

**Universidade do Minho**

Escola de Engenharia

Mestrado Integrado em Engenharia Informática

**Unidade Curricular de**

**Laboratórios de Informática IV**

Ano Lectivo de 2016/2017

**Taste Advisor**

**Ana Rita Marques, A74218**

**Célia Figueiredo, A67637**

**Humberto Vaz, A73236**

**Ricardo Lopes, A72062**

Fevereiro, 2016

**LI41617**

|  |  |
| --- | --- |
| 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](#_Toc480295185)

[1.1. Contextualização 1](#_Toc480295186)

[1.2. Apresentação do Caso de Estudo 1](#_Toc480295187)

[1.3. Motivação e Objetivos 2](#_Toc480295188)

[1.4. Justificação do sistema 3](#_Toc480295189)

[1.5. Utilidade do Sistema 3](#_Toc480295190)

[1.6. Estabelecimento da identidade do projeto 3](#_Toc480295191)

[1.7. Identificação dos recursos necessários 3](#_Toc480295192)

[1.8. Maqueta do sistema 6](#_Toc480295193)

[1.9. Definição de um conjunto de medidas de sucesso 6](#_Toc480295194)

[1.10. Plano de desenvolvimento 7](#_Toc480295195)

[1.11. Organização do documento 9](#_Toc480295196)

[2. Análise de requisitos 10](#_Toc480295197)

[2.1. Levantamento de requisitos de Utilizador 10](#_Toc480295198)

[2.2. Levantamento de requisitos de Sistema 11](#_Toc480295199)

[2.2.1 Funcionais 11](#_Toc480295200)

[2.2.2 Não Funcionais 13](#_Toc480295201)

[3. Modelos do sistema 14](#_Toc480295202)

[3.1. Diagramas UML 14](#_Toc480295203)

[3.1.1 Diagramas de Use Case 14](#_Toc480295204)

[3.1.2 Diagramas de Atividade 14](#_Toc480295205)

[3.1.3 Modelo de Domínio 15](#_Toc480295206)

[3.1.4 Diagrama de Classes 15](#_Toc480295207)

[4. Bases de Dados 17](#_Toc480295208)

[4.1. Modelo Concetual 17](#_Toc480295209)

[4.1.1 Identificar os tipos de entidades 17](#_Toc480295210)

[4.1.2 Identificar tipos de relacionamento 17](#_Toc480295211)

[4.1.3 Identificar e associar atributos com os tipos de entidades e relacionamentos 18](#_Toc480295212)

[4.1.4 Determinar domínios dos atributos 20](#_Toc480295213)

[4.1.5 Determinar chaves primárias, candidatas e alternativas 20](#_Toc480295214)

[4.1.6 Desenho do diagrama ER 20](#_Toc480295215)

[4.1.7 Revisão do modelo de dados com o utilizador 20](#_Toc480295216)

[4.2. Modelo Lógico 21](#_Toc480295217)

[4.2.1 Validação do modelo lógico através de normalização 21](#_Toc480295218)

[4.2.2 Elaboração e validação do esquema lógico da base de dados 21](#_Toc480295219)

[4.2.3 Definição do tamanho inicial da base de dados e análise do seu crescimento futuro 21](#_Toc480295220)

[4.2.4 Revisão do modelo lógico final com os futuros utilizadores 21](#_Toc480295221)

[5. Mockups 23](#_Toc480295222)

[6. Conclusões e Trabalho Futuro 25](#_Toc480295223)

**Anexos**

[I. Anexo 1 30](#_Toc477430263)

# Índice de Figuras

[Figura 1 - Maqueta do sistema 4](#_Toc477190681)

[Figura 2 - Diagrama de Gantt 7](#_Toc477190682)

# Índice de Tabelas

[Tabela 1 - Tabela relativa às entidades 17](#_Toc480294902)

[Tabela 2 – Tabela de relacionamentos 18](#_Toc480294903)

[Tabela 3 - Associação 20](#_Toc480294904)

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 nosso software deverá ser 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.

* 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 especifico 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. 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. 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. 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. 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. 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. 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.

1. **Microsoft Visual Studio;**

É o ambiente de desenvolvimento das aplicações .NET.

1. **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.

1. **Microsoft Project**

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

1. **Visual Paradigm**

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

1. **Microsoft SQL Server;**

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

1. **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.

1. **Bing Speech API**

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

1. **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 o utilizador pode enriquecer a base de dados da aplicação deixando os seus comentários e/ou fotografias 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 encontrem dentro dos limites geográficos do distrito de Portugal.
   1. 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.

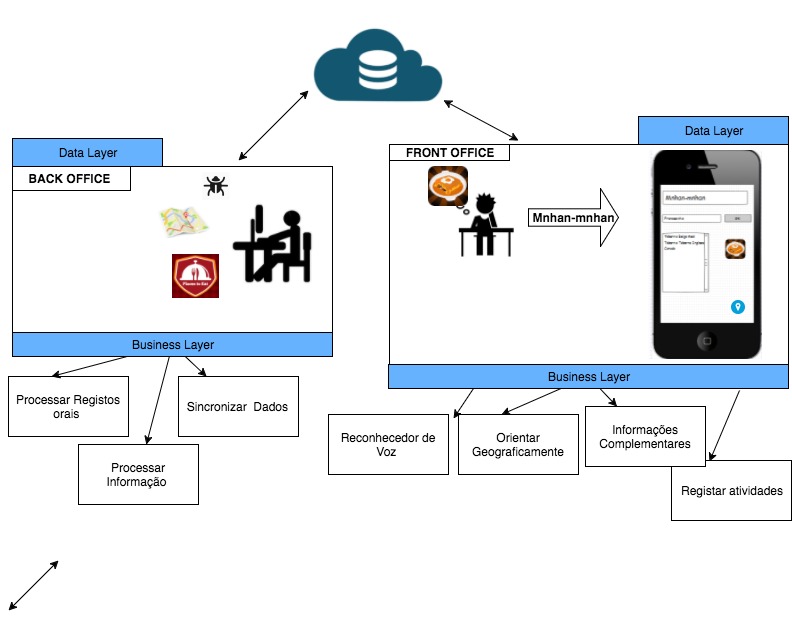


Figura 1 - Maqueta do sistema

* 1. 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. 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 perceção 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 aspetos 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.

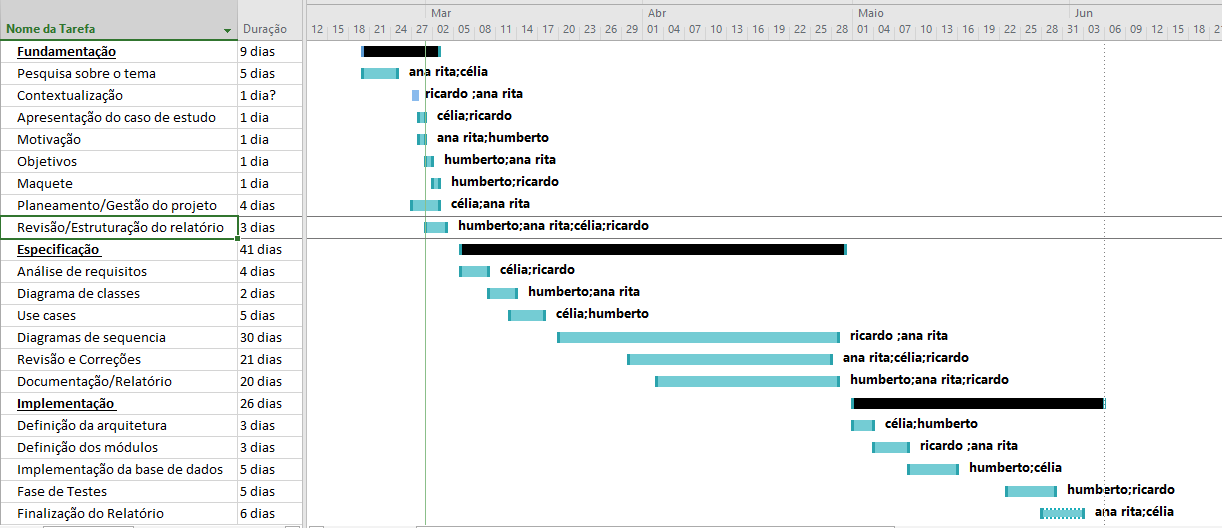


Figura 2 - Diagrama de Gantt

* 1. Organização do documento

Este documento está organizado em capítulos. Sendo que este primeiro relata sobre a fundamentação do projeto a realizar. Provavelmente irão existir mais dois capítulos um que relata a especificação e outro onde é descrita a implementação do projeto. O último capítulo é o das conclusões e trabalho futuro.

1. Análise de requisitos

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

* 1. Levantamento de requisitos de Utilizador

1. Poderá usar a aplicação com ou sem login efetuado.
2. Pesquisar na aplicação um prato de comida através de voz ou escrita.
3. O utilizador poderá visualizar o trajeto através da aplicação, que irá fornecer um mapa com a localização do restaurante.
4. O utilizador deve poder escolher qual a sua preferência entre os seguintes parâmetros: - Localização;

- Classificação de experiências de outros utilizadores;

- Preço;

1. O utilizador, com login efetuado, poderá adicionar/remover um restaurante aos seus favoritos.
2. O utilizador, com login efetuado, poderá visualizar a lista dos seus restaurantes favoritos.
3. O utilizador, com login efetuado, poderá adicionar/remover um restaurante à lista de locais que não gosta
4. O utilizador poderá visualizar a lista dos seus restaurantes marcados como “Não Gosto”.
5. O utilizador pode efetuar registo/login com a sua conta *Facebook* ou *Google*.
6. O utilizador, com login efetuado, poderá publicar a sua experiência na rede social *Facebook*.
7. No final da experiência, o utilizador poderá classificar a sua experiência de degustação no local escolhido bem como marcar o restaurante como “Gosto” ou “Não Gosto”.
8. O utilizador terá acesso ao histórico das suas avaliações
   1. Levantamento de requisitos de Sistema
      1. Funcionais
9. Para um utilizador ter acesso a algumas funcionalidades do sistema, como por exemplo, adicionar um restaurante aos seus favoritos, este tem que inserir as suas credenciais ou efetuar login com a sua conta do *Facebook*.
10. 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.
11. O requisito 2 aponta para a existência de 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*.
12. O *backOffice* deve permitir, aquando uma pesquisa, fornecer uma lista de locais ordenados decrescentemente de acordo com o parâmetro escolhido (localização, classificação e preço) que correspondem à pesquisa efetuada em *frontOffice*;
13. Para o local selecionado, o sistema irá apresentar uma lista de informações acerca do restaurante:
    1. **Nome do restaurante;**
    2. **A morada**
    3. **O tipo de comida**
    4. **O preço médio praticado**
    5. **Uma ou mais fotografias do local**
    6. **Classificação (Excelente, Muito Bom, Médio, Fraco, Terrível) e número de avaliações dos clientes**
14. A aplicação irá mostrar um mapa com o trajeto para o restaurante selecionado.
15. O sistema irá responder com uma lista de locais correspondentes à pesquisa do utilizador ordenados pelo parâmetro definido na mesma. Se o utilizador não selecionar nenhum parâmetro, o *default* será Localização (isto é, aquele que se encontra mais perto dele).
16. Cada item da lista apresentada terá: o nome do restaurante, uma imagem alusiva ao mesmo, a cidade, preço médio.
17. Haverá a possibilidade do utilizador poder adicionar e/ou remover um restaurante dos seus favoritos, para tal, haverá um botão específico.
18. Assim que o utilizador adiciona um restaurante aos seus favoritos, o *backOffice* irá colocar o mesmo na base de dados.
19. Logo que o utilizador pretenda ver a lista dos seus favoritos, o sistema em *backOffice* irá trazer para f*rontOffice* o resultado dos restaurantes adicionados anteriormente.
20. O sistema irá permitir que o utilizador marque um restaurante como “Não gosto”, tendo o *backOffice* a responsabilidade de guardar a informação na base de dados.
21. Quando o utilizador pretender visualizar a lista dos seus locais “Não Gosto”, o sistema em backOffice irá trazer para frontOffice o resultado da informação adicionada anteriormente.
22. O sistema guardará na base de dados os dados de autenticação do registo através do *Facebook*.
23. O sistema irá permitir que o utilizador faça uma publicação na sua rede social sobre a sua experiência de degustação, de forma simples e sem sair da aplicação.
24. No final da experiência, a aplicação permitirá que o utilizador faça uma avaliação da experiência realizada, assim como possibilitará que este marque o restaurante como “Gosto” ou “Não Gosto”.
25. Sempre que o utilizador, com login efetuado, realize uma avaliação à sua experiência gastronómica, o sistema irá guardar essa informação na base de dados;
26. Quando o utilizador pretender visualizar o seu histórico de avaliações, o sistema em *backOffice* irá trazer para *frontOffice* o resultado da informação adicionada previamente.
27. Toda a informação relativa aos restaurantes e aos utilizadores devem estar armazenadas numa base de dados relacional presente no *backOffice*;
28. Relativamente aos dados fornecidos pelo utilizador acerca da experiência (p.ex. avaliação) estes serão armazenados na base de dados do *backOffice*;
    * 1. Não Funcionais
29. O dispositivo móvel onde o *frontOffice* será instalado deve estar munido de microfone, acesso à Internete *GPS*.
30. Modelos do sistema
    1. Diagramas UML
       1. Diagramas de Use Case

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, sendo que por vezes é também representada a interação entre os dois subsistemas.

* + 1. 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.

Para ilustrar os conceitos e as relações entre estes que surgem no contexto do negócio das investigações dos detetives foi construído o Modelo de Domínio abaixo apresentado. Este modelo representa um esquema conceptual do problema em questão e nele podem ser vistas as entidades que têm participação no negócio da empresa, 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.

* + 1. 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.

* + 1. 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.

1. Bases de Dados
   1. Modelo Concetual
      1. Identificar os tipos de entidades

Após a leitura dos requisitos foram identificadas xx entidades:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome da entidade** | **Descrição** | **Sinónimos** | **Ocorrência** |
| Utilizador | Utilizadores do sistema. | *User*, Usuário, Membro. | Uma vez por cada utilizador do sistema. |
| Caso | Casos aceites pela empresa. | Situação, investigação, circunstância. | Sempre que o diretor da empresa aceita um caso, ele é inserido no sistema |
| Tarefa | Tarefas atribuídas aos Agentes. | Objetivo, atividade, função, missão. | Cada Inspetor-Chefe delega várias tarefas aos seus Agentes. |

Tabela 1 - Tabela relativa às entidades

* + 1. Identificar tipos de relacionamento

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entidade** | **Cardinalidade** | **Relacionamento** | **Entidade** | **Descrição** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Tabela 2 – Tabela de relacionamentos

* + 1. 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 |
| Utilizador | Código | Identifica inequivocamente um utilizador | Valor inteiro positivo | Não | - |
| Nome | Nome do utilizador | 5-30 caracteres variáveis | Não | - |
| *Password* | Palavra-chave do utilizador | 5-30 caracteres variáveis | Não | - |
| Tarefa | Código | Identifica inequivocamente uma tarefa | Valor inteiro positivo | Não | - |
| Título | Título da tarefa | 5-30 caracteres variáveis | Não | - |
| Descrição | Descrição da tarefa | Texto | Não | - |
| Objetivos | Objetivos a alcançar com a realização da tarefa | Texto | Não | - |
| Realizada | Indica se a tarefa já foi realizada | Boolean | Não | False |
| Exportada | Indica se a tarefa já foi exportada | Boolean | Não | False |
| Localização | Localização de uma tarefa | - | Não | - |
| Latitude | Latitude da localização da tarefa | Valor real | Não | - |
| Longitude | Longitude da localização da tarefa | Valor real | Não | - |
| Dado | Dado de uma tarefa | 5-30 caracteres variáveis | Não | - |
| Caminho | Caminho para o ficheiro | 5-30 caracteres varáveis | Não | - |
| Tipo | Tipo de dados (fotografias, registo de voz, etc.) | 5-30 caracteres variáveis | Não | - |
| Data | Data em que o dado foi recolhido | Data | Não | - |
| Localização | Localização de um dado | - | Não | - |
| Latitude | Latitude da localização do dado | Valor real | Não | - |
| Longitude | Longitude da localização do dado | Valor real | Não | - |
| Caso | Código | Identifica inequivocamente um caso | Valor inteiro positivo | Não | - |
| Nome | Nome do caso | 5-30 caracteres variáveis | Não | - |
| Descrição | Descrição do caso | Texto | Não | - |
| Objetivos | Objetivos a alcançar com a realização do caso | Texto | Não | - |
| Relatório | Relatório final do caso | 3-40 caracteres variáveis | Sim | NULL |
| Terminado | Indica se o caso está concluído | Boolean | Não | False |

Tabela 3 - Associação

* + 1. Determinar domínios dos atributos
    2. Determinar chaves primárias, candidatas e alternativas
    3. Desenho do diagrama ER
    4. Revisão do modelo de dados com o utilizador
  1. Modelo Lógico
     1. Validação do modelo lógico através de normalização
     2. Elaboração e validação do esquema lógico da base de dados
     3. Definição do tamanho inicial da base de dados e análise do seu crescimento futuro
     4. Revisão do modelo lógico final com os futuros utilizadores

1. Mockups

Quando um Utilizador inicia a aplicação “Taste Advisor”, a primeira interface que lhe é apresentada é a página de pesquisa.

1. 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 consiste 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.

No que toca ao levantamento de requisitos, procedeu-se à devida separação dos mesmos em requisitos de utilizador e de sistema, sendo estes últimos divididos ainda em requisitos funcionais e não funcionais. Esta organização permitiu 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 aspetos 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 perceção 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 concetual. Com o modelo concetual 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.

Referências

Sommerville, I. (2010). *Software Engineering*. *Software Engineering*. https://doi.org/10.1111/j.1365-2362.2005.01463.x

Lista de Siglas e Acrónimos

**BD** Base de Dados

**UML** *Unified Modeling Language*

**GPS** *Global Positioning System*

# Anexos

1. Anexo 1