

**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.

# Índice

[1. Introdução 1](#_Toc477430275)

[1.1. Contextualização 1](#_Toc477430276)

[1.2. Apresentação do Caso de Estudo 1](#_Toc477430277)

[1.3. Motivação e Objetivos 2](#_Toc477430278)

[1.4. Justificação do sistema 3](#_Toc477430279)

[1.5. Utilidade do Sistema 3](#_Toc477430280)

[1.6. Estabelecimento da identidade do projeto 3](#_Toc477430281)

[1.7. Identificação dos recursos necessários 3](#_Toc477430282)

[1.8. Maqueta do sistema 4](#_Toc477430283)

[1.9. Definição de um conjunto de medidas de sucesso 4](#_Toc477430284)

[1.10. Plano de desenvolvimento 5](#_Toc477430285)

[1.11. Organização do documento 7](#_Toc477430286)

[2. Análise de requisitos 8](#_Toc477430287)

[2.1. Levantamento de requisitos de Utilizador 8](#_Toc477430288)

[2.2. Levantamento de requisitos de Sistema 8](#_Toc477430289)

[2.2.1 Funcionais 8](#_Toc477430290)

[2.2.2 Não Funcionais 8](#_Toc477430291)

[3. Modelos do sistema 9](#_Toc477430292)

[3.1. Diagramas UML 9](#_Toc477430293)

[3.1.1 Modelo de Domínio 9](#_Toc477430294)

[3.1.2 Diagramas de Use Case 9](#_Toc477430295)

[4. Bases de Dados 10](#_Toc477430296)

[4.1. Modelo Concetual 10](#_Toc477430297)

[4.1.1 Identificar os tipos de entidades 10](#_Toc477430298)

[4.2. Modelo Lógico 10](#_Toc477430299)

[5. Mockups 11](#_Toc477430300)

[6. Mockups 12](#_Toc477430301)

[7. Conclusões e Trabalho Futuro 13](#_Toc477430302)

**Anexos**

[I. Anexo 1 17](#_Toc477430263)

# Índice de Figuras

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

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

# Índice de Tabelas

**Não foi encontrada nenhuma entrada do índice de ilustrações.**

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:

* Permitir que o utilizador dite o que quer comer no momento, quer por escrito, quer por voz;
* 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;
* 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;
* 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;
* Apresentar uma lista de tendências de degustação na zona onde o utilizador se encontra;
* 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:

* **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.

* **Microsoft Visual Studio;**

É o ambiente de desenvolvimento das aplicações .NET.

* **Microsoft .NET C#;**

C# foi desenvolvida pela [Microsoft](https://pt.wikipedia.org/wiki/Microsoft) como parte da [plataforma .NET](https://pt.wikipedia.org/wiki/Microsoft_.NET), é uma linguagem fortemente 'tipada', e, possui paradigmas de programação [imperativa](https://pt.wikipedia.org/wiki/Programa%C3%A7%C3%A3o_imperativa), [funcional](https://pt.wikipedia.org/wiki/Programa%C3%A7%C3%A3o_funcional), [declarativa](https://pt.wikipedia.org/wiki/Programa%C3%A7%C3%A3o_declarativa), orientada aos objetos, será a linguagem a utilizar para o desenvolvimento da aplicação.

* **Microsoft Project**

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

* **Visual Paradigm**

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

* **Microsoft SQL Server;**

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

* **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.

* **Bing Speech API**

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

* Dados externos

Como Buscar os nomes dos restaurantes e ementas ?????? temos de ir falar com o Belo

* 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.

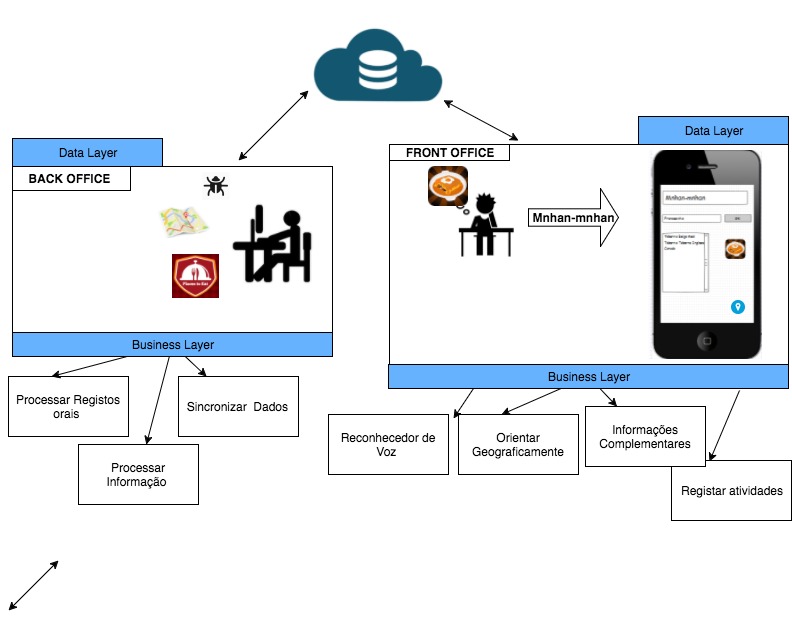


Figura 1 - Maqueta do sistema

* 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.

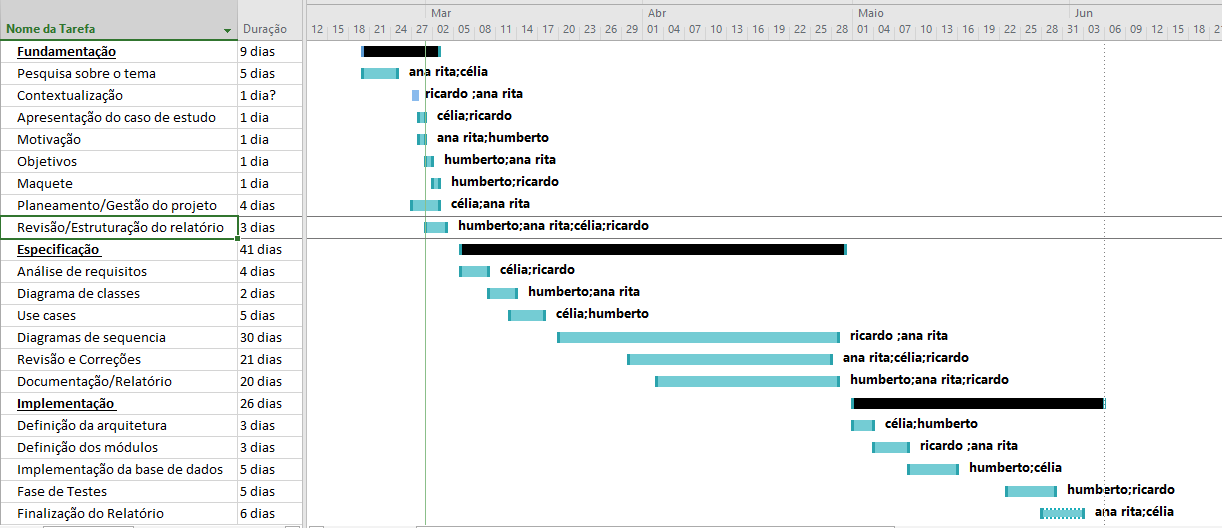


Figura 2 - Diagrama de Gantt

* 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 o restaurante aos seus favoritos
2. O utilizador pode efetuar registo/login com facebook
3. O utilizador com login efetuado, poderá publicar a sua experiência na rede social facebook
4. No final da experiência, o utilizador poderá classificar a sua experiência de degustação no local escolhido.
   1. Levantamento de requisitos de Sistema
      1. Funcionais
5. Ser-lhe-á apresentado aquando a pesquisa, as informações de contacto, horário de funcionamento, classificação do custo médio, rating
   1. 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*;*
   2. 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.
   3. 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.
   4. 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;
   5. Para o local selecionado, o sistema irá apresentar uma lista de informações acerca do restaurante
   * O **Nome do restaurante**;
   * A **morada**
   * O **tipo de comida**
   * O **preço médio praticado**
   * Uma ou mais **fotografias do local**
   * **Classificação** (Excelente,Muito Bom, Médio, Fraco, Terrível) **e número de avaliações dos clientes**
   1. Segundo o requisito 4, um Agente possui um plano de atividades que constitui um planeamento das tarefas (detalhadas no requisito 3) a executar durante uma sessão de trabalho;
   2. Para esse efeito, a criação de um plano de atividades consiste numa associação de tarefas a um Agente que, ao serem importadas pelo *frontOffice,* serão agregadas num conjunto de maneira a formar a sessão de trabalho pretendida;
   3. Pode ser constatado através do requisito 5 que os casos aceites (e consequentemente inseridos no sistema) pelo Diretor para investigação devem ser entregues a um Inspetor-Chefe;
   4. De forma a concretizar esta associação, o *backOffice* deve permitir atribuir a responsabilidade de um dado caso (mencionado no requisito 2) ao perfil de um Inspetor-Chefe.
   5. O sistema do *backOffice* deve possibilitar ao Diretor a criação de perfis de acesso correspondentes às permisssões de um Inspetor-Chefe;

* 1. Deve ser permitido ao Diretor que, no *backOffice*, possa criar perfis para os seus Agentes acederem ao sistema. Para além disso, deve poder associá-los a uma determinada equipa chefiada por um Inspetor-Chefe (mencionado no requisito 8);
  2. Esta associação vai fazer com que os Agentes em causa integrem apenas a equipa desse Inspetor-Chefe e, como tal, esta ligação deve ser única pois um Agente não deverá fazer parte de mais do que uma equipa de investigação simultaneamente (explicado no requisito 13).
  3. O *backOffice* deve permitir ao Inspetor-Chefe ver os dados (código de identificação, nome próprio e apelido) de todos os Agentes que coordena;
  4. Tal como mencionado no ponto 5, os Inspetores-Chefe são responsáveis por um conjunto de casos. Assim sendo deve ser permitido ao Inspetor-Chefe poder visualizar todos os seus casos bem como o estado em que se encontram;
  5. Um Inspetor-Chefe, com base nos casos que coordena (ponto 9.1) cria e atribui tarefas aos Agentes que coordena. Estas tarefas constituem um plano de atividades, como mencionado no ponto 4.
  6. Para além da atribuição de tarefas aos seus Agentes, um Inspetor-Chefe, sempre que necessário, deve poder verificar as tarefas que estão atribuídas a um dado Agente, bem como o seu estado de realização (terminadas / em curso);
  7. Caso um Inspetor-Chefe necessite remover uma tarefa que ainda não foi sincronizada pelo Agente, deve ser possível fazê-lo;
  8. Um Inspetor-Chefe só pode visualizar e gerir os casos a ele entregues;
  9. Uma tarefa não pode ser removida se já tiver sido sincronizada pelo Agente ao qual foi atribuída;
  10. Sempre que um caso é resolvido, o Inspetor-Chefe responsável deve marcá-lo como concluído;
  11. Um caso não pode ser marcado como concluído se ainda existirem tarefas em curso relacionadas com o caso;
  12. Além das funcionalidades específicas de um Inspetor-Chefe, este também pode atuar como um Agente, ou seja, tem acesso a todas as funcionalidades de um Agente;
  13. Como um Inspetor-Chefe é também um Agente (ponto 11.1), este tem a possibilidade de atribuir um plano de atividade a ele próprio;
  14. O *backOffice* deve dar a possibilidade a um Inspetor-Chefe de recolher e analisar toda a informação sobre um dado caso e criar um *PDF* com essa informação. Este relatório são as considerações finais, mencionadas no ponto 2;
  15. Após a geração do *PDF* sobre um certo caso (ponto 12.1), o Inspetor-Chefe deve ter a possibilidade de o poder enviar por email a qualquer endereço eletrónico;
  16. Agente deve poder importar o plano de atividade, a ele definido, para o seu *frontOffice*;
  17. Um Agente apenas recebe planos de Atividade do Inspetor-Chefe que o coordena; Agente deve poder consultar, no seu *frontOffice*, o seu plano de atividades a fim de o poder executar;
  18. Um Agente apenas pode visualizar os casos que estejam ao encargo do seu Inspetor-Chefe;
  19. Um Agente deve poder realizar e cumprir cada tarefa do plano de atividades;
  20. Existe a possibilidade do Agente suspender uma tarefa em execução.
  21. Sempre que um Agente seleciona uma tarefa, deve existir a possibilidade do *frontOffice* apresentar o percurso, indicando o caminho a percorrer para alcançar o último local assinalado (caso este tenha sido indicado sob a forma de dados recolhidos, dados estes mencionados no ponto 17.1). Caso não exista nenhum registo, é utilizado o local de investigação da tarefa;
  22. Quando um Agente está a realizar uma tarefa, o *frontOffice* deve permitir tirar fotografias, notas escritas e de voz. Todas estas ações devem ser registadas e guardadas (na base de dados do *frontOffice)* juntamente com a localização do sítio onde foram realizadas. Além disso, estas informações ficam associadas à tarefa em questão;
  23. Além das funcionalidades mencionadas no ponto 17.1, um Agente deve poder obter informações complementares (através do *frontOffice)* sobre um determinado assunto. Essa obtenção será realizada com o auxílio de um motor de busca;
  24. Quando o Agente se conecta ao *backOffice*, todas as tarefas que realizou durante a sua sessão de trabalho, bem como os dados recolhidos (mencionados no requisito 17) relativos a cada uma das tarefas que concretizou, devem ser exportados para o sistema central;
  25. Apenas as tarefas que estão dadas como concluídas no *frontOffice* do Agente devem ser sincronizadas (exportadas) com o *backOffice*;
  26. As tarefas que esperam a sua concretização devem permanecer no *frontOffice*.
  27. O *backOffice* deve analisar as gravações de voz registadas pelos Agentes, traduzindo-as para um documento anotado;
  28. Esta análise é feita com base em *tags* que têm o objetivo de indicar que informação se procede à sua ocorrência;
  29. Quando a análise do registo de voz deteta uma *tag*, dependendo do significado que lhe é atribuído, a informação correspondente é processada de forma adequada.
  30. Toda a informação relativa aos casos e às entidades da empresa devem estar armazenadas numa base de dados relacional presente no *backOffice*;
  31. Relativamente aos dados das tarefas recolhidos durante o trabalho de campo dos Agentes, estes encontram-se armazenados na base de dados do dispositivo móvel enquanto não são exportados para o *backOffice*;
  32. Depois dos dados recolhidos nas investigações serem exportados para o *backOffice*, estes deixam de estar presentes no *frontOffice*.
      1. Não Funcionais

1. Modelos do sistema
   1. Diagramas UML
      1. Modelo de Domínio
      2. Diagramas de Use Case
2. Bases de Dados
   1. Modelo Concetual
      1. Identificar os tipos de entidades
   2. Modelo Lógico
3. Mockups
4. Mockups
5. Conclusões e Trabalho Futuro

Terminada esta 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.

A próxima fase do projeto consistirá na modelação do sistema. Nesta fase será realizada a análise de requisitos que permitirá avançar para a construção dos diagramas em UML que irão constituir a idealização do sistema.

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