de software, com o nome de “Mnhan-mnhan”, com o objetivo de rotinar os alunos em processos de construção de software, com particular ênfase na especificação e desenvolvimento de aplicações do mundo real, utilizando metodologias e processos de Engenharia de Software. O software deverá ser capaz de recomendar e localizar locais nos quais se possa comer "algo" que verdadeiramente apeteça ao utilizador.

## **1.1. Contextualização**

O passatempo de comer em restaurantes é aproveitado por uma grande variedade de pessoas. Existem vários tipos de restaurantes - alguns de *fast food*, outros de alta gastronomia e ainda de comida casual, entre outros. Cada um oferece um tipo diferente de serviço ou cozinha e pode atrair consumidores diferentes, no entanto quando queremos comer algo em específico torna-se difícil encontrar o local.

Com base nestas premissas a empresa ‘*Belo Lda*’ sentiu a necessidade de ter uma aplicação que colocasse alguns restaurantes numa base de dados com as suas respetivas ementas. Assim sempre que um dos seus funcionários ou clientes sentisse o desejo de comer ‘alguma coisa’ haveria uma aplicação que ajudasse a encontrar o local com a tal iguaria desejada. Posto isto surgiu a ideia de criar a ‘*Taste Advisor*’ que ajudará a comunidade em geral nesse sentido.

## **1.2. Apresentação do Caso de Estudo**

O software que se irá desenvolver deverá ser capaz de recomendar e localizar locais nos quais se possa comer "algo" que verdadeiramente apeteça ao utilizador. E, portanto, a empresa '*Belo Lda*' precisa que aplicação a desenvolver satisfaça de um modo geral os seguintes requisitos:

1. Permitir que o utilizador dite o que quer comer no momento, quer por escrito, quer por voz;
2. Apresentar uma lista de sugestões de locais, ordenadas de acordo com um dado índice de avaliação, cada local acompanhado por uma caracterização, relatos de clientes, formas de contactos, localização do local e forma de lá chegar através da visualização do caminho num mapa, assim como uma descrição, fotografia e preço do "algo" pretendido;
3. Permitir ao utilizador efetuar registo e autenticação na aplicação, o que permitirá que este faça o registo das suas preferências (ou não preferências), que poderão influenciar as apresentações de sugestões futuras, assim como permitirá armazenar o histórico dos locais já frequentados (sugeridos pela aplicação), que poderá ser revisto e possibilitar o utilizador de escolher novamente um desses locais;
4. Permitir ao utilizador registar a sua opinião sobre o local selecionado e o "algo" degustado, possibilitando-o publicar a respetiva opinião numa rede social, assim como "twittar" a experiência de degustação em tempo real;
5. Apresentar uma lista de tendências de degustação na zona onde o utilizador se encontra;
6. Usar o assistente operacional da plataforma onde está instalado para complementar, se necessário, as suas sugestões.

## **1.3. Motivação e Objetivos**

O motivo que nos levou à realização deste projeto consistiu no facto de inesperadamente nos apetecer comer algo e não saber onde. Daí que surgiram as seguintes perguntas: "Quantas vezes, inesperadamente, lhe apeteceu comer "algo" específico? Quantas vezes teve curiosidade sobre a gastronomia de uma determinada região?" De forma a melhorar a experiência e qualidade de degustação dos utilizadores, temos como objetivo o

desenvolvimento de software capaz de auxiliar e incentivar os mesmos a satisfazer os seus desejos. Desde um simples pastel de nata até uma deliciosa francesinha, esta ferramenta será capaz de sugerir o melhor local, com mais qualidade, com localização mais próxima, ou com o preço mais baixo.

## **1.4. Justificação do sistema**

Em termos práticos, este sistema de recomendação e localização irá melhorar a qualidade de vida dos seus utilizadores, auxiliando os mesmos a satisfazer o seu apetite, a qualquer momento. Sem perder muito tempo com pesquisas em diferentes sites de restaurantes, ou aplicações que só indicam quais os melhores restaurantes da zona, estes que por vezes não indicam o tipo de comida que fazem.

## **1.5. Utilidade do Sistema**

O sistema irá permitir que o utilizador encontre o local onde poderá satisfazer o seu apetite, de forma mais rápida, de maneira a melhorar o seu dia a dia, tornando a sua procura mais eficiente. O utilizador também não terá a preocupação de como chegar ao seu destino, pois será função do sistema mostrar o caminho num mapa.

## **1.6. Estabelecimento da identidade do projeto**

A ideia do projeto é desenvolver um sistema capaz de ajudar o utilizador a encontrar um local onde possa degustar o que lhe estiver a apetecer no momento, de acordo com a sua localização atual, de forma a que o local sugerido para a degustação seja o mais rápido de alcançar, ofereça o melhor preço ou sirva o produto com mais qualidade.

## **1.7. Identificação dos recursos necessários**

Para que o planeamento da aplicação desejada seja cumprido é necessário dispor de recursos humanos, de hardware e de software.

Os recursos humanos englobam os trabalhadores envolvidos no projeto e as suas respetivas horas de trabalho. Neste caso haverá um gestor de projeto que deverá ter controlo

do projeto, comparando a situação atual face ao planeado, identificando desvios e propondo ações corretivas, este deverá ser o elo de ligação entre o docente e o resto do grupo.

Os recursos de hardware necessários para o desenvolvimento do ‘Taste Advisor’ serão as máquinas pessoais dos elementos do grupo.

Por outro lado, os recursos de software utilizados serão a maioria ferramentas desenvolvidas pela Microsoft. O software instalado nas máquinas para o desenvolvimento da aplicação pretendida é:

**1) Microsoft Office**

Utilizaremos o editor de documentos (Microsoft Office Word) para elaborar relatórios do projeto, um editor de apresentações (Microsoft Office Power Point) para suportar a apresentação das diversas fases do projeto.

**2) Microsoft Visual Studio;**

É o ambiente de desenvolvimento das aplicações .NET.

**3) Microsoft .NET C#;**

C# foi desenvolvida pela Microsoft como parte da plataforma .NET, é uma linguagem fortemente tipada e possui paradigmas de programação imperativa, funcional, declarativa, orientada aos objetos, esta será a linguagem a utilizar para o desenvolvimento da aplicação.

**4) Microsoft Project**

É o software que permite a gestão de projetos, poderão ser elaborados diagramas de Gantt.

**5) Visual Paradigm**

É uma ferramenta de UML para o desenvolvimento de aplicações de larga escala, segundo uma abordagem por objetos.

**6) Microsoft SQL Server;**

O SQL Server é uma base de dados de referência do mercado proporcionando uma base sólida e escalável do sistema.

## **7) Bing Maps API**

O Bing Maps V8 é um dos controlos de mapeamento mais universais disponíveis. É suportado em navegadores padrão para PC e Mac, e em muitas plataformas móveis. Esta API é ideal para aplicações web-based com suporte para JavaScript e TypeScript.

## **8) Bing Speech API**

Converte áudio para texto. A API pode ser direcionada para ativar e reconhecer o áudio proveniente do microfone em tempo real.

## **9) Dados externos**

A informação sobre um determinado restaurante é agrupada e apresentada ao utilizador, de forma a que, através de uma única pesquisa este possa ter conhecimento da informação que necessita. Os dados iniciais serão retirados de uma aplicação já existente, no entanto a aplicação será enriquecida com os dados que o utilizador acrescenta à base de dados da aplicação ao deixar os seus comentários, fotografias, classificações relativas à sua experiência. Por outro lado, será pedido aos donos dos próprios restaurantes para atualizarem as ementas em tempo útil para não haver discrepâncias entre o virtual e o real. A seguir fica a descrição da aplicação que nos servirá de base para os dados iniciais do sistema.

1. **Zomato** é uma rede social cuja temática é centrada na área de restauração funcionando como um guia para os utilizadores através do auxílio do seu sistema de recomendação, estando disponível através da versão web ou através das aplicações móveis disponíveis para os sistemas operativos Windows, Android e iOS. Esta plataforma possui informação sobre restaurantes localizados em várias cidades por todo o mundo, no entanto, no âmbito deste projeto apenas são considerados os restaurantes que se encontram dentro dos limites geográficos de Portugal.

## 1.8. Maqueta do sistema

O sistema a desenvolver será composto por uma aplicação móvel. A aplicação móvel irá usar três componentes essenciais: GPS, voz e fotografias.

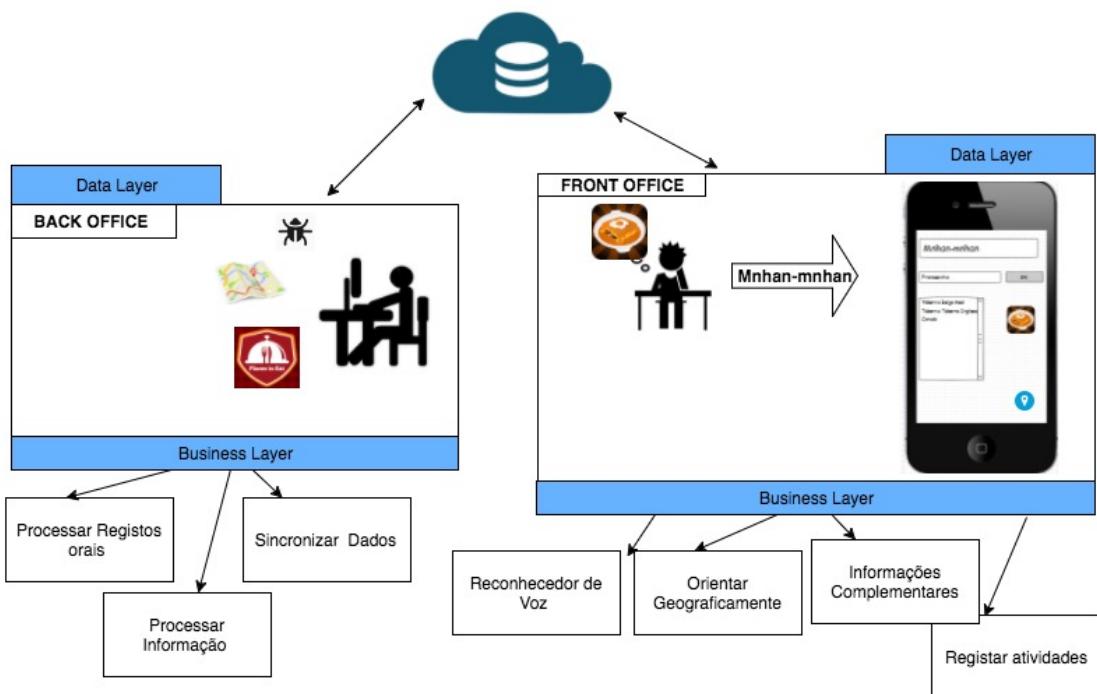


Figura 1 - Maqueta do sistema

## 1.9. Definição de um conjunto de medidas de sucesso

O produto a ser oferecido será uma aplicação disponibilizada para dispositivos móveis que utilizam os sistemas operacionais IOS e *Android*. Este produto (aplicação), não necessita de matéria prima nem de canais de distribuição, pois será acessível a qualquer pessoa que possua um *Smartphone* (utilizando os sistemas operacionais base). Dessa forma, não há custos relacionados a estes pontos em específico.

A aplicação será desenvolvida para qualquer pessoa que possua um *Smartphone* e sinta a necessidade de saber onde comer ‘algo’ que lhe apeteça. Não necessita que as

pessoas que usem a aplicação tenham um grau elevado de escolaridade, pois é bastante simples e apenas basta ter algum treino com *Smartphones*.

É imprescindível para o bom funcionamento e gestão de um negócio elaborar análises estratégicas e de mercado, a fim de obter vantagens competitivas frente aos concorrentes. O mercado de aplicações móveis é diferenciado e abrangente, sendo necessário para aceder ao mesmo apenas uma conexão com a Internet através do *Smartphone*. Dessa forma, o limite de alcance de mercado seria imposto apenas pelas questões de marketing e interesses pessoais.

Por mais que o levantamento de requisitos levante uma infinidade de informações que deverão estar dispostas na interface, é durante a interação real do utilizador com o sistema que os detalhes realmente são percebidos. Para projetar a experiência perfeita, a utilização de protótipos torna-se a forma mais eficaz de simular essa interação, devendo ser feita ainda no período de conceção do sistema, evitando gastos dispendiosos de tempo e monetários.

No desenvolvimento de software, a qualidade do produto está diretamente relacionada com qualidade do processo de desenvolvimento, desta forma, qualidade é a totalidade de características e critérios de um produto ou serviço que exercem suas habilidades para satisfazer às necessidades declaradas ou envolvidas.

Para concluir estes serão alguns dos pontos a ter atenção para que a aplicação tenha sucesso.

## 1.10. Plano de desenvolvimento

A primeira fase do projeto passou pela fundamentação do tema. Primeiramente, foi efetuada uma pesquisa acerca do assunto com o objetivo de recolher informação que auxiliasse o grupo a inserir-se no contexto do que é ser um assistente ao nível de restauração. Esta pesquisa permitiu ter uma percepção dos aspetos inerentes às atividades realizadas pelos estabelecimentos e possíveis desejos de utilizador, o que possibilitou sustentar a fundamentação do projeto. Partindo das informações recolhidas, concebemos grande parte do relatório (contextualização, apresentação do caso de estudo, motivação e objetivos).

A maquete (que consiste na idealização da arquitetura da aplicação) e a presente secção, foram realizadas de forma paralela, o que implicou a divisão do grupo em duas sub-equipas. Após a realização das tarefas correspondentes a esta fase do projeto, foram planeadas e delineadas as restantes fases que irão compor a realização do projeto, nomeadamente a modelação e a implementação. 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 período de desenvolvimento de cada uma delas.

Imaginámos que de forma a cumprir os prazos de entrega seja necessário realizar algumas das tarefas previstas simultaneamente. São previstos também períodos de tempo em que não se realiza qualquer atividade relacionada com o projeto. Estas paragens relacionam-se com a necessidade de reservar algum espaço no calendário para que os elementos do grupo possam desenvolver outros projetos.

Na fase de modelação, o grupo prevê que seja necessário um trabalho simultâneo dos quatro elementos constituintes para a análise de requisitos. De seguida, passar-se-á à conceção dos diagramas UML necessários à idealização do sistema, finalizando-se esta fase com um conjunto de revisões e correções. Julgamos que esta última tarefa será outra das partes em que teremos de investir mais tempo, de forma a garantir a coerência entre os diagramas construídos. Na última fase do projeto, correspondente à fase de implementação do sistema, foi onde tivemos maior dificuldade em prever o tempo que viria a ser despendido em cada tarefa. Contudo, julgámos que a implementação das bases de dados será uma das tarefas que levará mais tempo a ser completada. Relativamente às outras tarefas, o principal obstáculo que se opõe à capacidade de previsão do tempo é a falta de experiência por parte do grupo no uso das ferramentas e mecanismos que irão ser utilizados para a sua realização. Tal como na fase anterior do projeto, para a tarefa de testes e correção de erros foi alocado um período considerável de forma a garantir o correto funcionamento das funcionalidades implementadas. É previsto que a finalização do relatório se inicie (através de uma divisão em sub-grupos) antes da parte de testes terminar, de maneira a garantir que não só todo o processo de descrição de desenvolvimento do projeto esteja revisto e em conformidade com o que irá ser apresentado, mas também que o relatório seja acompanhado por documentação suficientemente aprofundada, clara e organizada.

Por último, é de realçar que a divisão das tarefas por elementos do grupo foi feita através de uma discussão em equipa em que foram tidos em conta os pontos fortes de cada pessoa. Contudo, esta atribuição inicial de responsabilidades é suscetível de ser alterada com base em aspectos futuros que o justifiquem, como por exemplo, dificuldades de execução de determinada tarefa que suscite necessidade de cooperação entre os vários membros do grupo. Um atraso no desenvolvimento do projeto poderá ser outro fator que leve a uma redistribuição dos trabalhos e, para além disso, a uma alteração no tempo destinado a cada tarefa.

Além disso, uma coordenação entre os elementos da equipa poderá ser necessária como forma de garantir a coesão e integração de todo o trabalho até então realizado. Posto isto e de forma a garantir um melhor planeamento de tarefas foi esboçado um Diagrama de Gantt.

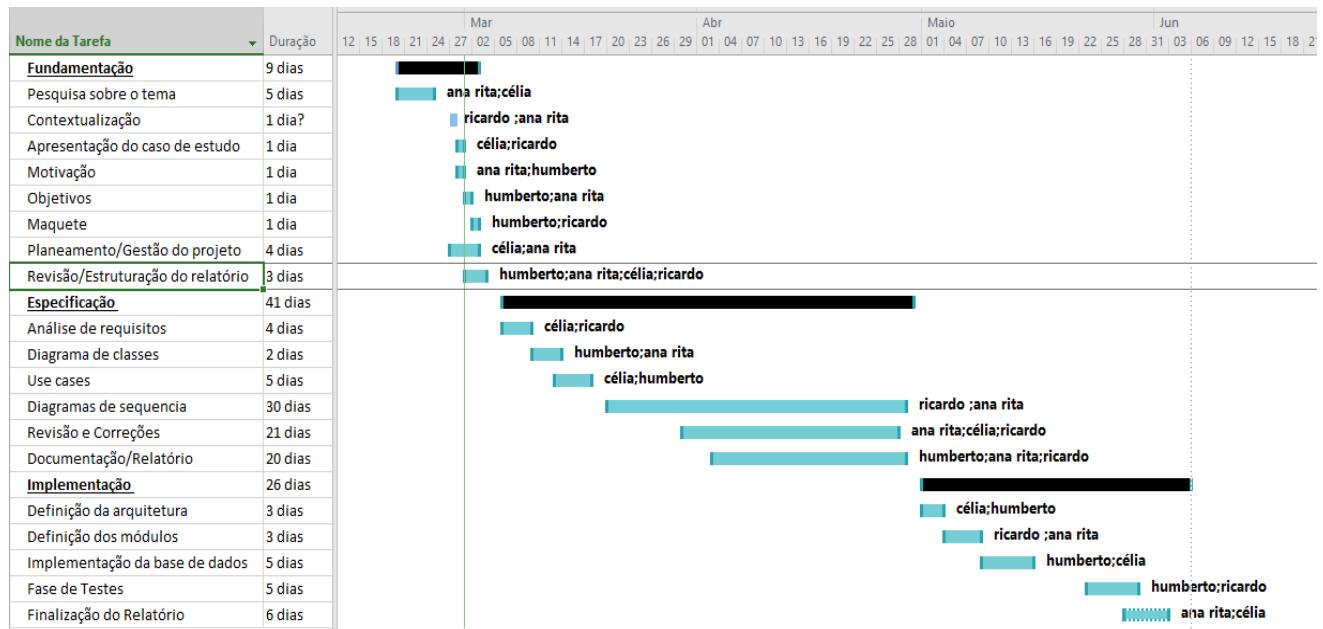


Figura 2 - Diagrama de Gantt

## 1.11. Organização do documento

Este documento está organizado em capítulos. Sendo que este primeiro relata sobre a fundamentação do projeto a realizar. O segundo capítulo relata sobre a análise de requisitos do sistema. O terceiro capítulo refere-se aos modelos de modelação (UML). O quarto capítulo trata sobre a base de dados. O quinto capítulo mostra os mockups pensados para este sistema. Por ultimo teremos o capítulo das conclusões.

## **2. Análise de requisitos**

O levantamento e análise de requisitos é uma das fases mais importantes na especificação de um sistema de *software*.

### **2.1. Levantamento de requisitos do Sistema e Utilizador**

O sistema tem como objetivo responder de forma simples às necessidades do utilizador. O sistema deverá filtrar a informação de acordo com as pesquisas dos utilizadores, se o utilizador tiver sessão iniciada a pesquisa poderá ser ainda mais filtrada de acordo com as suas preferências definidas no perfil. Para tal estão descritos abaixo os requisitos aos quais a aplicação deverá respeitar.

#### **2.1.1 Módulo Registo**

O utilizador deverá efetuar um registo para conseguir aceder a algumas funcionalidades da aplicação “*Taste Advisor*”. No registo será pedido o nome, email e password. Alternativamente o registo poderá ser efetuado de forma automática, isto é, fazendo o registo com os dados do *facebook* ou *Gmail*, isto implica que tenha a sessão iniciada nessas plataformas.

1. O sistema guardará na base de dados os dados de autenticação do registo através do registo automático com *Facebook* ou *Google* ou manual, fornecendo os dados pedidos.

O Utilizador necessita de estar registado para aceder às seguintes funcionalidades:

1. Adicionar/Remover comentário sobre determinado restaurante/prato
2. Adicionar/Remover fotografia alusiva ao restaurante/prato
3. Adicionar/Remover determinado restaurante/prato à sua lista de preferências
4. Editar outros campos relativos às preferências
5. Classificar determinado prato ou restaurante (0-5 estrelas)

## 2.2. Módulo Utilização/Pesquisa

O Utilizador poderá usar a aplicação com ou sem login efetuado. A aplicação efetuará pesquisas de acordo com o *profiling* do utilizador em questão, de acordo com as suas preferências degustativas. O ideal seria que a aplicação “adivinhasse” quais as preferências do utilizador, mas isso não sendo ainda possível, implementamos um módulo que trata as preferências do utilizador. Nesse módulo poderá editar algumas das suas preferências gastronómicas e também filtrar restaurantes de acordo com a localização e preço.

### 2.2.1 Sem login efetuado

1. O utilizador terá à sua disposição para pesquisa um campo de escrita e outro de reconhecimento de voz sob a forma de botão onde este poderá executar a sua pesquisa. A pesquisa será tratada em *backOffice*.
2. Cada item da lista apresentada terá: o nome do restaurante, uma imagem alusiva ao mesmo, a localização, preço médio (sendo exibido em três estados: baixo, médio, alto) e a classificação atribuída pelos utilizadores.
3. O utilizador poderá escolher como quer visualizar a informação podendo escolher entre os seguintes parâmetros:
  - a. Localização – baseado na localização mais próxima dos restaurantes ao utilizador
  - b. Classificação de experiências de outros utilizadores – poderá ordenar de forma crescente ou decrescente a listagem dos restaurantes com base na classificação atribuída.
  - c. Preço – ordenar de forma crescente ou decrescente os preços dos restaurantes abrangidos pela pesquisa efetuada

Se o utilizador não selecionar nenhum parâmetro, o *default* será Localização (isto é, aquele que se encontra mais perto dele).

4. Após selecionado um restaurante, poderá ver informações mais detalhadas acerca do mesmo, tais como fotografias, o tipo de comida, o preço médio praticado, classificação (Excelente, Muito Bom, Médio, Fraco, Terrível), comentários, a ementa, a localização assim como pedir a rota até ao restaurante pretendido.
5. O utilizador poderá visualizar o trajeto através da aplicação, que irá fornecer um mapa com o trajeto para o restaurante selecionado.

## 2.2.2 Com login efetuado

Podemos concluir que é imprescindível a possibilidade de uma autenticação no sistema por parte do utilizador, no entanto esta não é necessária para que o mesmo utilize a aplicação. Esta autenticação seguir-se-á pela apresentação da interface com que o utilizador em causa irá interagir. As permissões de todas as funcionalidades são geridas pelo motor de base de dados, sendo este o responsável por decidir se o utilizador pode usufruir delas ou não. Todas as funcionalidades acima descritas (sem login efetuado) estão acessíveis ao utilizador que efetuar login. Serão apresentadas novas funcionalidades ao utilizador que passaremos a descrever:

1. O utilizador pode efetuar login com a sua conta *Facebook* ou *Google* com um registo prévio, ou então com os dados fornecidos no ato de registo manual.
2. O sistema irá permitir que o utilizador faça uma publicação na sua rede social (*Facebook* ou *Google*) sobre a sua experiência de degustação, de forma simples e sem sair da aplicação.
3. No final da experiência, a aplicação permitirá que o utilizador faça uma avaliação da experiência realizada, assim como possibilitará que este marque o restaurante/prato como “Favorito” ou “Não Favorito” que será adicionado aos Favoritos ou Não Favoritos respetivamente nas preferências do utilizador.
4. Sempre que o utilizador, com login efetuado, realize uma avaliação à sua experiência gastronómica, o sistema irá guardar essa informação na base de dados;
5. Quando o utilizador pretender visualizar o seu histórico de avaliações, o sistema em *backOffice* irá trazer para *frontOffice* o resultado da informação adicionada previamente.
6. O utilizador, com login efetuado, poderá adicionar/remover um restaurante/prato aos seus favoritos ou não favoritos.
7. O utilizador, com login efetuado, poderá visualizar a lista dos seus restaurantes/pratos “Favoritos” assim como os “Não Favoritos”.
8. No final da experiência, o utilizador poderá classificar/comentar a sua experiência de degustação no local escolhido bem como marcar o restaurante/prato como “Favorito” ou “Não Favorito”.

9. Logo que o utilizador pretenda ver a lista dos seus “Favoritos” ou “Não Favoritos”, o sistema em *backOffice* irá trazer para *frontOffice* o resultado dos restaurantes adicionados anteriormente.
10. Relativamente aos dados fornecidos pelo utilizador acerca da experiência (p.ex. avaliação/comentários) estes serão armazenados na base de dados do *backOffice*;
11. O utilizador terá acesso ao histórico das suas avaliações, fotografias e comentários efetuados.

## 2.3. Módulo Preferências do Utilizador

Neste módulo serão tratados os filtros de pesquisa para cada utilizador autenticado. O filtro será baseado nas preferências do utilizador. Para cada restaurante ou prato presente no conjunto de “Não Favorito” será automaticamente removido de pesquisas futuras. Assim como serão removidos os pratos com os ingredientes não favoritos de cada utilizador.

Um possível problema será a remoção de um prato que contém um ingrediente que o utilizador não gosta, porém, o prato poderá ser apreciado pelo mesmo utilizador. (Por exemplo: um utilizador não gosta de pimento, porém gosta da pizza que contém pimento)

1. Haverá a possibilidade do utilizador poder adicionar e/ou remover um restaurante/prato dos seus Favoritos/Não Favoritos, para tal, haverá um botão específico para cada operação.
2. Assim que o utilizador adicionar um restaurante/prato aos seus “Favoritos”, o *backOffice* irá colocar o mesmo na base de dados.
3. Assim que o utilizador adicionar um restaurante/prato aos seus “Não Favoritos”, o *BackOffice* irá excluir esses itens de pesquisas futuras.
4. O utilizador poderá descrever quais os ingredientes que não gosta, a aplicação em pesquisas futuras não deverá apresentar pratos com esses ingredientes.
5. O utilizador poderá restringir a pesquisa limitando o valor de preço de pratos, assim como a localização.
6. O utilizador poderá adicionar preferências extras, tais como adicionar um ingrediente que não gosta e em pesquisas futuras se esse ingrediente estiver presente será

excluído esse prato. Ou então se tem alergia a determinado ingrediente o processo será o mesmo.

7. Toda a informação relativa aos restaurantes e aos utilizadores devem estar armazenadas numa base de dados relacional presente no *backOffice*;

## **2.4. Levantamento de requisitos do Sistema e Proprietário**

O proprietário de determinado restaurante poderá atualizar a informação relativa às suas ementas expostas ao público. Para tal forma definidos alguns requisitos para o Proprietário:

1. Será gerado um email e uma palavra passe para cada restaurante que será fornecida aos proprietários.
2. Cada proprietário do restaurante deverá ter a si associado um e-mail que o redirecione para a página do seu restaurante.
3. Os proprietários poderão editar o campo das ementas, este que será visível para todos os utilizadores do sistema.
4. Os proprietários poderão atualizar fotografias relativas ao restaurante e pratos (ementa).

### 3. Modelos do sistema

#### 3.1. Diagramas UML: Diagramas de Use Case

##### 3.1.1 Diagrama de Use Case: BackOffice

De maneira a melhor representar e diferenciar a interação com o sistema global, foram desenvolvidos dois modelos de *Use Cases* distintos, um para o *Front-Office* e outro para o *Back-Office*. Cada um destes reflete a interação dos vários tipos de utilizadores com o respetivo subsistema.

Mostramos de seguida os casos de uso pertencentes ao Utilizador, este que pode efetuar várias tarefas, o sistema de BackOffice deverá permitir que o utilizador realize as seguintes tarefas na aplicação: Efetuar Registo, Pesquisar por termos, Avaliar Restaurantes, Adicionar novas informações ao sistema, remover informações por ele adicionadas e editar as preferências e ainda se pretender efetuar *login* e *logout*. De salientar que algumas funcionalidades só estarão disponíveis se o utilizador estiver autenticado.

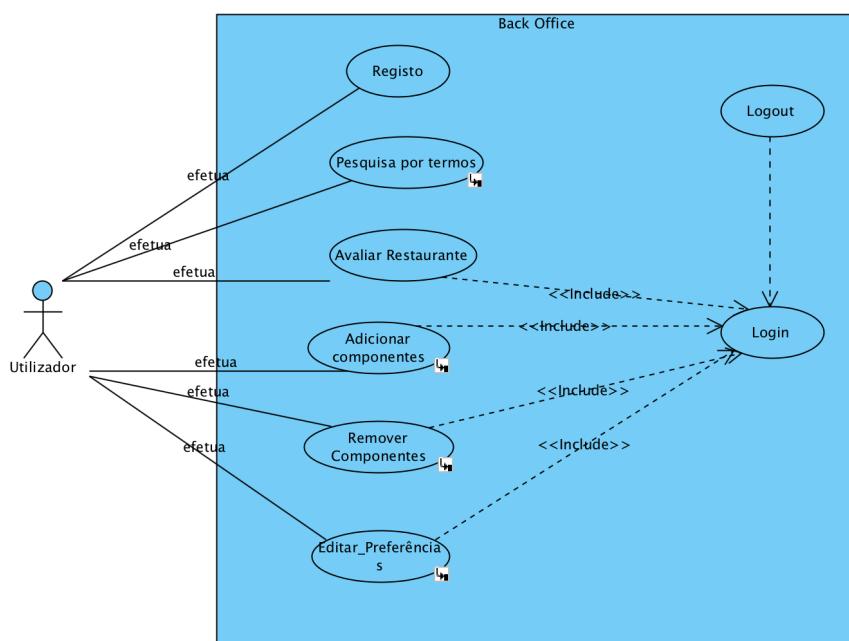


Figura 3 - Use Case: BackOffice

Por outro lado, temos os casos de uso que o Utilizador consegue aceder, isto são as ações que lhe permitem usufruir da aplicação, este é um caso de uso simplista, pois as ações permitidas pelo frontOffice foram implementadas no backOffice.

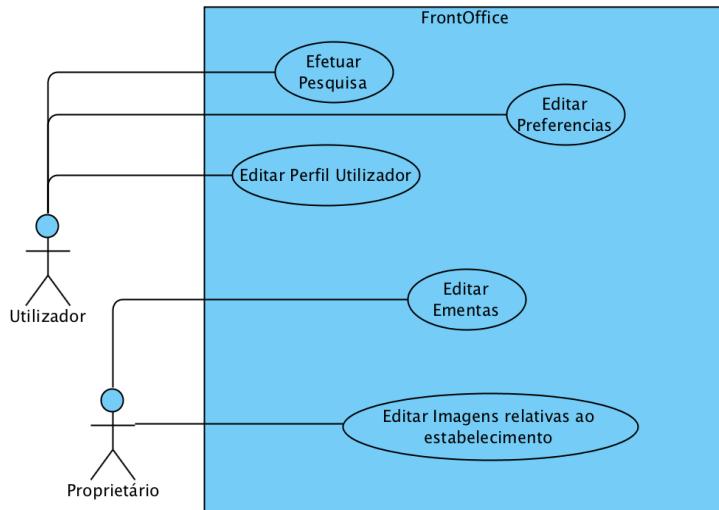


Figura 4 - Use Case: *FrontOffice*, ações permitidas aos utilizadores (proprietário e utilizador)

## Especificação de Use Case: Efetuar Registo

O caso de uso “Efetua Registo” permite que um utilizador normal efetue um registo na aplicação. Será pedido um nome, email e palavra passe.

**Pré-Condição:** Neste caso não há nenhuma pré condição

**Pós-Condição:** O utilizador fica com as credenciais de registo, para uso posterior

|   | Efetuar Registo | Actor Input                    | System Response                              |
|---|-----------------|--------------------------------|--|
| Flow of events  | 1               |                                | Pede para inserir Nome Completo              |
|   | 2               | Insere Nome                    | Pede para Inserir Email                      |
|   | 3               | Insere Email                   | valida email                                 |
|   | 4               |                                | Pede para Inserir Palavra passe              |
|   | 5               |                                |  |
|   | 6               |                                |  |
|   | 7               | Insere Palavra passe           | Pede para confirmar palavra passe            |
|   | 8               |                                |  |
|   | 9               | Insere Palavra passe novamente | Verifica se as palavras passes coincidem     |
|   | 10              |                                | Regista os dados                             |
|   | 11              |                                | Registo Efetuado com sucesso                 |
|   | 12              |                                |  |
| Exceção 1<br>[Email Inválido]<br>(passo 5)              |                 | Actor Input                    | System Response                              |
|   | 1               |                                | Informa que o email é inválido               |
| Exceção 2<br>[Palavra passe não<br>coincide] (passo 10) |                 | Actor Input                    | System Response                              |
|   | 1               |                                | Informa que as palavras passes não coincidem |

Figura 5 - Especificação de use case: Efetuar Registo

## Especificação de Use Case: Pesquisa por termos

O caso de uso de pesquisa por termos permite que um utilizador logado ou não efetue uma pesquisa no sistema.

**Pré-Condição:** Neste caso não há nenhuma pré condição, pois qualquer utilizador pode efetuar uma pesquisa.

**Pós-Condição:** Deverá ser apresentada uma lista com uma pesquisa filtrada

|  | Pesquisar por termos |  |  |
|--|----------------------|--|--|
| Flow of events                                   |                      | Actor Input                                      | System Response                            |
|  | 1                    | Pede para efetuar Pesquisa                       |  |
|  | 2                    | Visualiza conteúdo da pesquisa                   | Mostra opção de pesquisar por voz ou texto |
|  | 3                    | Escolhe uma das opções apresentadas              |  |
|  |                      | Actor Input                                      | System Response                            |
| Alternativa 1<br>[pesquisa voz]<br>(passo 2)     | 1                    |  | Aciona o microfone                         |
|  | 2                    | Transmite com a voz o input                      |  |
|  | 3                    |  | Converte para texto o input                |
|  | 4                    |  | Aciona filtros de pesquisa                 |
|  | 5                    |  | Devolve pesquisa encontrada                |
| Alternativa 2<br>[pesquisa teclado]<br>(passo 2) |                      | Actor Input                                      | System Response                            |
|  | 1                    |  | Aciona o teclado                           |
|  | 2                    | escreve através do teclado a pesquisa que deseja |  |
|  | 3                    |  | Aciona filtros de pesquisa                 |
|  | 4                    |  | devolve pesquisa encontrada                |

Figura 6 - Especificação do caso de uso: Pesquisar por termos

## Diagrama de Sequência: Pesquisa por termos

pesquisa por termos

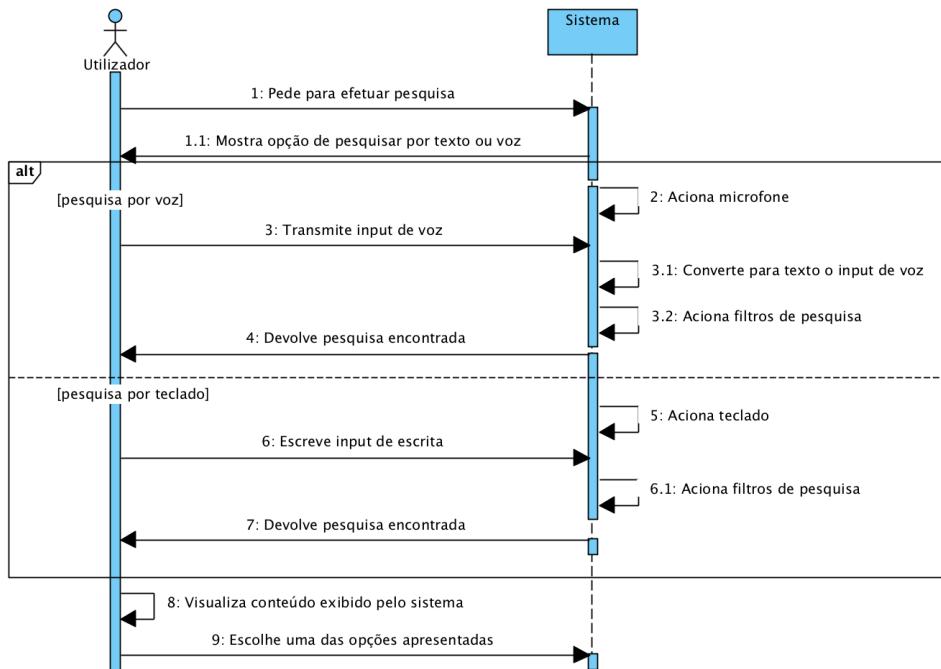


Figura 7 - Diagrama de sequência: Pesquisa por termos

## Especificação de Use Case: Avaliar Restaurante

O caso de uso de Avaliar Restaurante permite que um utilizador logado, deixe a sua avaliação relativa a determinado restaurante

**Pré-Condição:** O utilizador precisa de estar autenticado para que as informações fiquem registadas tanto no sistema como no seu histórico.

**Pós-Condição:** Deverá ser mostrada a avaliação aplicada a determinado restaurante

|   | Avaliar Restaurante | Actor Input                            | System Response                                    |
|---|---------------------|--|--|
| Flow of events                                | 1                   | Actor pede para avaliar restaurante    |  |
|   | 2                   |  | Disponibiliza opções para avaliar (1 a 5 estrelas) |
|   | 3                   | Insere o número de estrelas pretendido |  |
|   | 4                   |  | Regista informação                                 |
| Alternativa 1<br>[Cancela opção]<br>(passo 1) | 1                   | Actor Input                            | System Response                                    |
|   |                     |  | Volta para ação anterior                           |

Figura 8 - Especificação de Use Case: Avaliar Restaurante

## Especificação de Use Case: Editar Preferências

O caso de uso de Editar Preferências permite que um utilizador logado, edite as suas preferências.

**Pré-Condição:** O utilizador precisa de estar autenticado para que as informações fiquem registadas tanto no sistema como no seu histórico.

**Pós-Condição:** Deverá ser guardada toda a informação relativa às alterações efetuadas pelo utilizador

|   | Editar Preferências | Actor Input   | System Response                                  |
|---|---------------------|---|--|
| Flow of events  | 1                   | Pede para visualizar as Preferências                              |  |
|   | 2                   |   | Mostra as preferências atuais                    |
|   | 3                   | Escolhe a preferência que necessita de editar                     |  |
|   | 5                   |   |  |
|   |                     | Actor Input   | System Response                                  |
| Alternativa 1<br>[Altera favoritos]<br>(passo 3)        | 1                   | Pede para remover prato/restaurante dos favoritos                 |  |
|   | 2                   |   | Pergunta se tem a certeza que pretende remover   |
|   | 3                   | Diz que sim   | Remove prato/restaurante dos favoritos           |
|   |                     | Actor Input   | System Response                                  |
|   |                     |   |  |
| Excessão 1<br>[diz que não] (Alt1 - passo 3)            | 1                   | diz que não   |  |
|   | 2                   |   | volta ao passo 2 (mostra as preferencias atuais) |
|   |                     | Actor Input   | System Response                                  |
|   |                     |   |  |
|   |                     |   |  |
| Alternativa 2<br>[Altera outras Preferências] (passo 3) | 1                   | Pede para alterar ingrediente não gostado                         |  |
|   | 2                   |   | Disponibiliza campo editável                     |
|   | 3                   | Edita campo com novas informações<br>Pede para guardar alterações |  |
|   |                     | Actor Input   | Regista novas informações<br>System Response     |
|   |                     |   |  |
| Alternativa 3<br>[Filtrar preço] (passo 3)              | 1                   | Pede para filtrar pratos pelo preço                               |  |
|   | 2                   |   | Apresenta opções disponíveis                     |
|   | 3                   | Escolhe opção pretendida  |  |
|   |                     | Actor Input   | Regista novas informações<br>System Response     |
|   |                     |   |  |
| Alternativa 4<br>[Filtrar localização] (passo 3)        | 1                   | Pede para filtrar pesquisa de acordo com determinada localização  |  |
|   | 2                   |   | Apresenta opções disponíveis                     |
|   | 3                   | Escolhe opção pretendida  |  |
|   |                     | Actor Input   | Regista novas informações<br>System Response     |
|   |                     |   |  |
| Alternativa 5<br>[Filtrar tipo de comida] (passo 3)     | 1                   | Pede para filtrar tipo de comida a apresentar nas pesquisas       |  |
|   | 2                   |   | Apresenta opções disponíveis                     |
|   | 3                   | Escolhe opção pretendida  |  |
|   |                     | Actor Input   | Regista novas informações                        |
|   |                     |   |  |

Figura 9 - Especificação Use Case: Editar Preferências

## Diagrama de sequência: Editar Preferências

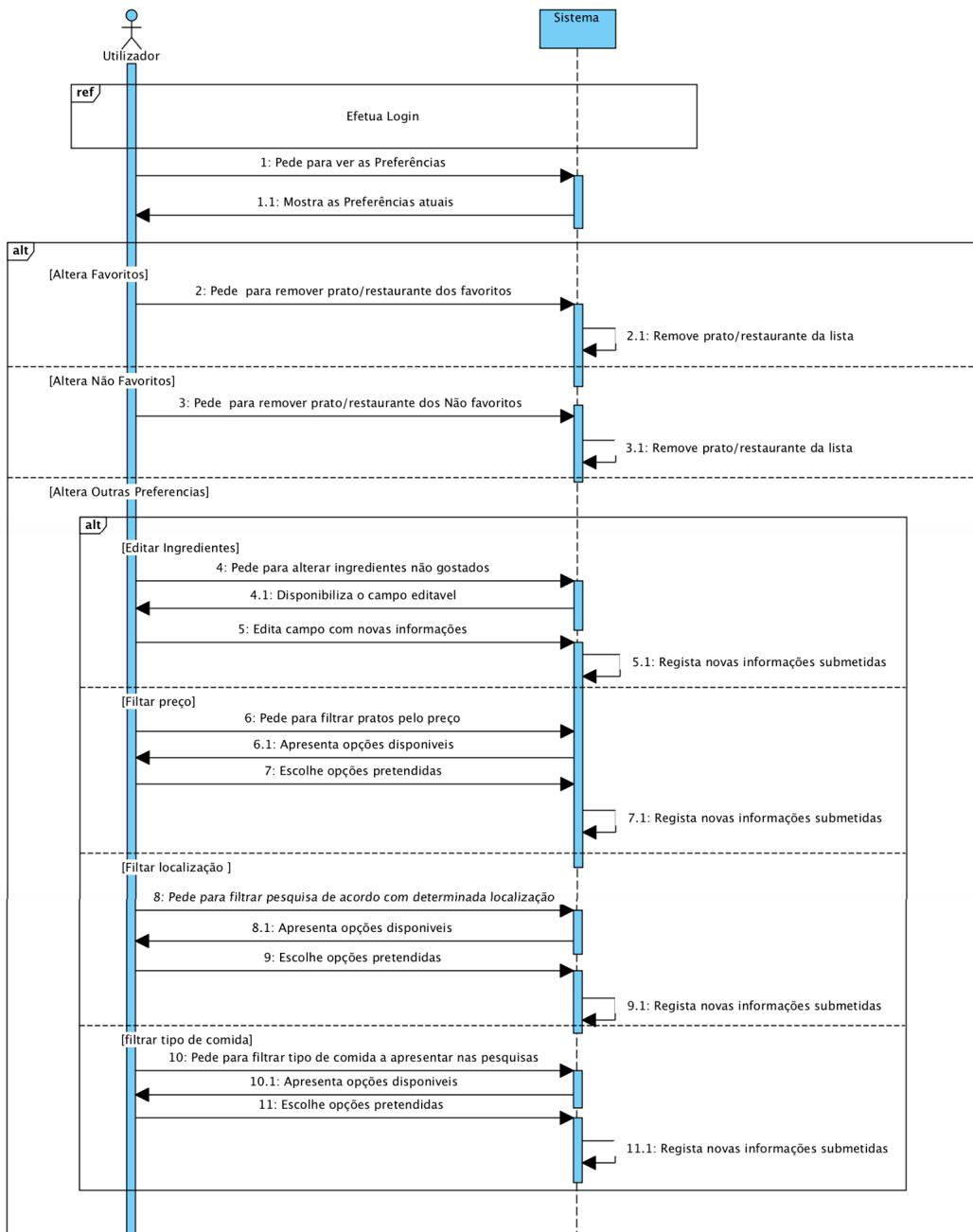


Figura 10 - Diagrama de Sequencia: Editar Preferências

## Subdiagrama Pesquisa Por Termos

De modo a simplificar a leitura dos casos de uso, decidimos separá-los por agregação de ações semelhantes: então a partir do momento que o utilizador realiza uma pesquisa, os resultados obtidos poderão ser traduzidos nas seguintes ações: Ver detalhes do restaurante, ver as fotografias do restaurante, ver menu do restaurante, ver opiniões, obter trajeto até destino.

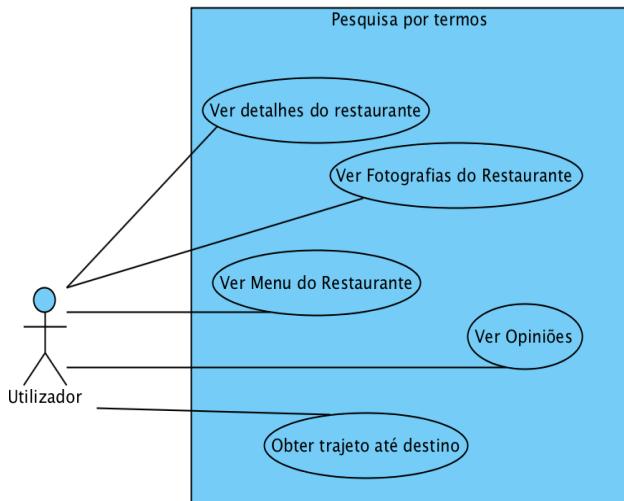


Figura 11 - Subdiagrama Use Case: Pesquisa Por Termos

## Subdiagrama Adicionar Componentes

O subdiagrama “Adicionar componente” mostra quais as componentes que poderão ser adicionadas pelos utilizadores da aplicação. Esta parte de adição de componentes é a mais importante na aplicação pois permitirá que a aplicação cresça e se mantenha atualizada.

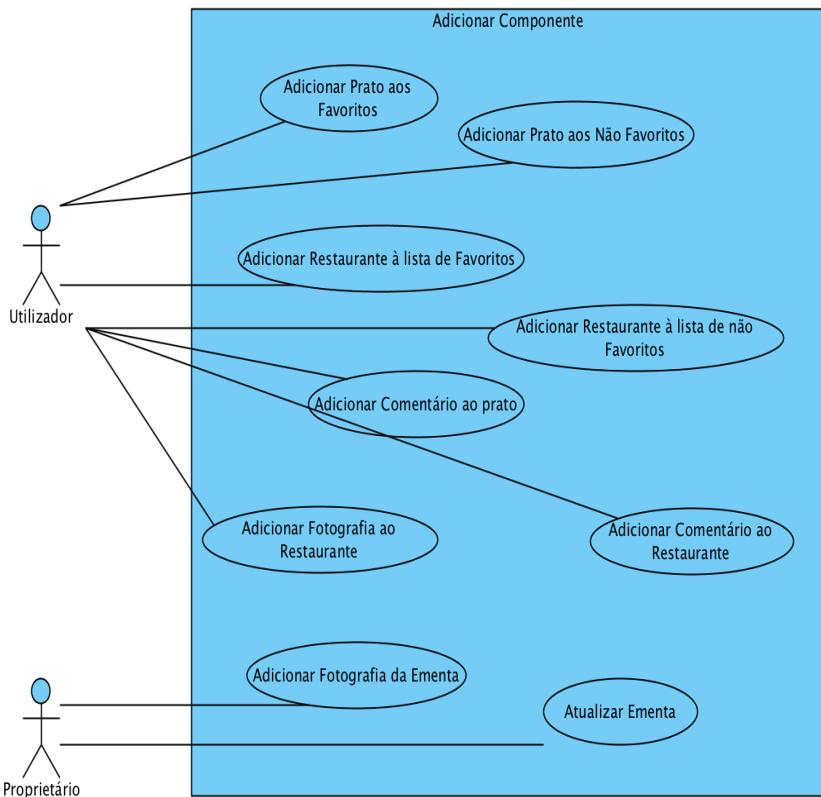


Figura 12 - Subdiagrama: Adicionar Componente

Resolvemos apenas mostrar uma especificação dos casos de uso, pois a ação de adicionar algo à aplicação será semelhante para todos os casos de inserção.

## Especificação de Use Case: Adicionar Restaurante à lista de Favoritos

O caso de uso de Adicionar Restaurante à lista de Favoritos permite que um utilizador logado, aceda ao seu histórico e a aplicação consiga a partir analisar o perfil do utilizador e sugerir outros restaurantes com base nas preferências adicionadas.

**Pré-Condição:** O utilizador precisa de estar autenticado para que as informações fiquem registadas tanto no sistema como no seu histórico.

**Pós-Condição:** Deverá ser mostrado o histórico dos restaurantes adicionados no separador de Restaurantes favoritos.

| Adicionar Restaurante aos favoritos        |   | Actor Input   | System Response  |
|--|---|---|--|
| Flow of events                             | 1 | Actor pede para adicionar restaurante aos favoritos |  |
|  | 2 |   | Sistema adiciona Restaurante à lista de Restaurante de Favoritos<br>Informa que restaurante foi adicionado aos Favoritos |
| Actor Input                                |   |   | System Response  |
| Alternativa 1<br>[Cancela opção] (passo 1) | 1 |   | Volta para o menu anterior   |

Figura 13 - Especificação de Use Case: Adicionar Restaurante à lista de Favoritos

## Subdiagrama Remover Componentes

O subdiagrama “Remover Componentes” mostra as ações que os utilizadores poderão executar quando pretenderem excluir informação do sistema. Isto permitirá que a informação se mantenha mais atualizada. Assim como atualizar as preferências do próprio utilizador.

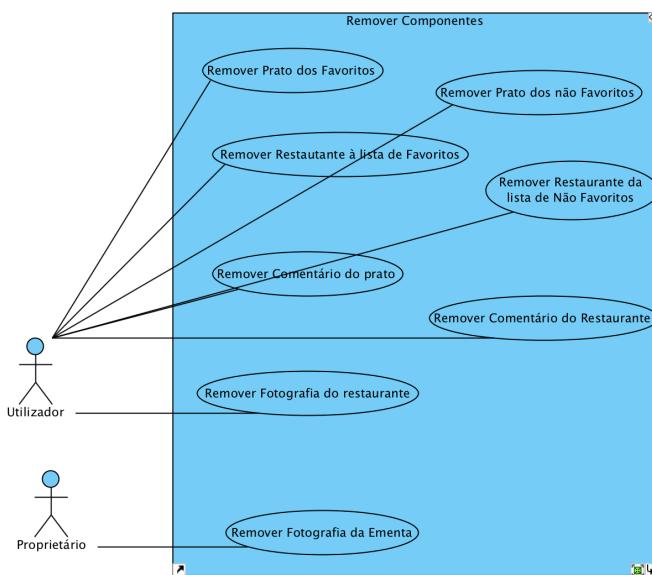


Figura 14 - Subdiagrama: Remover Componentes

Resolvemos apenas mostrar uma especificação dos casos de uso, pois a ação de remover algo da aplicação será semelhante para todos os casos de remoção.

## Especificação de Use Case: Remover Restaurante da lista de Favoritos

O caso de uso de Remover Restaurante da lista de Favoritos permite que um utilizador logado, aceda ao seu histórico. Esta ação de remoção permite que a aplicação atualize a informação de acordo com as preferências do utilizador.

**Pré-Condição:** O utilizador precisa de estar autenticado para que as informações fiquem registadas tanto no sistema como no seu histórico.

**Pós-Condição:** Deverá ser mostrado a lista dos restaurantes atualizada, isto é com as opções removidas no separador de Restaurantes favoritos.

| Remover Restaurante dos favoritos          |             |   |   |
|--|-------------|---|---|
| Flow of events                             | Actor Input |   | System Response   |
|  | 1           | Actor pede para remover restaurante aos favoritos | Sistema remove Restaurante da lista de Restaurante de Favoritos<br>Informa que restaurante foi removido da lista de favoritos |
|  | 2           |   |   |
| 3  |             |   |   |
| Actor Input                                |             | System Response                                   |   |
| Alternativa 1<br>[Cancela opção] (passo 1) | 1           |   | Volta para o menu anterior  |

Figura 15 - Especificação de Use Case: Remover Restaurante dos Favoritos

## Subdiagrama Editar Preferências

O subdiagrama “Editar Preferências” permite visualizar quais as preferências do utilizador que poderão ser editadas na aplicação, neste caso em específico poderá editar um campo com os ingredientes que não gosta, adicionalmente foram adicionados campos que permitem filtrar ainda mais a informação tais como filtrar preço do prato, localização e tipos de comida.

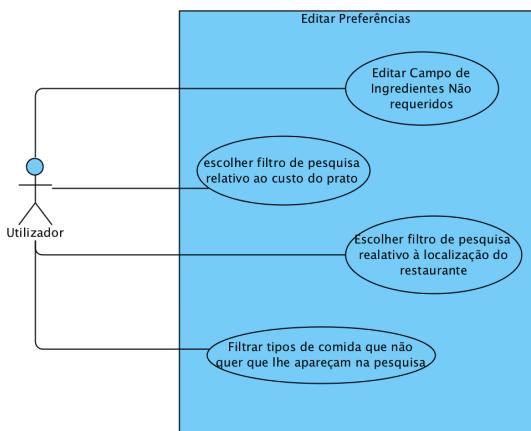


Figura 16 - Subdiagrama: Editar Preferências

## 3.2. Diagramas UML: Diagramas de Atividade

Os diagramas de atividade podem ter uma grande importância na especificação de sistemas. A sua utilidade no contexto deste projeto é relevante, visto que ajudam a especificar o comportamento dos elementos de software a desenvolver, o fluxo de dados que as várias funcionalidades geram, bem como o encadeamento dos processos envolvidos.

Como tal, de forma a complementar a informação já transmitida através dos outros tipos de modelos apresentados anteriormente, procedeu-se ao desenvolvimento de um diagrama de atividade para cada funcionalidade que o sistema suportará. Desta forma, facilitar-se-á a tarefa da equipa de desenvolvimento da peça de software, de forma a fornecer-lhe uma espécie de guião a seguir que levará o programa a corresponder ao pedido pelo cliente, conforme foi especificado na secção de análise de requisitos.

De forma a melhor descrever o processo de desenvolvimento deste tipo de diagramas, e ajudar a traduzir a informação neles contida, apresentam-se de seguida alguns exemplos desenvolvidos deste tipo de diagramas.

### 3.2.1 Diagrama de Atividade: Efetuar Pesquisa

A aplicação a ser desenvolvida será nada mais que um motor de pesquisa com a restrição que será uma pesquisa filtrada na área da restauração, com base no *profiling* dos utilizadores. Por forma a simplificar a atividade de pesquisa mostramos de seguida um diagrama de atividade da pesquisa efetuada através do input por voz ou teclado.

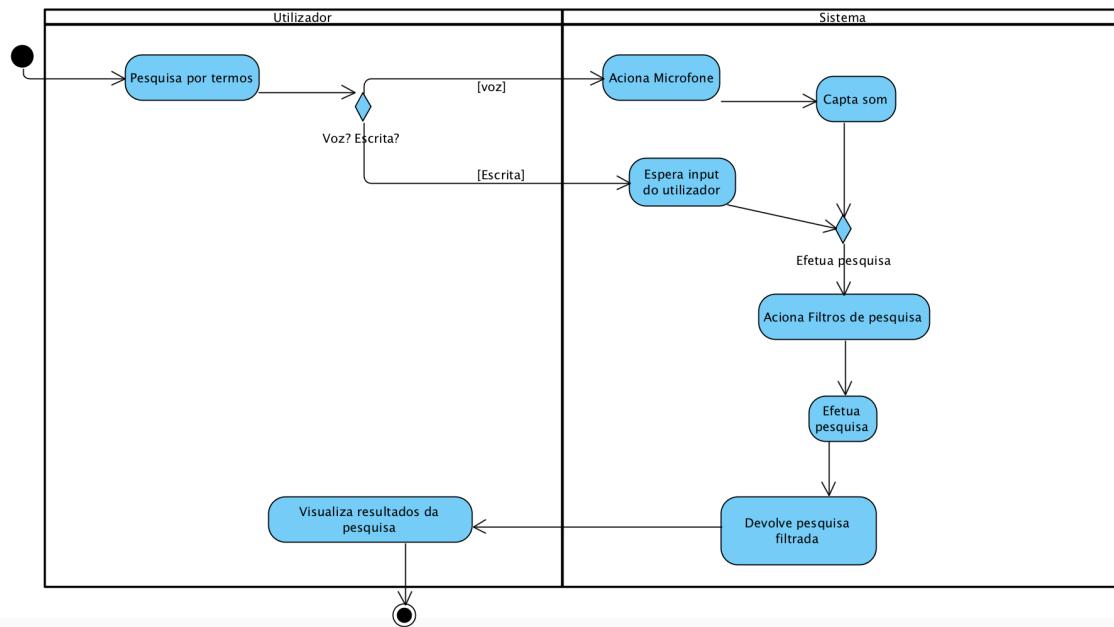


Figura 17 - Diagrama de Atividade: Efetuar pesquisa

### 3.2.2 Diagrama de Atividade: Editar Preferências

O diagrama Atividade “Editar Preferências” mostra as atividades que o utilizador pode efetuar depois de selecionar o campo das Preferências. Ser-lhe-á apresentado várias opções, podendo editá-las de acordo com o seu perfil. No final o sistema deverá guardar todas as alterações efetuadas pelo utilizador, para no futuro adicionar aqueles campos ao filtro de pesquisa.

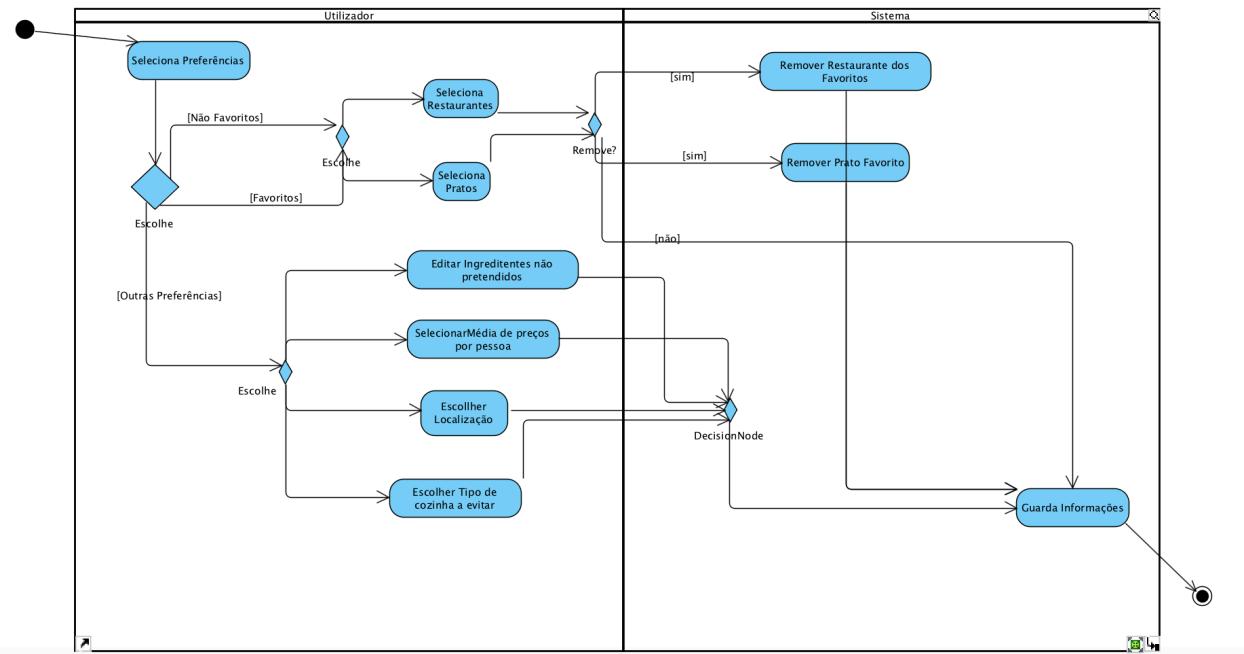


Figura 18 - Diagrama Atividades: Editar Preferências

### 3.3. Diagramas UML: Modelo de Domínio

Para ilustrar os conceitos e as relações que surgem no contexto do negócio do sistema de recomendação de restaurantes foi apresentado o Modelo de Domínio. Este modelo representa um esquema conceptual do problema em questão e nele podem ser vistas as entidades que têm participação no negócio do sistema, bem como alguns dos seus atributos e papéis que estas desempenham. Como tal, neste modelo não são visíveis funcionalidades concretas que o sistema deve apresentar, pelo que estas são abordadas nas secções próprias.

Na imagem abaixo pode ser visto o Modelo de Domínio mencionado.

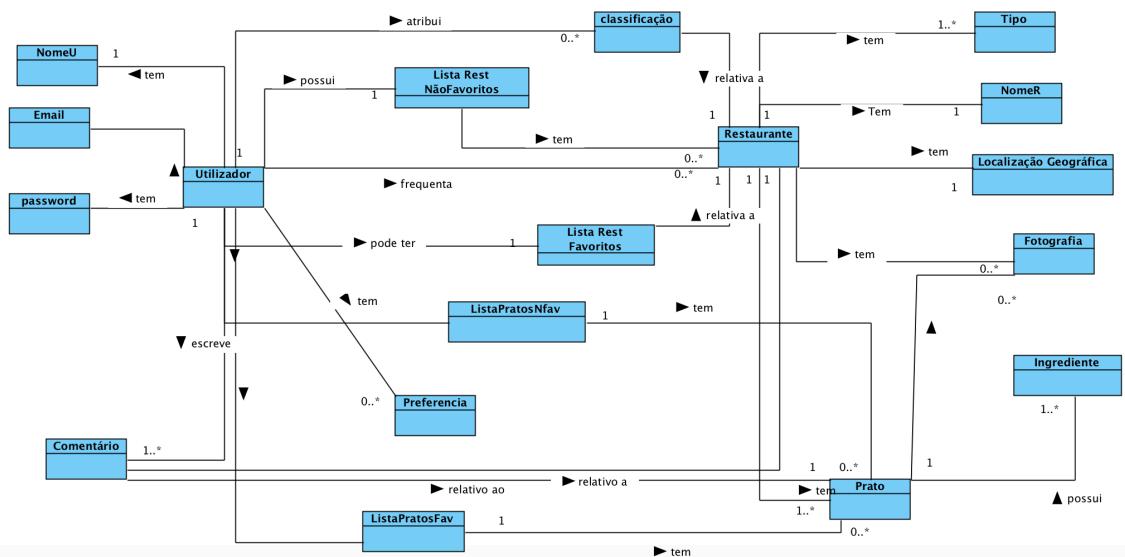


Figura 19 - Modelo de Dominio

### 3.4. Diagramas UML: Diagrama de Classes

De seguida, é apresentado o diagrama de classes. Este diagrama apresenta todas as classes, os seus respetivos atributos e métodos. Além disto, também são especificadas as relações que as classes têm umas com as outras. Para concluir, também é especificado nas relações o tipo de associação (agregação ou composição).

Para a realização deste diagrama foram analisados os requisitos e o modelo de domínio.

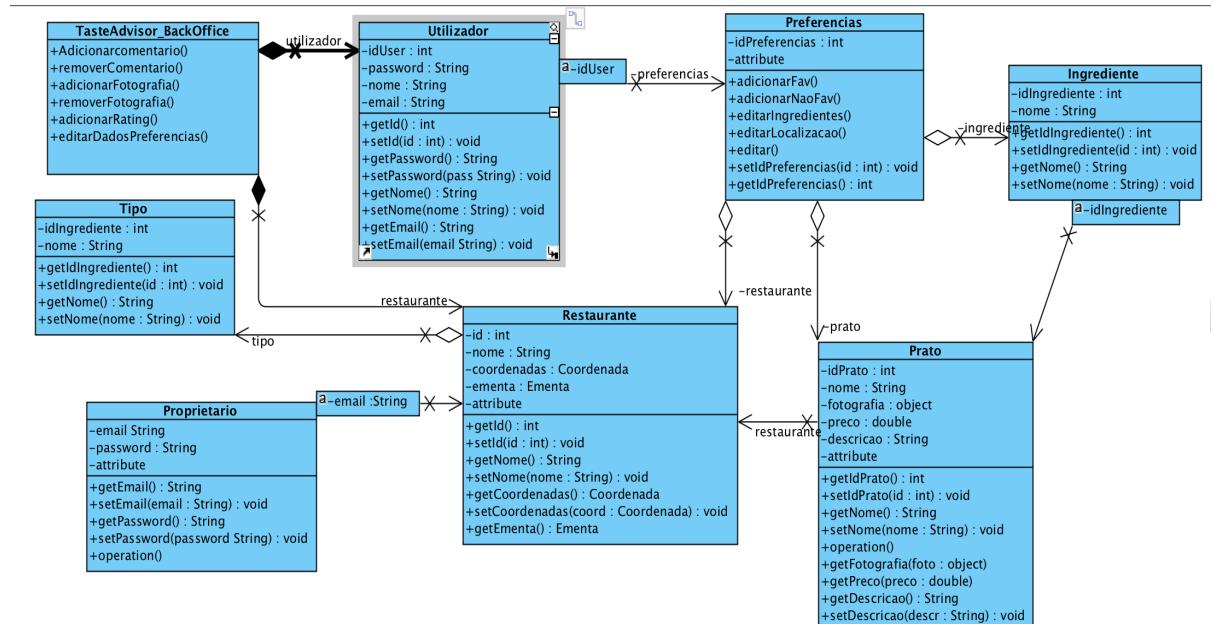


Figura 20 - Diagrama de Classes

## 4. Bases de Dados

### 4.1. Modelo Conceitual

#### 4.1.1 Identificar os tipos de entidades

Após a leitura dos requisitos foram identificadas 5 entidades:

| Nome da entidade | Descrição  | Sinónimos                    | Ocorrência  |
|------------------|--|------------------------------|---|
| Restaurante      | Esta entidade representa o estabelecimento que irá servir os produtos gastronómicos                | Estabelecimento gastronómico | Restaurante é das entidades mais importantes pois são o local onde se executa o processo de degustação  |
| Utilizador       | Esta entidade representa o cliente que irá usufruir da aplicação e                                 | Cliente                      | Utilizador é outra das entidades mais relevantes do problema. É esta entidade que irá usufruir dos serviços disponibilizados pela aplicação               |
| Prato            | Esta entidade representa a comida servida nos restaurantes presentes na base de dados da aplicação | Comida                       | O Prato é o produto que irá ser degustado pelo Utilizador. É uma das entidades mais importantes pois o objetivo principal da aplicação é encontrar pratos |
| Tipo             | Esta entidade representa o género de comida que o restaurante serve                                | Género                       | O Tipo é a natureza/género da comida servida no restaurante   |
| Ingrediente      | Esta entidade representa os vários elementos que compõem um prato.                                 | Elemento                     | O Ingrediente é o elemento de comida presente nos vários pratos   |

Tabela 1 – Tabela relativa às Entidades

#### 4.1.2 Identificar tipos de relacionamento

| Nome Entidade | Multiplicidade | Relacionamento         | Multiplicidade | Nome Entidade |
|---------------|----------------|------------------------|----------------|---------------|
| Restaurante   | (1,N)          | Tem                    | (1,N)          | Tipo          |
| Restaurante   | (1,N)          | Ementa                 | (N,N)          | Prato         |
| Utilizador    | (0,N)          | Restaurantes Visitados | (0,N)          | Restaurante   |
| Utilizador    | (0,N)          | Pratos Degustados      | (0,N)          | Prato         |
| Prato         | (N,N)          | Receita                | (1,N)          | Ingrediente   |

Tabela 2 - Tabela de relacionamentos

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

Neste tópico serão abordados todos os atributos que compõem as entidades e os relacionamentos. A escolha dos atributos foi realizada com base numa análise dos requisitos a fim de detetar todos os atributos necessários para representar cada entidade e relacionamento no sistema.

#### Associação entre atributos e entidades/relacionamentos

| Entidade    | Atributo              | Descrição                                 | Tipo e tamanho | Nulo | V.P.D |
|-------------|-----------------------|---|----------------|------|-------|
| Restaurante | <u>Id_Restaurante</u> | Identifica inequivocamente um Restaurante | nchar(45)      | Não  | -     |
|             | Nome                  | Nome do Restaurante                       | nchar(45)      | Não  | -     |
|             | Coordenadas           | Local geográfico                          | float          | Não  | -     |

|             |                       |   |           |     |   |
|-------------|-----------------------|---|-----------|-----|---|
| Utilizador  | <u>Id_User</u>        | Identifica inequivocamente um Utilizador          | nchar(45) | Não | - |
|             | Password              | Palavra chave para login                          | nchar(45) | Não | - |
|             | Nome                  | Nome do Utilizador                                | nchar(45) | Não | - |
|             | Email                 | Email do utilizador                               | nchar(45) | Não | - |
| Prato       | <u>Id_prato</u>       | Identifica inequivocamente um prato               | nchar(45) | Não | - |
|             | Descrição             | Descrição do prato                                | nchar(45) | Não | - |
|             | Preco                 | Preço do prato                                    | float     | Não | - |
|             | Nome                  | Foto  | varbinary | Não | - |
| Ingrediente | <u>Id_Ingrediente</u> | Identifica inequivocamente um ingrediente         | nchar(45) | Não |   |
|             | Nome                  | Nome do Ingrediente                               | nchar(45) | Não |   |
| Tipo        | <u>Id_Tipo</u>        | Identifica inequivocamente um tipo de restaurante | nchar(45) | Não |   |
|             | Nome                  | Nome do tipo de restaurante                       | nchar(45) | Não |   |

Tabela 3 - Tabela de Associação entre atributos e entidades

#### 4.1.4 Desenho do diagrama ER

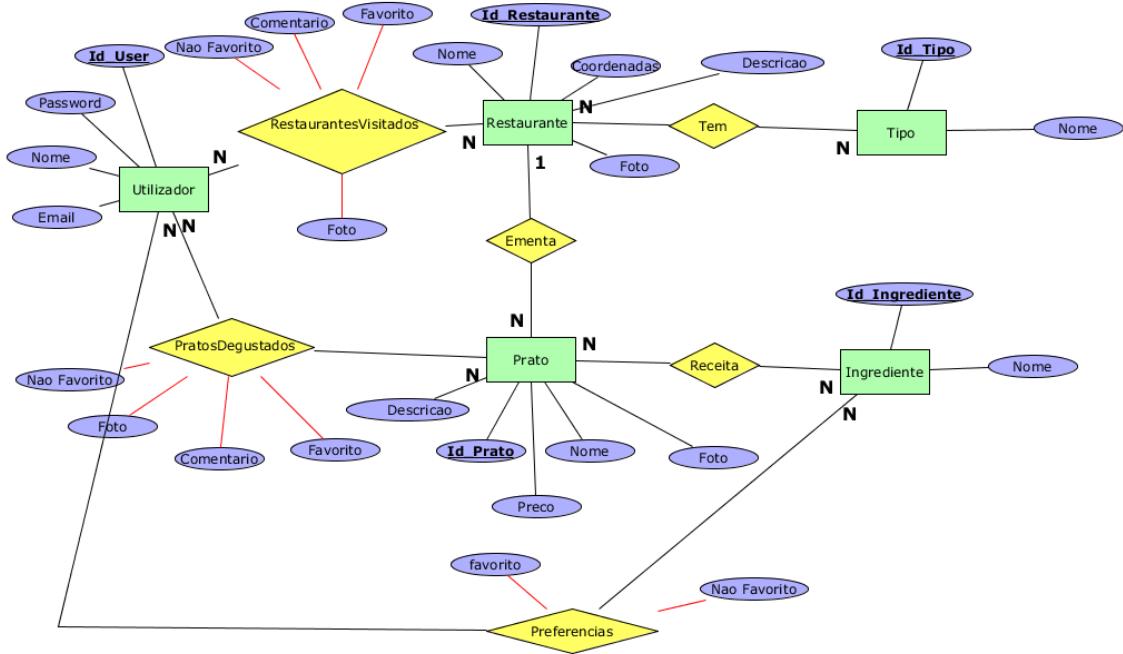


Figura 21 - Modelo conceitual

#### 4.1.5 Revisão do modelo de dados com o utilizador

Após a conceção do modelo conceptual, este foi revisto junto do utilizador. Dado que nenhum erro ou problema foi encontrada, este modelo foi aceite.

## 4.2. Modelo Lógico

### 4.2.1 Validação do modelo lógico através de normalização

#### 1FN - 1<sup>a</sup> forma normal:

Para aplicar a 1FN numa tabela desnormalizada, é necessário identificar os atributos ou grupos de atributos repetidos. Um atributo, ou grupo de atributos repetidos, é aquele que ocorre numa tabela várias vezes para uma mesma chave primária.

Sendo que as regras da primeira forma normal já foram previamente aplicadas e que todos os atributos já foram devidamente identificados, resta-nos apenas verificar que todos os atributos presentes nas relações se tratam de atributos atómicos.

Como o nosso modelo relacional não possui atributos multi-valor nem grupos repetidos, concluímos que respeita a 1<sup>a</sup>Forma Normal.

#### 2FN – 2<sup>a</sup> Forma Normal:

Para que a segunda forma normal seja verificada é necessário que o modelo de dados se encontre já na primeira forma normal e que os atributos não-chave de relações que contenham chaves primárias compostas sejam funcionalmente dependentes da totalidade da chave, ou seja, não podem existir atributos numa relação que sejam dependentes apenas de uma parte da chave, pois caso isso se verificasse, o modelo iria sofrer de anomalias de atualização.

Neste caso como também já foram tomadas considerações acerca das dependências funcionais nas relações, também já não é possível encontrar no modelo casos em que a segunda forma normal seja desrespeitada.

Assim como os atributos nessa tabela são dependentes da totalidade da chave, a segunda forma normal já se verifica.

#### 3FN – 3<sup>a</sup> Forma Normal:

Para que uma relação se encontre na terceira forma normal, é necessário que se encontre já na segunda forma normal e que para além disso não existam nela atributos não- chave-primária que sejam transitivamente dependentes da chave primária, ou seja, não pode existir um atributo que seja ao mesmo tempo funcionalmente dependente da chave primária e de um outro atributo. Assim, para que seja verificada a terceira forma normal, é necessário eliminar as dependências transitivas presentes nas relações do modelo. Sendo assim, concluímos que todas as relações respeitam a 3<sup>a</sup> Forma Normal pois todos os atributos não-chave dependem, inteira e exclusivamente, da totalidade da chave.

Após a verificação de todas as relações do nosso modelo relacional, concluímos que este se encontra normalizado até à 3<sup>a</sup>Forma Normal.

#### 4.2.2 Elaboração e validação do esquema lógico da base de dados

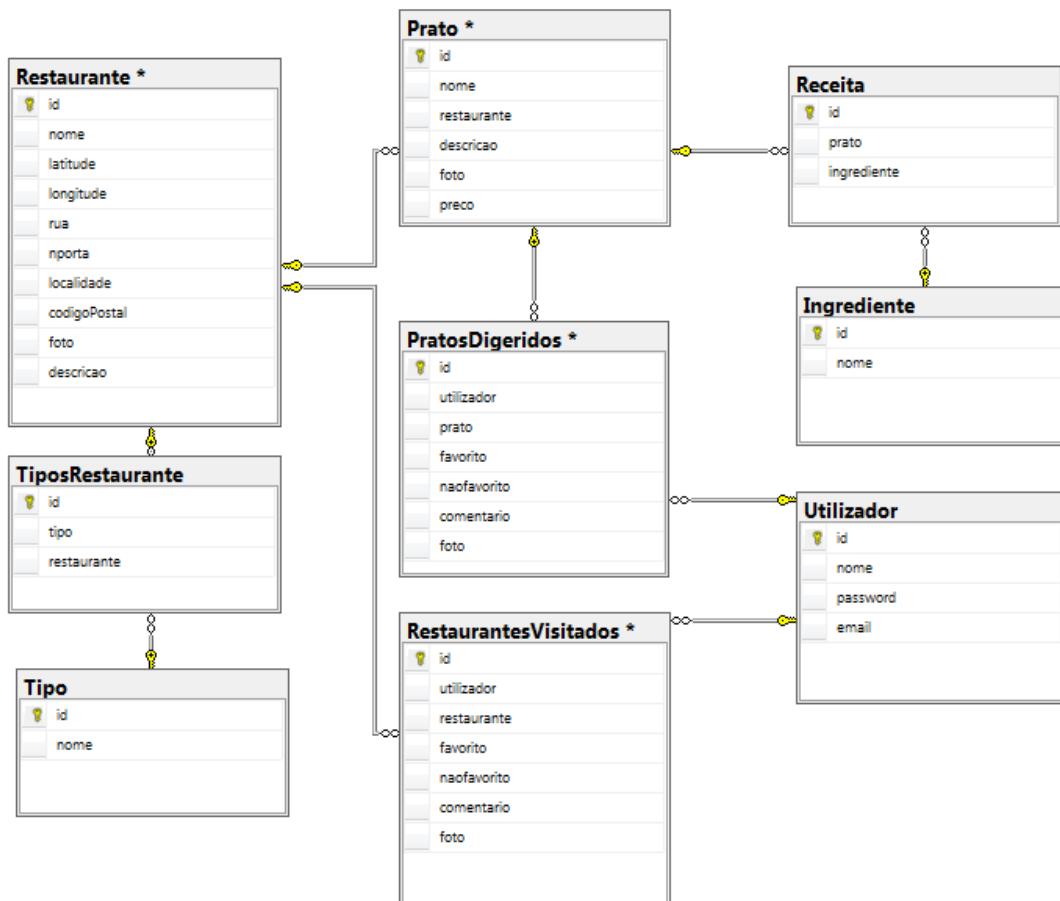


Figura 22 - Modelo Lógico

#### **4.2.3 Definição do tamanho inicial da base de dados e análise do seu crescimento futuro**

Tendo em conta os requisitos do sistema previamente analisados temos que inicialmente a base de dados terá:

- Cerca de 7000 Restaurantes
- Cerca de 42000 Pratos
- Cerca de 168000 Ingredientes
- Cerca de 7000 Utilizadores (contas gerentes)
- Cerca de 20 Tipos

Após o início da utilização do sistema de recomendação prevemos que a base de dados cresça em pouco tempo. Pois esperamos que em média sejam criadas 50 novas contas por mês, adicionados 7000 comentários por mês, 6000 fotografias novas de restaurantes e pratos e também 5000 avaliações de restaurantes. Em termos do crescimento de pratos, ingredientes e tipos não prevemos que estas entidades cresçam muito mais do que aquilo que já existe. Eventualmente podem ser removidos uns e adicionados outros, mas nada muito relevante no crescimento da base dados.

#### **4.2.4 Revisão do modelo lógico final com os futuros utilizadores**

Para finalizar esta fase, o modelo conceptual desenhado deve ser revisto com o utilizador. Este processo é extremamente importante na medida em que o utilizador deve ser capaz de reconhecer, assinando, que o modelo de dados concebido é uma representação real dos requisitos da empresa a ser modelada. Este passo é também uma forma de avaliar o trabalho feito face aos requisitos do utilizador. Foi avaliada toda a documentação associada ao modelo de dados, ou seja, o dicionário de dados de entidades, o dicionário de dados de relacionamentos, o dicionário de dados de atributos e o diagrama ER. Na revisão do dicionário de entidades foi possível identificar todas as entidades importantes no modelo e constatar com o cliente que correspondem aos objetos considerados importantes no problema e às necessidades da empresa. Na revisão do dicionário de relacionamentos foi justificada a forma como as entidades estão associadas e o utilizador verificou que estes relacionamentos correspondem à realidade do problema. Já na revisão do dicionário de atributos o utilizador pode verificar mais concretamente a informação associada a cada entidade e validar o domínio

de cada atributo, constatando que estes correspondem aos requisitos anteriormente apresentados. Por fim, na revisão do diagrama ER foi possível obter uma visão geral do modelo de dados e realizar a verificação de algumas das perguntas mais frequentes no problema de modo a testar as capacidades do mesmo. O modelo de dados foi aprovado pelo cliente '*Belo Lda*' como sendo uma solução que dá resposta a todas as necessidades da empresa.

O objetivo desta revisão foi permitir ao cliente detetar eventuais problemas ou falhas que possam causar complicações no futuro. Após esta revisão, concluiu-se que o problema em questão se encontrava representado corretamente no modelo lógico construído.

## 5. Mockups

Os Mockups servem para mostrar uma representação do design do possível produto final, permitem avaliar um produto mesmo antes de estar pronto, este é um benefício pois permitirá ao cliente avaliar de uma forma superficial se o que pediu está de acordo com o desenhado.

Passaremos de seguida a mostrar o possível funcionamento da aplicação através dos mockups:

Inicialmente será apresentada uma página que permite que um Utilizador (com sessão iniciada ou não) inicie a aplicação “Taste Advisor”, a primeira interface que lhe é apresentada é a página de pesquisa. Aqui é possível observar que a aplicação quando aberta oferece sugestões de restaurantes quando a aplicação é iniciada. Após a pesquisa efetuada pelo utilizador será possível ordenar a pesquisa por localização, classificação ou custo. A aplicação permitirá efetuar uma pesquisa através da recolha de input por voz ou teclado.

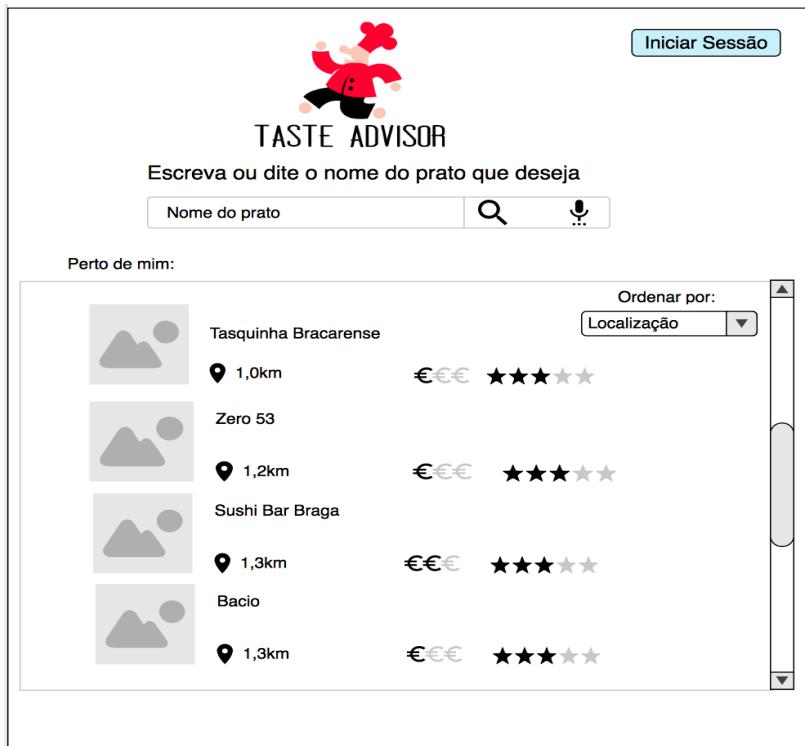


Figura 23 - Mockup: Página Inicial

Após o utilizador selecionar um restaurante com o prato requerido, ser-lhe-á apresentada uma página onde poderá ver mais detalhes sobre o restaurante. Tais como a localização, prato e ingredientes, fotografias e comentários relativos ao prato ou restaurante. A qualquer momento poderá voltar para o menu anterior, efetuar outra pesquisa ou se desejar iniciar sessão.

**Nome Restaurante**

Geral    Menu    Opiniões    Fotografias

Número de Telefone: 252666777

Custo: 15€ / pessoa (aprox.)

Tipo de cozinha: Tradicional, Churrasco

Avaliação: ★★★★★

Localização: Rua dos Peões, São Victor, Braga 4710-416

Avalie o restaurante: ★★★★★

**Nome Restaurante**

Geral    Menu    Opiniões    Fotografias

Franceseinha

Ingredientes: bife de vaca, bacon, ovo, batata frita

Preço: 12€

Recolhido pela TasteAdvisor no restaurante / Recebido por email/ Menu carregado pelo restaurante. Preços sujeitos a alteração sem aviso prévio. Confirmar os preços antes de visitar ou de encotrar algo neste espaço.

Comentários:

Excelente 10:31 AM >  
Este prato estava fenomenal, comida muito boa!!!! <3

**Nome Restaurante**

Geral    Menu    Opiniões    Fotografias

Avalie o restaurante: ★★★★★

Comentários:

Deixar Comentário:

Escreva o seu comentário relativo ao funcionamento do restaurante:

José 10:31 AM >  
Adorei o restaurante, comida muito boa e funcionários muito prestáveis

Carlos 9:46 AM >  
A comida estava boa, o atendimento foi normal

Anita Yesterday >  
Nada de extraordinário

**Nome Restaurante**

Geral    Menu    Opiniões    Fotografias

Figura 24 - mockup: Separador "Geral", separador "Menu", separador "Opiniões" e "Fotografias"

A aplicação permite também que o utilizador efetue login, para tal também foi criada um página específica para esse efeito. O utilizador que não pretender inserir as suas credenciais pode fazê-lo de forma automática através do facebook ou google, porém é necessário um registo prévio para poder fazer isso de forma automática.

Passaremos então a mostrar os mockups de Inicio de sessão:

O mockup mostra a interface de login da TASTE ADVISOR. No topo, há uma ilustração de uma figura com chapéu vermelho e avental, seguida pelo nome da empresa "TASTE ADVISOR". Abaixo, o formulário de login com campos para "Endereço de Email" e "Palavra-passe", e um botão "Entrar". A direita, uma seção intitulada "Iniciar sessão com:" oferece opções para logar via Facebook ou Google, com uma opção "OU" entre elas. Abaixo do formulário, há links para "Criar Conta" e "Cancelar".

Figura 25 - Mockup: Inicio de Sessão

E de seguida o de registo no sistema:

O mockup mostra a interface para a criação de uma nova conta. No topo, a ilustração da figura e o nome "TASTE ADVISOR" seguidos por "Criar Conta". Abaixo, campos para "Insira o nome:", "Insira o endereço de e-mail:" e "Escreva uma palavra-passe:", cada um com seu respectivo placeholder e ícone. A direita, uma seção intitulada "ou" oferece opções para logar via Facebook ou Google. Abaixo do formulário, há links para "Criar Conta" e "Cancelar".

Figura 26 - Mockup: Criação de conta

Após o registo e iniciação de sessão na aplicação, o utilizador poderá fazer tudo o podia antes, porém agora tem a opção de editar as suas preferências de acordo com o seu perfil. Para tal foi criada uma nova janela de mockup que permite ao utilizador indicar ao sistema quais as suas reais preferências. Mostrámos de seguida os mockups criados para o efeito:

Neste mockup o utilizador poderá remover da sua lista de preferências os restaurantes que adicionou anteriormente à lista:

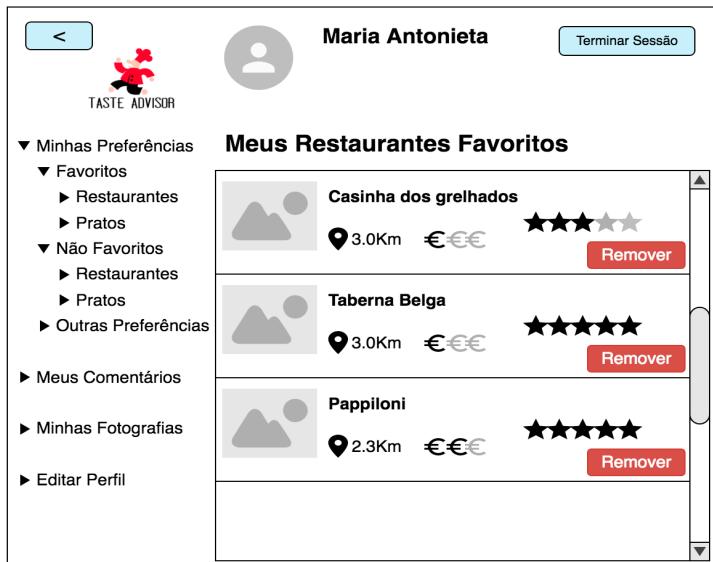


Figura 27 - Mockup: Ver Restaurantes Favoritos e remover

Nesta página podem ser visualizados quais os pratos que estão na lista não favoritos de cada utilizador, estes serão usados no futuro para filtrar ainda mais a pesquisa, fazendo com que estas informações não apareçam em pesquisas futuras.



Figura 28 - Mockup: Ver pratos não Favoritos adicionados à lista de preferências

Foram também adicionadas “Outras Preferências” que permite que cada utilizador edite os dados de acordo com o seu perfil. Foram criados campos que permitem editar informação adicional dos ingredientes que o utilizador não gosta ou é alérgico, fazendo com que assim esses alimentos sejam eliminados de pesquisas futuras. Adicionalmente também podem ser adicionados filtros que permitirão reduzir ainda mais a pesquisa. O utilizador poderá selecionar restaurantes baseados no tipo de comida, localização e preços médios por pessoa.

The screenshot shows a mobile application interface for 'TASTE ADVISOR'. At the top, there's a back button (<), a user profile icon (a person with a red bow), the name 'Maria Antonieta', and a 'Terminar Sessão' (End Session) button. Below the header, on the left, is a sidebar menu with the following items:

- ▼ Minhas Preferências
  - ▼ Favoritos
    - ▶ Restaurantes
    - ▶ Pratos
  - ▼ Não Favoritos
    - ▶ Restaurantes
    - ▶ Pratos
  - ▶ Outras Preferências
- ▶ Meus Comentários
- ▶ Minhas Fotografias
- ▶ Editar Perfil

The main content area is titled 'Editar Minhas Preferências' (Edit My Preferences). It contains several configuration sections:

- Ingredientes que não gosto:** A text input field containing 'grelhos' and 'salchicha' with an 'editar' (edit) link next to it.
- Média de preços por pessoa:** Radio buttons for price range selection:
  - Menos de 10€
  - Entre 10 e 25 €
  - Mais de 25 €
- Localização:** Radio buttons for distance from location:
  - Menos de 10 km
  - Entre 10 e 25 km
  - Mais de 25 km
- Tipo de cozinha a evitar:** A dropdown menu labeled 'Escolher tipo de cozinha' (Select kitchen type) with a '▼' arrow, currently showing 'Chinesa'.

## **6. Conclusões e Trabalho Futuro**

Terminada a primeira etapa, damos por concluído a fundamentação deste projeto. Inicialmente demos especial ênfase à recolha de informações juntamente com o cliente de modo a compreender, da melhor forma possível, as suas necessidades. Adquirida esta base de conhecimento tornaram-se claras as motivações e objetivos do sistema a implementar. Assim sendo, concluímos que um investimento de trabalho nesta parte é recompensado no futuro, uma vez que uma compreensão detalhada do que constitui o problema em causa permite uma idealização e implementação mais eficazes.

No que diz respeito à segunda fase de desenvolvimento do projeto, que consistiu na especificação do sistema, foram levantados os vários tipos de requisitos e elaborada a modelação do sistema através de diferentes tipos de diagramas em UML.

Relativamente ao levantamento de requisitos, procedeu-se à separação dos mesmos em módulos para um melhor agrupamento de funcionalidades. Esta organização permitiu traduzir de melhor forma o que foi pedido pelo cliente e facilitar o entendimento das funcionalidades que o sistema deve fornecer, bem como a maneira como as deve fornecer.

Na elaboração dos modelos em UML, foram desenvolvidos vários tipos de diagramas, sendo que cada um dos tipos de diagramas escolhidos refletem aspectos diferentes sobre o projeto a realizar. O modelo de domínio, por exemplo, permitiu perceber quais os elementos que vão constituir o sistema. Os modelos de Use Cases permitem visualizar as funcionalidades a implementar e quais os utilizadores que irão usufruir das mesmas. Já os diagramas de sequência permitem refinar a descrição efetuada em cada Use Case, permitindo ter uma percepção dos vários elementos que intervêm em cada funcionalidade, bem como o ordenamento temporal das mensagens trocadas por estes elementos. A elaboração dos diagramas de atividade permitiu adicionar ainda mais informação ao que foi transmitido através dos diagramas de sequência, sendo demonstrado o fluxo de dados e especificada cada ação envolvida em cada funcionalidade. Por fim o diagrama de Classes permite obter uma perspetiva sobre a arquitetura do sistema a desenvolver, já refletindo o paradigma de programação que vai ser utilizado.

Com os modelos de sistema construídos procedeu-se à idealização e realização da base de dados em que foi feita uma análise aos requisitos levantados previamente, construindo-se assim o modelo conceitual. Com o modelo conceitual construído e validado pelo utilizador procedeu-se à passagem para o modelo lógico. Findada esta passagem, foi

verificado se o modelo estava normalizado segunda as primeiras três formas normais, foi definido um tamanho inicial da base dados e por fim, a revisão do modelo com os utilizadores.

Por fim, foram construídos *mockups* que procuram idealizar a *interface* e respetivas camadas de apresentação dos dois sistemas.

A última fase do projeto corresponde à implementação das funcionalidades do sistema e aos testes que nos permitirão apurar a sua operacionalidade. Será também nesta fase que será produzida a versão final da documentação que acompanhará o *software* produzido de forma a que esta se encontre coerente com o projeto realizado. Todo o projeto será elaborado com recurso a ferramentas Microsoft.

## **Referências**

Sommerville, I. (2010). *Software Engineering*. *Software Engineering*.

## **Lista de Siglas e Acrónimos**

**BD** Base de Dados

**UML** *Unified Modeling Language*

**GPS** *Global Positioning System*

## **Anexos**

## **I. Anexo 1**