



Universidade do Minho
Escola de Engenharia
Mestrado Integrado em Engenharia Informática

Unidade Curricular de Laboratórios de Informática IV

Ano Letivo de 2018/2019

FeellItaly

Diogo Braga - A82547

João Silva - A82005

Ricardo Caçador - A81064

Ricardo Ferreira - A82568

Ricardo Veloso - A81919

Fevereiro, 2019

L | 4

Data de	
Recepção	
Responsável	
Avaliação	
Observações	

FeeItaly

Diogo Braga - A82547

João Silva - A82005

Ricardo Caçador - A81064

Ricardo Ferreira - A82568

Ricardo Veloso - A81919

Fevereiro, 2019

Resumo

Este relatório irá apresentar todo o processo efetuado para a realização de um software correspondente a um assistente de cozinha, denominado *FeelItaly*.

Inicialmente apresentamos a fundamentação, onde toda a informação relativa à contextualização do projeto bem como as características mais gerais são devidamente abordadas.

De seguida, iremos realizar a fase de especificação e construção que abordam aspectos mais técnicos da implementação do software.

Por último, concluiremos o projeto com uma apreciação crítica dos diferentes itens abordados e uma previsão para o futuro do nosso software.

Área de Aplicação: Implementação de Software; Modelação de Software; Sistemas de Bases de Dados; Gestão de Projetos.

Palavras-Chave: C#; .NET; SQL Server; Visual Studio; UML; Gantt.

Índice

Resumo	i
Índice de Figuras.....	v
Índice de Tabelas.....	vii
1. Introdução	1
<i>1.1. Contextualização.....</i>	<i>1</i>
<i>1.2. Motivação e Objetivos</i>	<i>2</i>
1.2.1 Motivação	2
1.2.2 Objetivos	2
<i>1.3. Justificação, Validação e Utilidade do sistema</i>	<i>3</i>
<i>1.4. Estabelecimento da identidade do projeto</i>	<i>4</i>
<i>1.5. Identificação dos recursos necessários</i>	<i>4</i>
<i>1.6. Maquete do sistema</i>	<i>5</i>
<i>1.7. Definição de medidas de sucesso</i>	<i>6</i>
<i>1.8. Plano de desenvolvimento</i>	<i>7</i>
<i>1.9. Descrição da estrutura do relatório</i>	<i>8</i>
2. Requisitos.....	10
<i>2.1. Requisitos do Utilizador e de Sistema Funcionais</i>	<i>10</i>
2.1.1 Registo no sistema	10
2.1.2 Autenticação no sistema.....	10
2.1.3 Estabelecimento de uma configuração inicial de pratos e ingredientes.....	11
2.1.4 Visualização das estatísticas.....	11
2.1.5 Listagem de receitas	11
2.1.6 Filtragem de receitas.....	11
2.1.7 Comunicação via voz.....	12
2.1.8 Seleção de uma receita.....	12
2.1.9 Preparar uma receita	13
2.1.10 Avaliação de uma receita	13

2.1.11 Comentar uma receita	13
2.1.12 Adição de uma receita	14
2.1.13 Agendamento de uma ementa semanal.....	14
2.1.14 Gestão da despensa	14
2.1.15 Pesquisa de locais para comprar ingredientes em falta.....	15
2.1.16 Listagem de locais para comprar ingredientes em falta	15
2.2. Requisitos de Sistema Não Funcionais	16
2.2.1 Requisitos de Produto.....	16
2.2.2 Requisitos Organizacionais.....	16
2.2.3 Requisitos Externos.....	16
3. Modelação UML	17
3.1. Modelos de Contexto	17
3.1.1 Modelo de Domínio	18
3.1.2 Diagrama de Atividade.....	19
3.2. Modelos de Interação	20
3.2.1 Diagrama de Use Cases	21
3.2.2 Especificação Textual de Use Cases	22
3.2.3 Diagramas de Sequência de Sistemas	25
3.2.4 Diagramas de Sequência de Implementação.....	28
3.3. Modelos Estruturais	31
3.3.1 Diagrama de Classe	31
3.3.2 Diagrama de Classe ORM	32
3.4. Modelos Comportamentais.....	33
3.4.1 FeellItaly.....	33
3.4.2 Preparar Receita.....	34
3.4.3 Adicionar Receita	34
3.4.4 Registo	35
4. Base de Dados	36
4.1. Modelo Conceptual	36
4.1.1 Identificação e caracterização das entidades	36
4.1.2 Identificação e caracterização dos relacionamentos	37
4.1.3 Domínio dos atributos.....	37
4.1.4 Apresentação do modelo conceptual	40
4.2. Modelo Lógico.....	40

4.2.1 Construção e validação do modelo de dados lógico	41
4.2.2 Desenho do modelo lógico	43
4.2.3 Normalização e Integridade	43
4.2.4 Verificação da Integridade das Restrições	43
4.2.5 Dados Necessários	44
4.2.6 Domínio das Restrições dos Atributos	44
4.2.7 Cardinalidade	44
4.2.8 Integridade das Entidades.....	45
4.2.9 Integridade Referencial.....	45
4.2.10 Restrições gerais	46
4.2.11 Revisão do modelo Lógico com o Utilizador	46
5. Prototipagem da Interface	47
6. Conclusão e Trabalho Futuro	52
7. Bibliografia	53

Índice de Figuras

Figura 1 - Maquete do sistema	5
Figura 2 - Diagrama de Gantt: Previsão da Especificação	7
Figura 3 - Diagrama de Gantt: Previsão da Fundamentação	7
Figura 4 - Diagrama de Gantt: Previsão da Construção	7
Figura 5 - Diagrama de Gantt: Realidade da Fundamentação	8
Figura 6 - Modelo de Domínio	18
Figura 7 - Diagrama de Atividade de Seleção duma Receita	19
Figura 8 - Diagrama de Atividade de Preparação duma Receita	20
Figura 9 - Diagrama de Use Cases	21
Figura 10 - Use Case Pré-Visualizar Receitas	22
Figura 11 - Use Case Filtrar Receitas	22
Figura 12 - Use Case Selecionar Receita	23
Figura 13 - Use Case Preparar Receita	23
Figura 14 - Use Case Realizar Passo	24
Figura 15 - DSS Pré-Visualizar Receitas	25
Figura 16 - DSS Filtrar Receitas	25
Figura 17 - DSS Selecionar Receita	26
Figura 18 - DSS Preparar Receita	26
Figura 19 - DSS Realizar Passo	27
Figura 21 - DS Implementação Pré-Visualizar Receitas	28
Figura 20 - DS Implementação Filtrar Receitas	28
Figura 22 - DS Implementação Selecionar Receita	29
Figura 23 - DS Implementação Preparar Receita	29
Figura 24 - DS Implementação Realizar Passo	30
Figura 25 - Diagrama de Classe	31
Figura 26 - Diagrama Geral do Sistema com as respetivas submáquinas	33
Figura 27 - Sub-Máquina de Estado 'Preparar Receita'	34
Figura 28 - Sub-Máquina de Estado 'Adicionar Receita'	34
Figura 29 - Sub-Máquina de Estado 'Registo'	35
Figura 30 – Mockup Página Inicial	47

Figura 31 - Mockup Registrar	48
Figura 32 - Mockup LogIn	48
Figura 33 - Mockup Listar Receitas	49
Figura 34 - Mockup Selecionar Receita	50
Figura 35 - Mockup Preparar Receita	51

Índice de Tabelas

Tabela 1 - Identidade do Projeto	4
Tabela 2 - Domínio dos atributos das entidades	39
Tabela 3 - Domínio dos atributos dos relacionamentos	40
Tabela 4 - Relações do modelo Lógico	42

1. Introdução

Nos seguintes subcapítulos, apresenta-se a Fundamentação do projeto em causa.

1.1. Contextualização

No nosso quotidiano, cozinhar é uma atividade muito presente na vida de quase todos os cidadãos. Esta atividade é tão importante que chega a ser fulcral para a sobrevivência humana. É, portanto, muito importante saber a arte de cozinhar. Muitas são as pessoas (ex.: estudantes universitários) com pouca experiência na cozinha que utilizam as receitas como meio de aprendizagem autónoma. No caso destes, comida italiana é um tipo de comida particularmente mais rápida e de fácil confeção.

Para todas as pessoas que cozinham e tentam melhorar as suas capacidades, existem certas incomodidades que podiam ser resolvidas: estar com as mãos sujas de mexer em certos alimentos e ter que mudar a página dum livro de receitas, ter o espaço próprio para cozinhar e necessitar de sair dessa área para ler a receita, ter um livro com um número limitado de receitas, má gestão dos produtos alimentares em casa tendo em conta as necessidades semanais, etc.

Perante estas dificuldades presentes na vida de quem cozinha, surgiu o projeto *FeelItaly*.

Este projeto tem por base a criação de um software com a função de auxiliar qualquer pessoa nos seus cozinhados domésticos, neste caso, cozinhados italianos. Esta ideia partiu de um estudante de Erasmus proveniente da zona de Nápoles, cujo negócio familiar passa pela exportação de produtos de origem italiana. Com a sua vinda para Portugal estudar, chegou à conclusão que a gastronomia do seu país não se encontrava presente no quotidiano dos portugueses. Para mudar este paradigma, decidiu colocar os produtos da família em Portugal. Com o objetivo de causar um maior vínculo com o cliente e de modo a transmitir melhor o conceito italiano, este contactou um grupo de amigos do curso de Engenharia Informática da Universidade do Minho. Consequentemente o grupo concluiu que a criação do software apresentado a seguir seria a ferramenta mais apropriada para colocar esses produtos no dia-a-dia dos portugueses através da sua utilização na confeção de receitas tipicamente italianas.

Além de resolver inúmeros problemas envolvidos com a atividade de cozinhar, este assistente pessoal facilita outros pontos. Possui uma aglomeração de receitas de diferentes livros, que consequentemente aumenta em grande escala a variedade de escolha. Possui uma

secção de comentários de avaliações que proporciona interação entre utilizadores e consequentemente partilha de ideias. Por último, o facto de haver uma interação entre os utilizadores torna a aprendizagem e a cozinha mais fácil e saborosa.

1.2. Motivação e Objetivos

1.2.1 Motivação

O principal motivo que nos leva à realização deste projeto consiste no facto de existirem imensas dificuldades na hora de cozinhar. Este problema é partilhado por várias pessoas, seja porque não se lembram da receita do prato que querem cozinhar, seja porque necessitam de fazer um prato menos calórico ou mesmo porque não sabem cozinhar.

Durante o estudo do caso aqui apresentado, deparamo-nos com o facto de que muitas das pessoas que cozinham frequentemente, seguem receitas que estão em diferentes sites na internet, tutoriais de YouTube, livros de receitas, e ainda anotações que retiram de experiências já vividas. Deste modo conseguimos deduzir aqui alguns inconvenientes que podem ser eliminados, e uma das nossas motivações passa pelo utilizador conseguir aceder de forma fácil e eficaz a um conjunto de receitas, neste caso da cozinha italiana.

Como a arte de cozinhar bem não está ao alcance de todos, ou mesmo porque as fontes das receitas podem não ser confiáveis, é de grande importância que qualquer pessoa possa ser corretamente guiada no processo de confeção dum determinado prato.

Analizando o mercado nesta área deparamo-nos com produtos altamente qualificados, que envolvem tanto Software como Hardware e a sua obtenção pode tornar-se bastante dispendiosa. Ainda de referir que a maior parte dos produtos que providenciam este tipo de serviço não se encontram facilmente em Portugal, e mesmo encontrando, estes normalmente, não possuem o idioma português. É facilmente perceptível que a tarefa de assistência na hora de cozinhar e na gestão inteligente da despensa de cada cliente, pode assim tornar-se mais complicada para utilizadores que não compreendam tão bem inglês, ou não compreendam inglês de todo. Uma das nossas motivações é então trazer este mercado para terras lusas, e providenciar um serviço prestável e confiável a cada utilizador deste produto, levando a cada um deles o melhor que a cozinha italiana pode providenciar.

1.2.2 Objetivos

O principal objetivo deste produto informático será providenciar ao utilizador um conjunto alargado de receitas de comida italiana, bem como o seu modo de preparação. Temos em vista dois modos de apresentação para cada receita, sendo a primeira a mais clássica, a receita em modo textual. Este modo de apresentação visa chegar às pessoas que

queiram usar esta solução informática, mas não se deem bem com as novas tecnologias, querendo apenas disfrutar das receitas que estão disponíveis.

O segundo modo de apresentação, e o mais desafiante porventura, consiste em acompanhar o utilizador em cada passo duma dada receita. Este acompanhamento deverá ser feito por voz, que é o mais prático possível aquando do ato de cozinhar.

Para melhorar a experiência de cada utilizador será ainda possível deixar comentários a uma dada receita e sugerir diferentes formas de confeção da mesma. Esta medida necessita de estar presente para que uma dada receita atinja mais e diferentes utilizadores, que à *priori* não pudesse ingerir certos ingredientes e logo ignorariam esta.

Outro dos objetivos considerado e deveras importante, visa chegar a um grupo de pessoas porventura mais restrito, que são aquelas que não tem tempo ou paciência para organizar os produtos alimentares que têm armazenados em casa. Pretendemos então, que a nossa solução agende uma ementa semanal para o utilizador a pedido do mesmo, e desta forma conseguiremos gerir os produtos que este necessita durante a semana.

1.3. Justificação, Validação e Utilidade do sistema

Após a realização de uma análise e recolha de informação do mercado das aplicações correspondentes a assistentes pessoais de cozinha, o grupo deparou-se com algumas lacunas que limitam a atividade do utilizador.

Em algumas das aplicações disponíveis não existem funcionalidades imprescindíveis como, por exemplo, realizar um planeamento semanal e criar uma lista de compras para esse plano. Noutros casos existe uma grande falta de interação com o utilizador durante o processo de cozinhar e, ainda associado à criação de uma lista de compras, não são referidos locais de compra (supermercados, minimercados e restantes) perto do local onde o utilizador se encontra.

Resumindo, existe uma falha enorme nas aplicações em geral em ajudar o utilizador nos múltiplos processos que constituem a arte de cozinhar. Por isso, com base nestas lacunas encontradas, a nossa ideia foi criar uma nova aplicação que, resolvendo todas estas situações, ofereça um sistema interativo e com todas as potencialidades para facilitar a vida do utilizador quando se trata de confeccionar pratos. Com estes problemas encontrados e com o objetivo de fazer face aos objetivos pretendidos, foi agendada uma reunião com o Georgio e com o seu filho Riccardi para verificar se todos os requisitos eram resolvidos.

Importante referir que esta aplicação vai funcionar sobre o método *Freemium*, isto é, vai funcionar em modo *Free* para alguns utilizadores e em modo *Premium* para outros. Desta forma o grupo perspetiva ter lucros. Os utilizadores *Free* terão acesso a funcionalidades como: acesso a 2/3 das receitas de cada secção (Entradas, Prato, Sobremesa), interação com o *FeelItaly* e configuração inicial de cada utilizador (Preferências de pratos do utilizador). Os utilizadores *Premium* terão acesso a estas funcionalidades e também: totalidade das receitas,

planeamento da ementa semanal e indicação de onde encontrar e adquirir os produtos em falta.

1.4. Estabelecimento da identidade do projeto

O projeto a desenvolver pela equipa, denominado *FeellItaly*, é um assistente pessoal de culinária direcionado para a comida Italiana. Esta funcionalidade disponível em web browser e uma versão app vai permitir ao utilizador obter ajuda na hora de criar um típico prato italiano. Utilizando apenas a própria voz, o utilizador conseguirá identificar o estilo de prato e quais os ingredientes que pretende utilizar, usufruindo assim de uma lista variada de receitas. Cada receita terá disponível o seu modo de confeção, os seus ingredientes e a sua informação nutricional. Com isto, o utilizador vai conseguir também criar um plano semanal dos pratos que pretende desenvolver e quais os ingredientes necessários para o efeito.

Resumem-se de seguida algumas características identificativas do sistema a desenvolver, com um objetivo meramente elucidativo, uma vez que as mesmas serão desenvolvidas à frente no relatório.

Nome	FeellItaly
Categoria	Alimentação
Slogan	Com FeellItaly na mão, é mais simples a confeção
Características	<ul style="list-style-type: none">• Freemium• Escolha de uma variada lista de pratos• Avaliações de pratos• Planeamento semana de receitas• Acompanhamento por voz e/ou textual nas várias tarefas• Navegação GPS
Faixa Etária	16+

Tabela 1 - Identidade do Projeto

1.5. Identificação dos recursos necessários

Para este projeto estima-se ser necessária a alocação de cinco alunos do curso de Engenharia Informática da Universidade Do Minho aos quais vão ser atribuídos os papéis de engenheiro de software (um aluno) e de programadores (quatro alunos). Para além de ser responsável pela especificação do projeto, o engenheiro de software irá assumir também o papel extraordinário de gestor de projeto.

Na primeira fase, serão realizadas duas reuniões com o Georgio e com o seu filho Riccardi para um levantamento de requisitos e, no fim do período de planeamento do projeto, para a confirmação do modelo elaborado. Será ainda necessário fazer um estudo de mercado

relativo ao melhor método para adquirir uma maior receita mensal com o software, que será encomendado a uma empresa externa.

Para cumprir estes requisitos, considera-se necessária a utilização do seguinte software:

- Microsoft Visual Studio
- Microsoft SQL Server
- Microsoft Visual C#
- Bing Maps Platform
- Visual Paradigm for UML
- Microsoft Speech Platform
- Microsoft avatar on Windows 10
- Microsoft Cortana

Será também necessário alocar um servidor responsável pelo alojamento dos serviços 24/7 assim como se prevê o possível uso dos smartphones e computadores da equipa de desenvolvimento para realizar testes nas diferentes plataformas existentes no mercado.

Os diferentes softwares e suas licenças estarão, à partida, garantidos pela equipa de desenvolvimento pelo que não acarretam nenhum encargo financeiro para o grupo.

Por fim, o projeto será desenvolvido nas instalações da Universidade do Minho.

1.6. Maquete do sistema

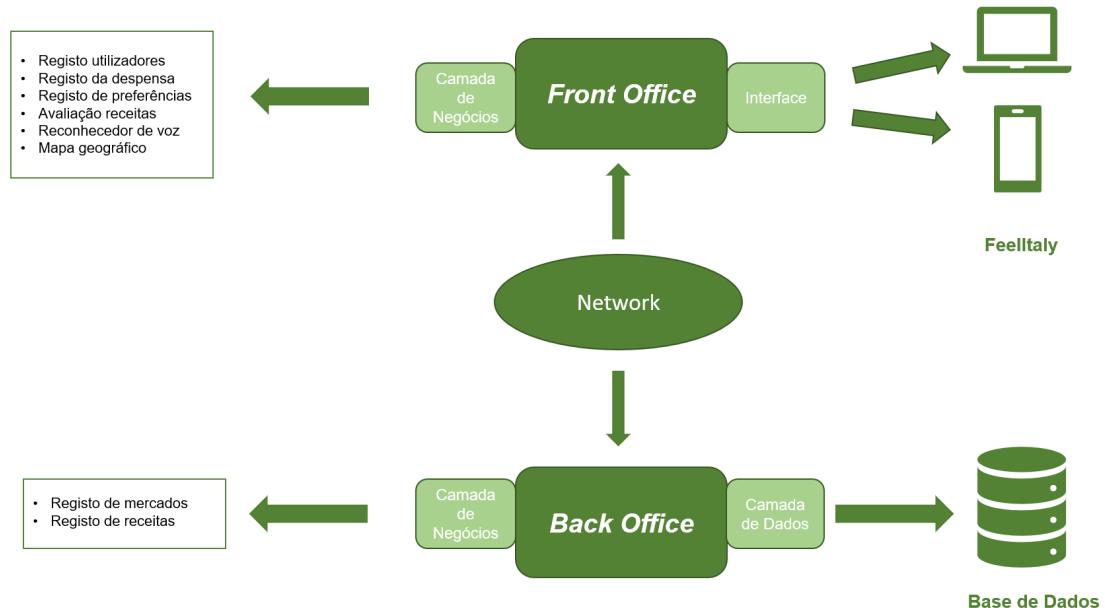


Figura 1 - Maquete do sistema

O projeto divide-se em três camadas, que são respetivamente, a camada de negócio, a camada de apresentação (Interface) e a camada e dados. A implementação será projetada

para que estas três camadas estejam devidamente separadas e se consiga trabalhar independentemente em cada uma, respeitando.

No *Front Office* do produto residirá a camada de apresentação. Esta camada será responsável pelas várias interações com o utilizador, seja via voz, via touch ou ainda, via os clássicos cliques. Neste lado do produto ainda reside uma pequena parte da camada de negócio responsável por materializar os pedidos dos utilizadores em comunicações com o servidor.

Por outro lado, no *Back Office* residirá uma camada de negócio bem mais complexa, capaz de tratar das comunicações entre servidor e utilizador, e capaz de organizar e gerir da melhor forma os dados trocados entre estes. É também neste lado do produto que residirá toda a camada de dados responsável por conter os diferentes dados relativos ao projeto, tais como as receitas, utilizadores e restante informação relevante.

1.7. Definição de medidas de sucesso

Na fundamentação do projeto, é vital construir um conjunto de medidas que mais à frente no projeto nos indiquem se o mesmo foi implementado com o êxito que pretendíamos.

Nesta secção apresentamos uma lista com os tópicos que consideramos fundamentais para o sucesso do nosso projeto.

- Todas as fases e etapas do projeto devem cumprir os prazos estabelecidos. Com este indicativo sabemos se o projeto está a avançar e permite realizar uma melhor gestão do tempo dedicado a cada etapa e ao projeto por cada elemento do grupo.
- Todos os requisitos identificados no levantamento devem estar presentes, aquando da terminação do projeto, na aplicação desenvolvida. Caso isto aconteça, podemos considerar que os objetivos foram realizados com sucesso.
- Para confirmar a escalabilidade do nosso sistema, consideramos que a nossa aplicação teria pelo menos 100 utilizadores e 25 receitas no primeiro mês de utilização.
- Para verificarmos que todas as funcionalidades da aplicação funcionam corretamente quando usadas pelos utilizadores, o sistema terá no mínimo 25 comentários e 25 avaliações durante o primeiro mês de utilização.

Durante os meses seguintes do sistema é imprescindível que os números sejam sempre superiores ao mês anterior.

1.8. Plano de desenvolvimento

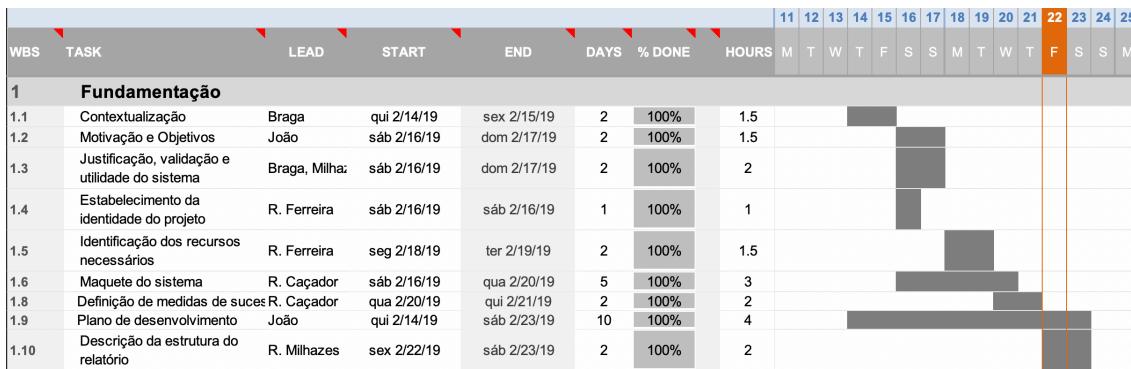


Figura 3 - Diagrama de Gantt: Previsão da Fundamentação

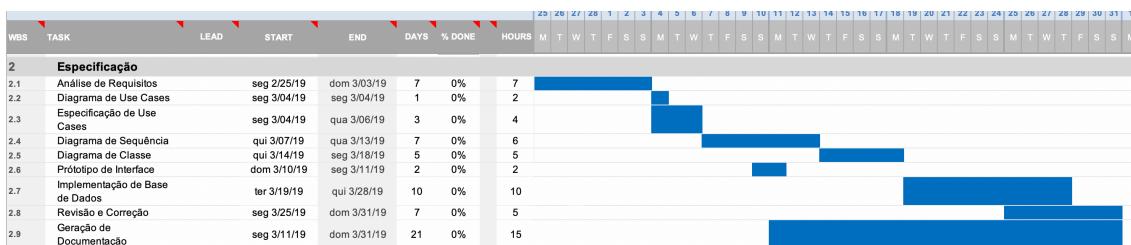


Figura 2 - Diagrama de Gantt: Previsão da Especificação

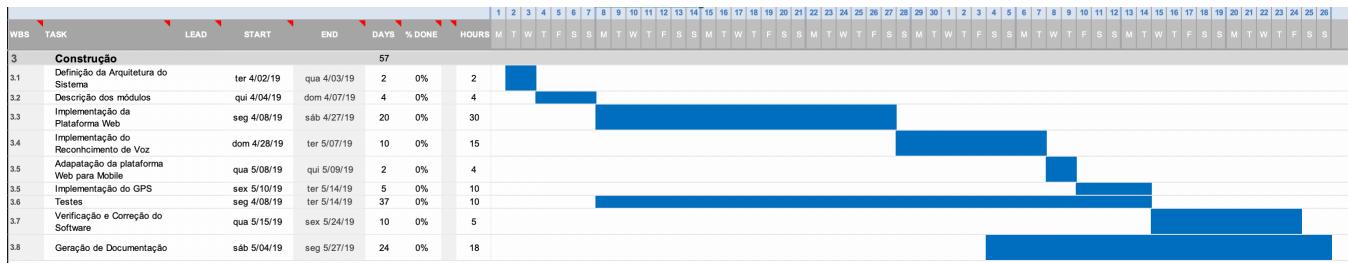


Figura 4 - Diagrama de Gantt: Previsão da Construção

Acima apresentam-se três imagens que correspondem à previsão que o grupo fez atendendo à carga trabalho que irá ter. É portanto, de referir que uma vez que as fases de Especificação e Construção são posteriores à fase atual, o grupo apenas pôde apresentar uma previsão das tarefas que pensa que irão ser necessárias, não sendo possível prever com exatidão quais realmente o serão.

Ao longo deste projeto, o grupo tentará cumprir ao máximo o planeamento previsto para cada fase, contudo para se realçar que o desenvolvimento dum produto de software pode não ser tão exato quanto o planeado, tentaremos fazer uma comparação com o que foi previsto e com o que realmente foi executado.

Tal já pode ser visto na seguinte imagem, correspondente à presente fase (Fundamentação):

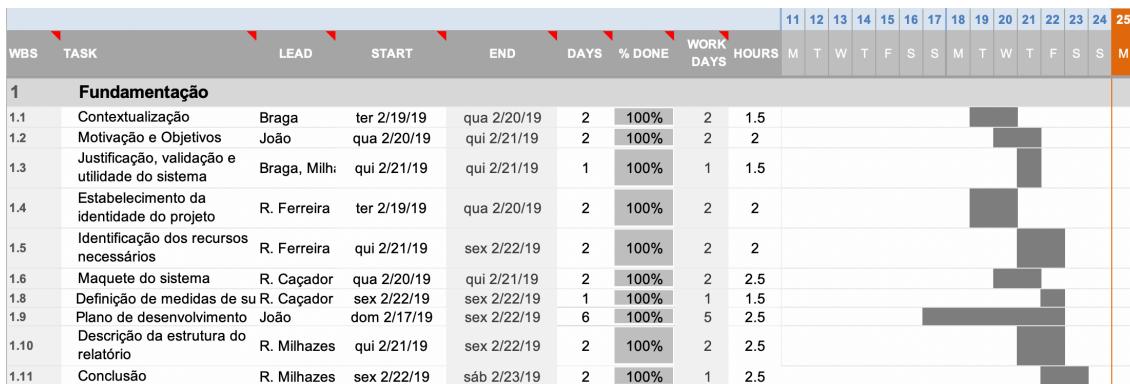


Figura 5 - Diagrama de Gantt: Realidade da Fundamentação

Nesta primeira fase notam-se já algumas diferenças com a previsão feita, fruto da falta de organização inicial do trabalho. O grupo espera que nas seguintes fases tal discrepância não aconteça, uma vez que elas já estão bem definidas.

A divisão de tarefas nas fases seguintes será realizada no começo da mesma. Esta divisão terá em conta uma discussão da equipa por forma a otimizar o projeto, tendo em conta os pontos fortes de cada elemento.

Para já, foi alocado para cada tarefa um período de tempo que achamos adequado para a sua realização, contudo estes tempos são estimados o que pode querer dizer que não sejam cumpridos à risca. Numa fase mais avançada do projeto iremos atualizar esta secção para que seja possível entender as diferenças entre o que foi previsto e o que realmente foi feito.

1.9. Descrição da estrutura do relatório

Como foi exposto na presente secção, esta introduz o problema a resolver e consequentemente o procedimento necessário para a realização do projeto.

Na secção 2 iremos proceder ao levantamento de todos os requisitos necessários para a realização de um Software que satisfaçam as necessidades dos clientes que o irão utilizar.

Na secção 3 iremos apresentar toda a modelação que envolve o projeto. Vamos apresentar o Diagrama de *Use Cases*, bem como as suas especificações e também os seus Diagramas de Sequência respetivos. De seguida iremos apresentar o Diagrama de Classes. Se for possível, ainda que não conste no Diagrama de Gantt acima apresentado, será realizada uma modelação um pouco mais completa (possíveis diagramas tais como Diagramas de Máquinas de Estado e Diagramas de Atividade).

Na Secção 4 será apresentada toda a estruturação e implementação da Base de Dados. Esta secção incluirá o Modelo Lógico onde vamos explicar todas as entidades,

relacionamentos e atributos. Iremos também identificar as chaves primárias, candidatas e alternativas.

Na Secção 5 vamos mostrar e explicar a interface criada para o Software *FeelItaly* bem como as funcionalidades que poderão ser executadas em cada janela de interface.

Na Secção 6 serão apresentados todos os detalhes de implementação. Em primeiro lugar será explicada a implementação da plataforma Web seguida da sua respetiva adaptação a Mobile. De seguida serão apresentados os detalhes de implementação relativos ao GPS e ao reconhecimento de voz. Finalmente, será reservado um campo para a amostra de alguns testes feitos durante a implementação do programa e os seus respetivos resultados.

Por último, na secção 7, passaremos a apresentar as reflexões finais e uma apreciação do sistema construído de uma forma construtiva e justificada.

2. Requisitos

2.1. Requisitos do Utilizador e de Sistema Funcionais

2.1.1 Registo no sistema

Definição de Requisito de Utilizador
1. O utilizador deve conseguir registar-se no sistema.
Especificação dos Requisitos de Sistema
1.1. O sistema deve solicitar o e-mail, o nome, o username e a password para autenticação.
1.2. O sistema não deve permitir o registo de utilizadores com mail ou username já em utilização.
1.3. O sistema deve conseguir armazenar os dados referentes a um novo utilizador.

2.1.2 Autenticação no sistema

Definição de Requisito de Utilizador
1. O utilizador deve conseguir autenticar-se no sistema.
Especificação dos Requisitos de Sistema
1.1. O sistema deve solicitar o username e a password para autenticação.
1.2. O sistema deve verificar a validade dos dados inseridos.
1.3. O sistema deve associar uma sessão a um utilizador que se tenha autenticado.
1.4. O sistema à entrada deve sugerir, conforme as estatísticas do utilizador, que receitas cozinar e outras que contenham os ingredientes mais utilizados.

2.1.3 Estabelecimento de uma configuração inicial de pratos e ingredientes

Definição de Requisito de Utilizador
1. O utilizador deve conseguir estabelecer uma configuração inicial de pratos e ingredientes.
Especificação dos Requisitos de Sistema
1.1. O sistema deve mostrar uma pré-visualização das receitas existentes (imagem e nome) para possível seleção. 1.2. O sistema deve conseguir armazenar os dados referentes à configuração do utilizador.

2.1.4 Visualização das estatísticas

Definição de Requisito de Utilizador
1. O utilizador deve conseguir visualizar as estatísticas relativas ao seu uso da aplicação.
Especificação dos Requisitos de Sistema
1.1. O sistema deve mostrar os dados relativos aos cozinhados realizados, os tempos de preparação, as dificuldades encontradas e os ingredientes utilizados. 1.2. O sistema deve atualizar as estatísticas consoante a utilização da aplicação por parte do utilizador.

2.1.5 Listagem de receitas

Definição de Requisito de Utilizador
2. O utilizador deve conseguir pré-visualizar todas as receitas disponíveis.
Especificação dos Requisitos de Sistema
2.1. O sistema deve mostrar uma pré-visualização das receitas existentes (imagem e nome).

2.1.6 Filtragem de receitas

Definição de Requisito de Utilizador
1. O utilizador deve conseguir filtrar as receitas por ingrediente(s)/tipo de prato/tipo de refeição/tempo de confecção.
Especificação dos Requisitos de Sistema
1.1. O sistema deve disponibilizar os filtros pré-definidos.

1.2. O sistema deve filtrar as receitas consoante a preferência do utilizador.
1.3. O sistema deve listar as receitas antes filtradas.

2.1.7 Comunicação via voz

Definição de Requisito de Utilizador
1. O utilizador deve conseguir comunicar com o sistema via voz.
Especificação dos Requisitos de Sistema
1.1. O sistema deve reconhecer comandos via voz pré-definidos, referentes à preparação de uma receita.
1.2. O sistema deve responder a estímulos do utilizador.

2.1.8 Seleção de uma receita

Definição de Requisito de Utilizador
1. O utilizador deve conseguir selecionar uma receita.
Especificação dos Requisitos de Sistema
1.1. O sistema deve apresentar o nome duma receita.
1.2. O sistema deve apresentar os ingredientes necessários duma receita.
1.3. O sistema deve apresentar o tempo de preparação duma receita.
1.4. O sistema deve apresentar todos os passos necessários para a sua confeção duma receita.
1.5. O sistema deve apresentar o conteúdo calórico total duma receita.
1.6. O sistema deve apresentar os comentários duma receita.
1.7. O sistema deve apresentar a avaliação duma receita.

2.1.9 Preparar uma receita

Definição de Requisito de Utilizador
1. O utilizador deve ser acompanhado durante a confeção duma receita.
Especificação dos Requisitos de Sistema
1.1. O sistema deve descrever textualmente cada passo da confeção duma receita.
1.2. O sistema deve mostrar os ingredientes do passo em causa.
1.3. O sistema deve contar o tempo que o utilizador demora a realizar cada passo.
1.4. Cada passo deve ter associado a si um tempo de execução.
1.5. O sistema deve providenciar links para tutoriais (vídeos, sites, etc) que expliquem detalhadamente a realização de cada passo.
1.6. O sistema deve passar para o passo seguinte consoante a ordem do utilizador (via voz ou clique).
1.7. O sistema deve repetir o passo consoante a ordem do utilizador (via voz ou clique).
1.8. O sistema deve voltar para o passo anterior consoante a ordem do utilizador (via voz ou clique).
1.9. O sistema deve estender uma sub-receita que esteja contida num dos passos da receita principal, consoante a ordem do utilizador (via voz ou clique).
1.10. O sistema deve guardar na configuração do utilizador as estatísticas associadas à receita terminada.
1.11. O sistema deve guardar na configuração do utilizador as dificuldades que este teve na confeção da receita terminada.

2.1.10 Avaliação de uma receita

Definição de Requisito de Utilizador
1. O utilizador deve poder avaliar uma receita.
Especificação dos Requisitos de Sistema
1.1. O sistema deve providenciar uma secção onde o utilizador possa inserir a sua avaliação.

2.1.11 Comentar uma receita

Definição de Requisito de Utilizador
1. O utilizador deve poder comentar uma receita.

Especificação dos Requisitos de Sistema
1.1. O sistema deve providenciar uma área de texto onde o utilizador possa inserir o seu comentário.

2.1.12 Adição de uma receita

Definição de Requisito de Utilizador
1. O utilizador deve poder adicionar uma receita.
Especificação dos Requisitos de Sistema
1.1. O sistema deve solicitar os dados da receita.

1.2. O sistema deve deixar a nova receita pendente para verificação posterior pelo administrador.

2.1.13 Agendamento de uma ementa semanal

Definição de Requisito de Utilizador
1. O utilizador deve poder agendar uma ementa semanal.
Especificação dos Requisitos de Sistema
1.1. O sistema deve providenciar espaços de inserção para as receitas da ementa semanal.
1.2. O sistema deve verificar os ingredientes presentes nas receitas incluídas na ementa semanal, e apresentar um possível plano de compras.
1.3. O sistema deve guardar a ementa semanal.

2.1.14 Gestão da despensa

Definição de Requisito de Utilizador
1. O utilizador deve poder inserir e remover os ingredientes presentes na sua despensa.
Especificação dos Requisitos de Sistema
1.1. O sistema deve providenciar uma secção com os ingredientes presentes na despensa do utilizador.

1.2. O sistema deve inserir ou remover os ingredientes presentes da referida secção.
--

2.1.15 Pesquisa de locais para comprar ingredientes em falta

Definição de Requisito de Utilizador
1. O utilizador deve poder localizar estabelecimentos que disponibilizem os produtos em falta.
Especificação dos Requisitos de Sistema
1.1. O sistema deve disponibilizar um mapa com as soluções encontradas para as necessidades de compra.

2.1.16 Listagem de locais para comprar ingredientes em falta

Definição de Requisito de Utilizador
1. O utilizador deve poder ter acesso a uma lista dos estabelecimentos que disponibilizem os produtos em falta.
Especificação dos Requisitos de Sistema
1.1. O sistema deve disponibilizar uma lista com as soluções encontradas para as necessidades de compra, por ordem de distância.

2.2. Requisitos de Sistema Não Funcionais

2.2.1 Requisitos de Produto

- O sistema deve estar disponível 24 horas por dia;
- O sistema deve ser de fácil uso;
- O sistema deve ser produzido de forma a funcionar corretamente em todos os browsers;
- O sistema deve suportar o registo de 1500 utilizadores e 300 receitas num espaço de 1 ano;
- A avaliação da aplicação e a satisfação dos utilizadores com a sua utilização deverá ser muito satisfeita.

2.2.2 Requisitos Organizacionais

- A aplicação será desenvolvida para web e para mobile;
- A linguagem de programação utilizada será C# juntamente com a framework ASP.net;
- O sistema geral de Base de Dados a usar será o SQL Server.

2.2.3 Requisitos Externos

- O sistema deve recorrer a uma API de mapas para representar a localização do utilizador e a localização de estabelecimentos comerciais de alimentos.
- O sistema deve manter a privacidade dos utilizadores e dos seus dados, sendo estes visíveis apenas para cada utilizador.
- O sistema deve recorrer a um sistema de reconhecimento de voz para reconhecer e efetuar as ações ditas pelo utilizador.
- O sistema deve recorrer a um sistema de voz para ditar e explicar as receitas ao utilizador.

3. Modelação UML

Após a recolha e descrição detalhada dos requisitos, cabe-nos modelar o sistema que irá ser construído. A modelação pode ser vista como uma abstração do sistema que será construído (abstrai muito do detalhe de implementação), e não como uma representação alternativa do mesmo.

Nesta secção abordamos então, a modelação realizada para o sistema. Para tal utilizamos **UML** que é uma linguagem-padrão para a elaboração da estrutura de projetos de software. Para uma especificação fundamentada do produto construído, o grupo decidiu elaborar, como é possível de notar ao longo desta secção, modelos de contexto, modelos de interação, modelos estruturais, e por último, modelos comportamentais.

Os modelos criados em **UML** ajudaram a derivar alguns dos requisitos do sistema, futuramente ajudarão na tarefa de implementação e mais tarde ainda, quando for necessário documentar a estrutura do sistema e o modo como opera.

3.1. Modelos de Contexto

Este tipo de modelos devem ser os primeiros a serem elaborados uma vez que representam ainda uma visão muito superficial e primordial sobre o sistema. É nestes modelos que traçamos as fronteiras do sistema, basicamente o que este deve ou não deve fazer. São produzidos para que possam ser discutidos num nível muito elevado de abstração com os stakeholders.

Devem ser construídos baseando-se na simplicidade e num conceito minimalista, mas que sejam uma representação real do sistema a construir e consigam responder a todos os requisitos definidos.

Desta forma, elaboramos um modelo de domínio, e um diagrama de atividade que representa as áreas que realmente são o cerne do sistema (preparar/aprender uma receita).

3.1.1 Modelo de Domínio

Este modelo começou a ser produzido após uma análise dos requisitos levantados. Começamos por identificar as entidades/intervenientes no sistema e de que forma estas poderiam interagir entre si. Inicialmente demos destaque a 5 entidades que pudessem ser as mais relevantes no sistema, que são respetivamente, o **Utilizador**, o **Administrador**, a **Receita**, o **Ingrediente** e o **Passo** duma receita.

A evolução deste modelo foi algo que esteve sempre em aberto e o resultado final não é um diagrama que se possa dizer que esteja bem ou mal, mas após a especificação e adição de algumas entidades, e a remodelação de algumas interações entre entidades, chegamos a um resultado final que acreditamos que identifique na íntegra o domínio do sistema.

A seguinte figura representa o modelo de domínio construído.

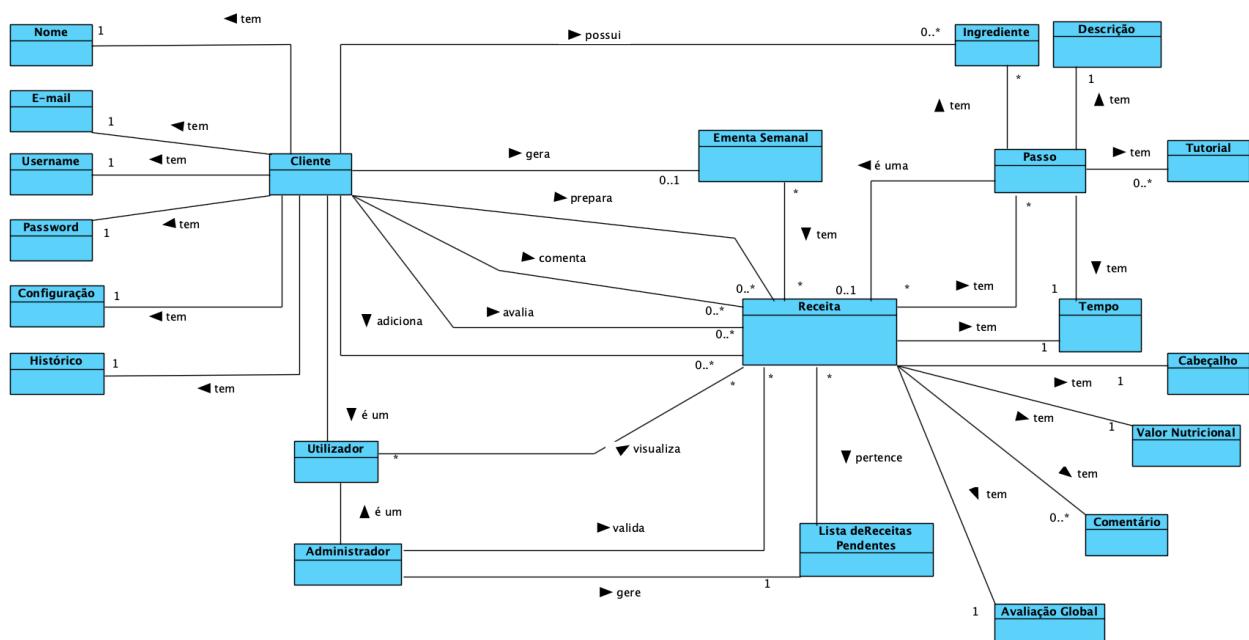


Figura 6 - Modelo de Domínio

De referir ainda que o modelo de domínio é um exemplo de diagrama de classes que não é explorado totalmente, faz parte da modelação estrutural do sistema. É usado nesta etapa de modelação pois providencia uma forma muito trivial de representar o sistema.

3.1.2 Diagrama de Atividade

Na tentativa de reproduzir um modelo que fosse uma representação muito próxima daquilo que será implementado, mas ainda com um nível de abstração relativamente alto, construímos dois diagramas de atividade que representam as várias atividades e os fluxos de acontecimentos. Neste caso representam as mudanças de ações que ocorrem aquando da pré-visualização e seleção duma receita e da preparação/aprendizagem duma receita.

Para tal identificamos as duas principais entidades que estarão envolvidas neste processo, o utilizador e o sistema.

O primeiro diagrama de atividade representa as ações que tanto o utilizador como o sistema executam aquando da pré-visualização e seleção duma receita.

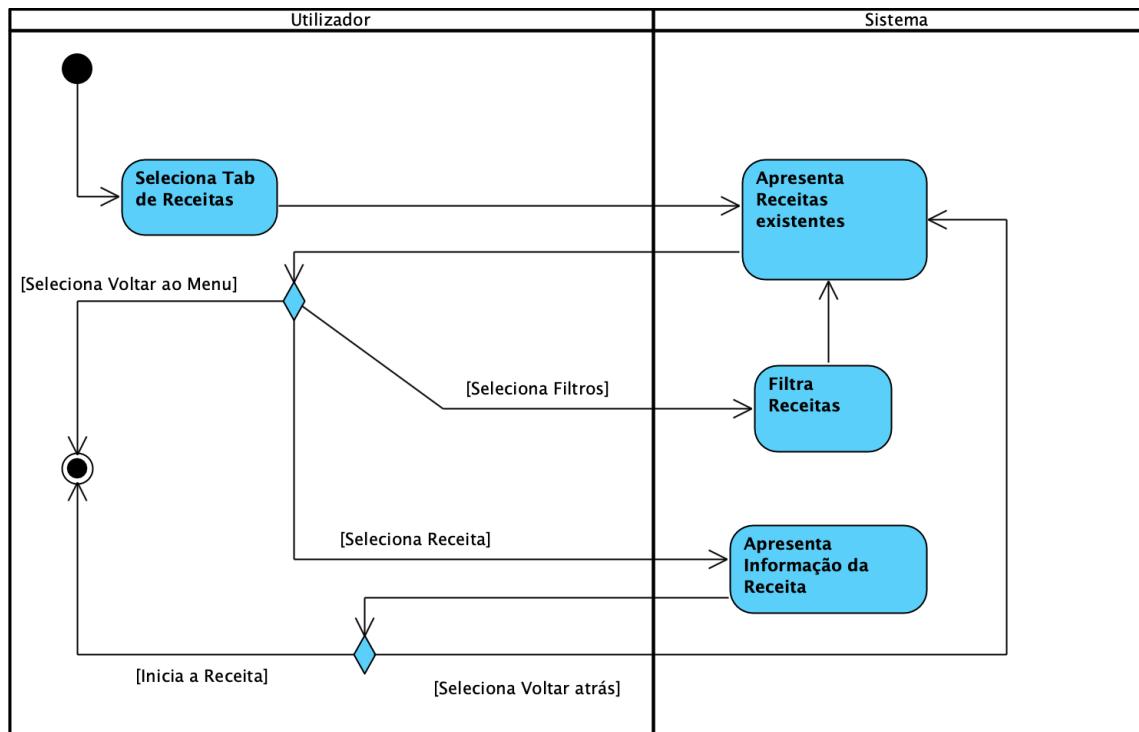


Figura 7 - Diagrama de Atividade de Seleção duma Receita

O segundo diagrama representa as ações entre o utilizador e o sistema aquando da preparação/aprendizagem duma receita.

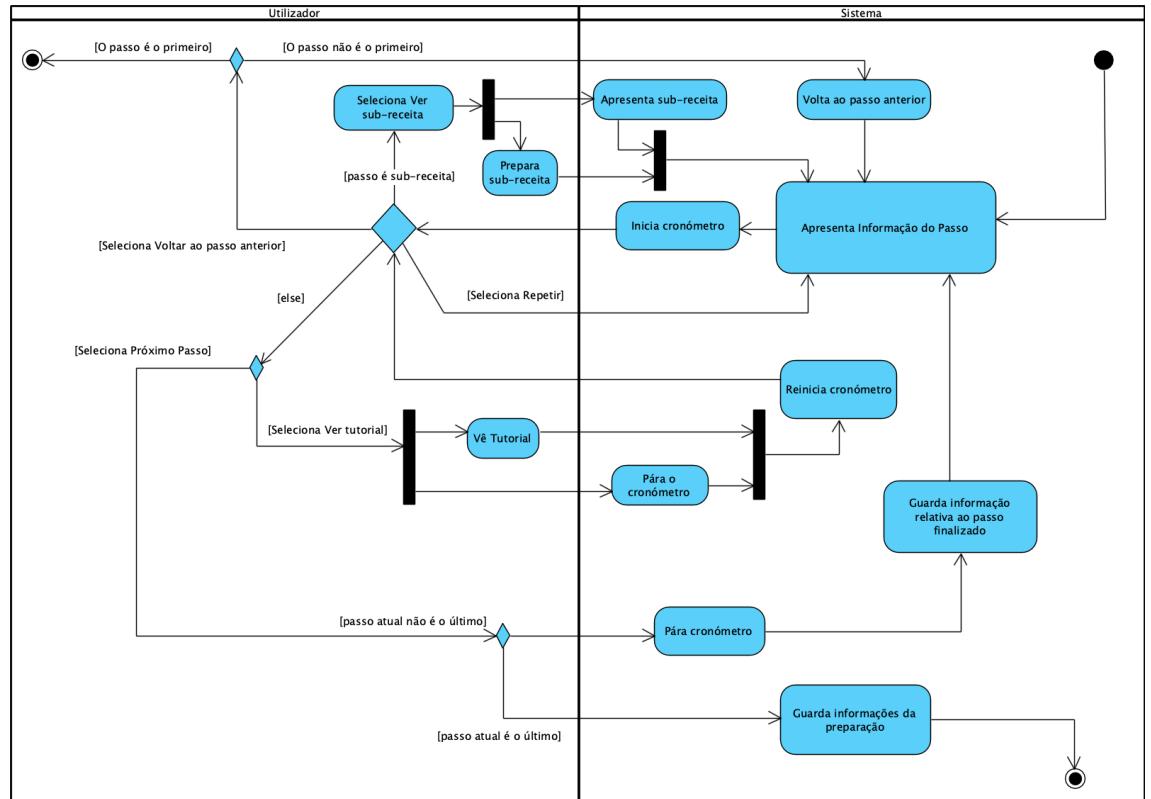


Figura 8 - Diagrama de Atividade de Preparação duma Receita

De referir ainda que os diagramas de atividade são um exemplo de diagramas comportamentais, que especificam atividades e fluxos de acontecimentos. São usados nesta etapa de modelação pois providenciam uma forma muito trivial de representar o sistema e ter uma ideia para trabalho futuro de como estas ações possivelmente se comportarão.

3.2. Modelos de Interação

Tendo em conta que todos os sistemas envolvem interação, com o utilizador ou mesmo entre as componentes do próprio sistema, é importante elaborar estes modelos. Para além de uma boa noção da interação no sistema, estes modelos proporcionam uma ajuda importante para identificar requisitos de utilizador.

Neste capítulo vamos apresentar duas formas de modelar a interação no sistema, que diferem no nível de detalhe que são construídos, baseando-nos no diagrama de Use Cases.

Primeiro, especificámos textualmente os use cases, utilizando o formato tabular.

Segundo, construímos os respetivos diagramas de sequência, que apresentam de forma mais representativa a interação entre utilizador e sistema uma vez que se consegue perceber claramente que mensagens são trocadas entre os intervenientes e que tipo de operações o sistema faz em resposta a uma mensagem do utilizador.

3.2.1 Diagrama de Use Cases

Um use case pode ser definido como um cenário que descreve o que o utilizador espera do sistema. Desta forma, cada use case representa uma tarefa que envolve interação com o sistema.

O diagrama de Use Cases fornece, portanto, uma visão geral de toda a interação possível entre o utilizador e o sistema.

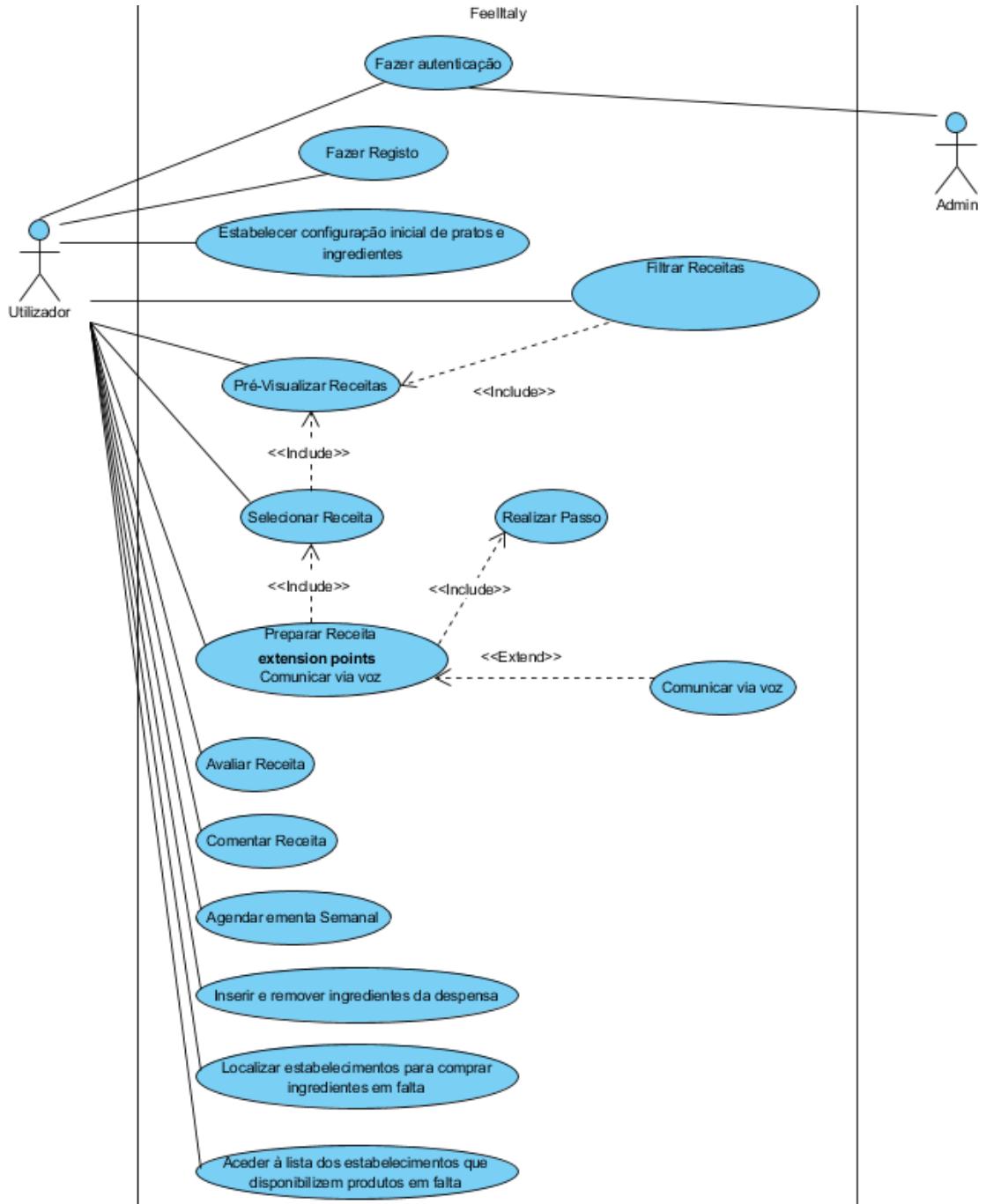


Figura 9 - Diagrama de Use Cases

Ao longo da realização do relatório vamos modelar e especificar os Use Cases que achamos fundamentais ao funcionamento do nosso produto de software. Neste caso estamos a

falar dos Use Cases: Pré-Visualizar Receitas, Selecionar Receita, Preparar Receita e Realiza Passo.

3.2.2 Especificação Textual de Use Cases

O subcapítulo anterior mostra a aglomeração de todos os use cases que o utilizador pode requerer ao sistema. No entanto, é também muito importante saber detalhadamente como cada uma das ações progride, e perante esta necessidade, surge a especificação textual de use cases.

Neste relatório, optamos por especificar os use cases através da descrição tabular.

Use Case:	Pré-visualizar receitas	
Actor:	Utilizador	
Pré condição:	Estar logado	
Pós condição:	Lista de receitas disponibilizadas	
Cenário Normal	Actor input	System response
	1. Solicita lista de receitas	2. Apresenta lista de receitas

Figura 10 - Use Case Pré-Visualizar Receitas

Use Case:	Filtrar Receitas	
Actor:	Utilizador	
Pré condição:	Estar logado	
Pós condição:	Lista de receitas disponibilizadas	
Cenário Normal	Actor input	System response
	1. <<include>> Pré-visualizar Receitas	2. Apresenta filtros pré-definidos
Exceção 1 [Não encontra receitas para a filtragem pretendida] (passo 4)	3. Seleciona filtros pretendidos	4. Procura receitas na base de dados. 5. Apresenta nova lista de receitas consoante os filtros selecionados
		4.1. Indica que não existem receitas disponíveis.

Figura 11 - Use Case Filtrar Receitas

Use Case:	Selecionar Receita	
Actor:	Utilizador	
Pré condição:	Estar logado	
Pós condição:	Receita iniciada	
Cenário Normal	Actor input	System response
	1.<<include>> Pré-visualizar Receitas	
	2. Seleciona Receita	
	3. Apresenta dados da receita	
Exceção 1 [Voltar](passo 4)		4.1. Retorna à janela de pré-visualização

Figura 12 - Use Case Selecionar Receita

Use Case:	Preparar Receita	
Actor:	Utilizador	
Pré condição:	Estar logado	
Pós condição:	Receita confeccionada	
Cenário Normal	Actor input	System response
	1. <<include>> Selecionar Receita	2. Inicia cronómetro global
		3. Para cada passo:
		3. a) <<include>> Realizar Passo
		3. b) Guarda as estatísticas do utilizador neste passo
	4. Termina a receita	5. Termina cronómetro
Cenário Alternativo 1 [Valor retornado é "Voltar"] (passo 3. a))		6. Guarda as estatísticas do utilizador
		3. a). 1. Volta uma iteração atrás
Cenário Alternativo 2 [Valor retornado é "Repetir"] (passo 3. a))		3. a). 2. Retorna a 3.
		3. a). 1. Repete a iteração
		3. a). 2. Retorna a 3.

Figura 13 - Use Case Preparar Receita

Use Case:	Realizar Passo	
Actor:	Utilizador	
Pré condição:	Estar a preparar receita	
Pós condição:	Passo realizado	
Cenário Normal	Actor input	System response
		1. Inicia cronómetro parcial
		2. Apresenta os ingredientes do passo
		3. Apresenta textualmente o passo
		4. Apresenta tempo de execução do passo
		5. Apresenta links para tutoriais que expliquem o passo
	6. Próximo Passo	6. Espera por decisão do utilizador
		7. Termina cronómetro
		8. Informa que o passo foi realizado
Exceção 1 [Voltar] (Passo 6)		6.1. Termina cronómetro 6.2. Retorna "Voltar"
Exceção 2 [Repetir] (Passo 6)		6.1. Termina cronómetro 6.2. Retorna "Repetir"

Figura 14 - Use Case Realizar Passo

3.2.3 Diagramas de Sequência de Sistemas

Com base nos usos cases e nas respetivas especificações construímos os Diagrama de Sequência de Sistemas para cada um dos use cases que envolvem o processo de Preparar Receita. Estes diagramas permitem-nos compreender com um maior pormenor o fluxo de eventos necessário para o bom funcionamento do nosso sistema.

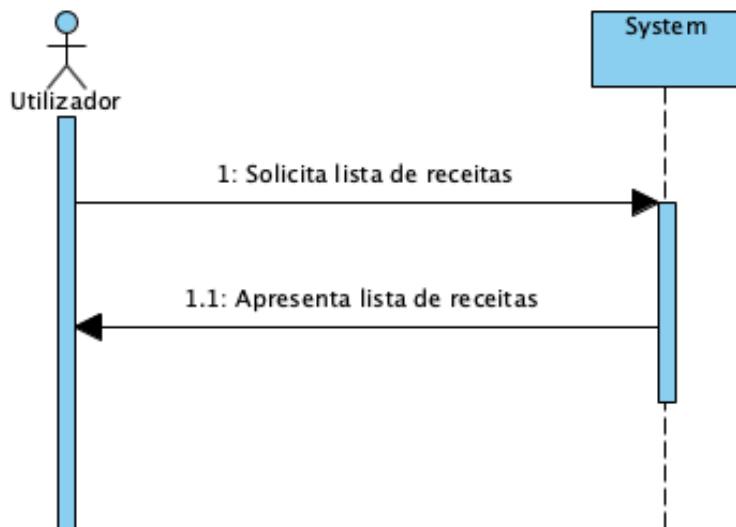


Figura 15 - DSS Pré-Visualizar Receitas

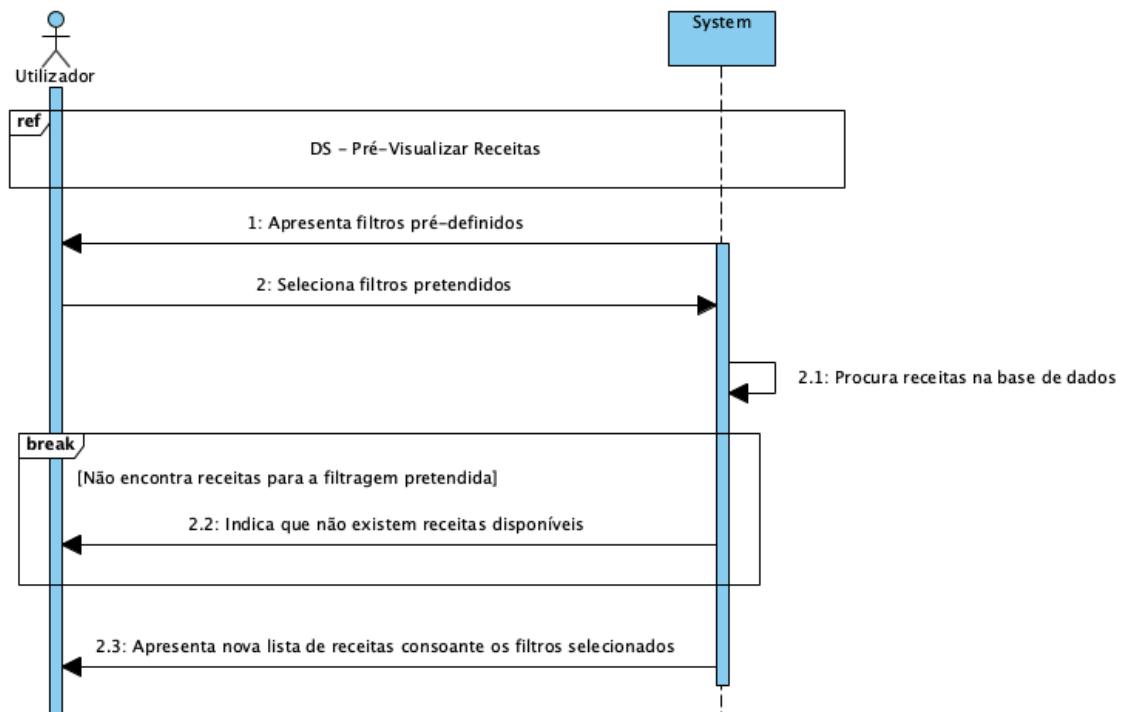


Figura 16 - DSS Filtrar Receitas

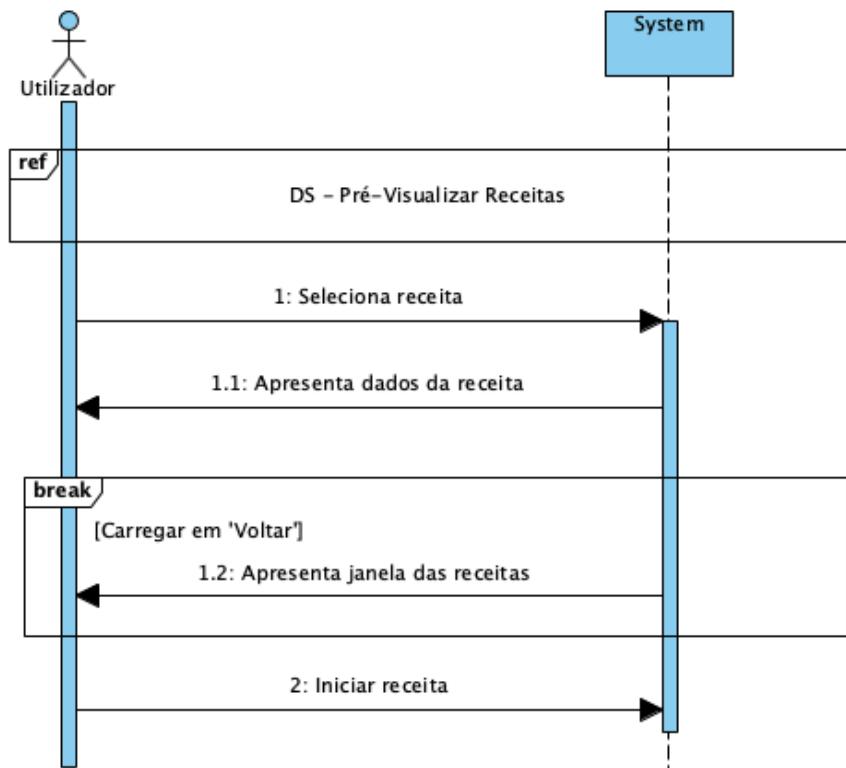


Figura 17 - DSS Selecionar Receita

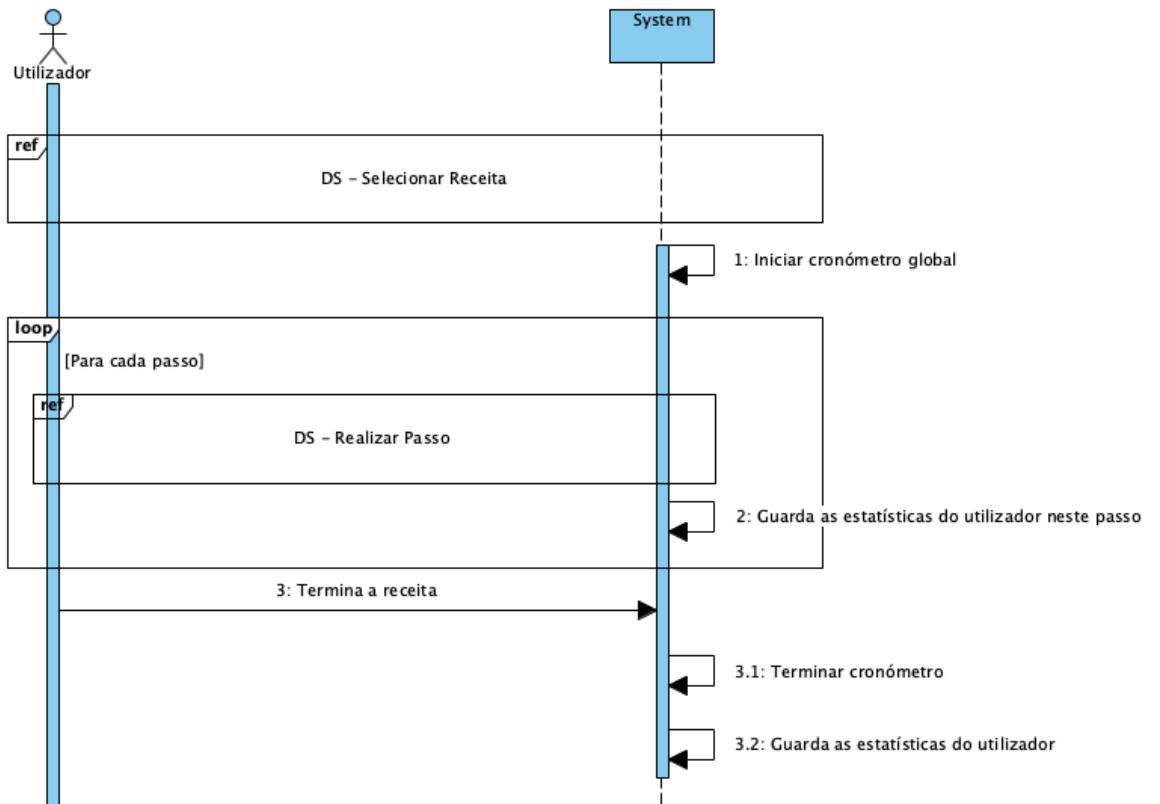


Figura 18 - DSS Preparar Receita

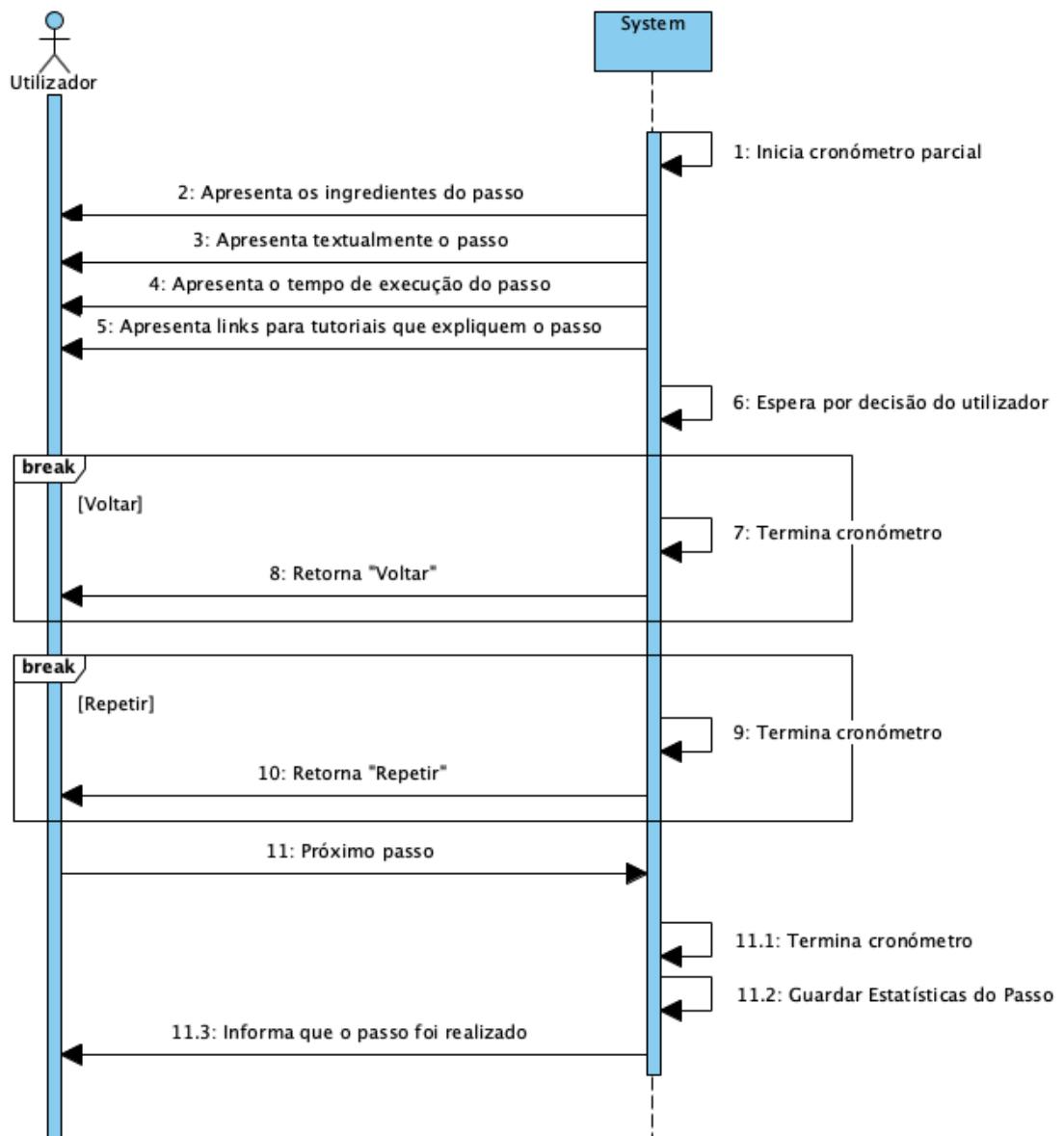


Figura 19 - DSS Realizar Passo

3.2.4 Diagramas de Sequência de Implementação

Os diagramas de sequência são usados para modelar a interação entre o utilizador e os objetos do sistema, assim como as interações entre os próprios objetos do sistema.

Deste modo, um diagrama destes tem como objetivo mostrar a sequência de interações durante um use case em particular. Para estes diagramas baseámo-nos no diagrama de classes ORM apresentado na subsecção 3.3.2.

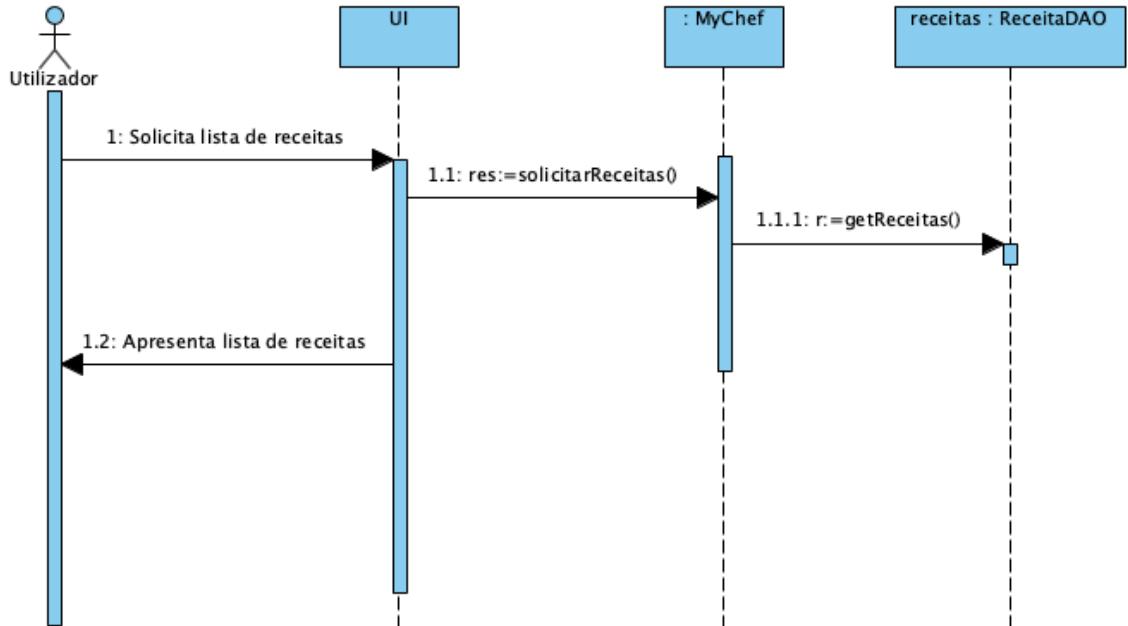


Figura 21 - DS Implementação Pré-Visualizar Receitas

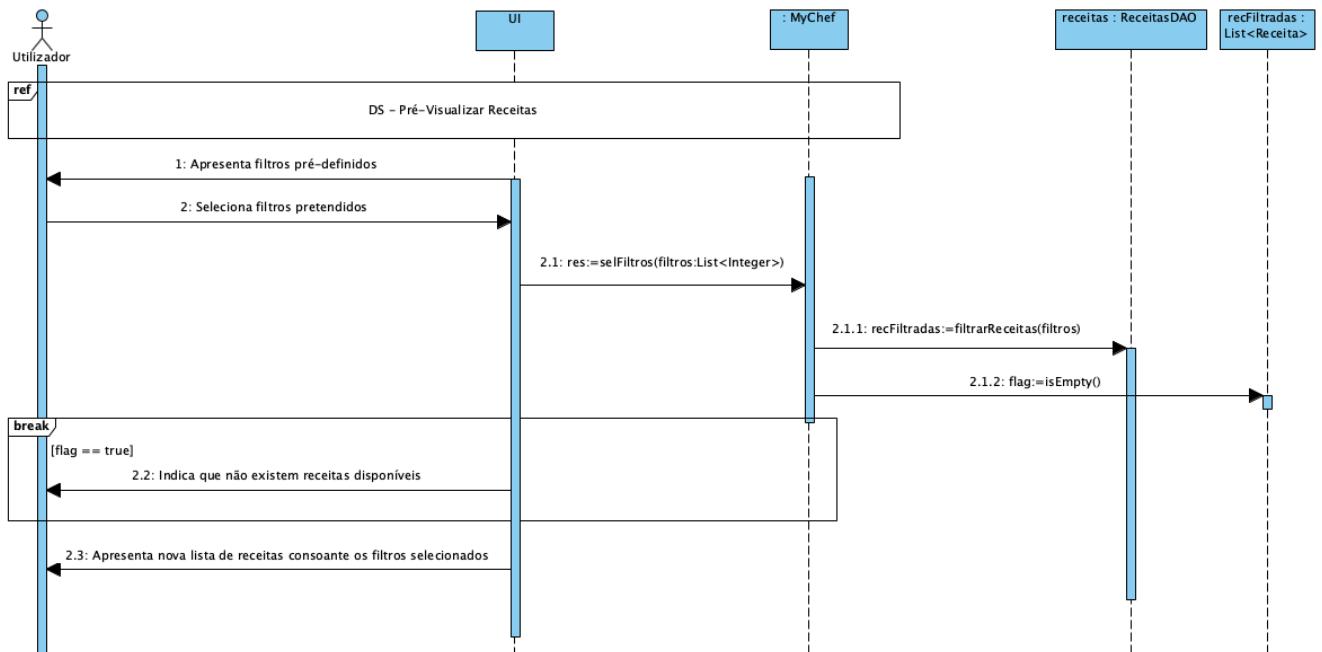


Figura 20 - DS Implementação Filtrar Receitas

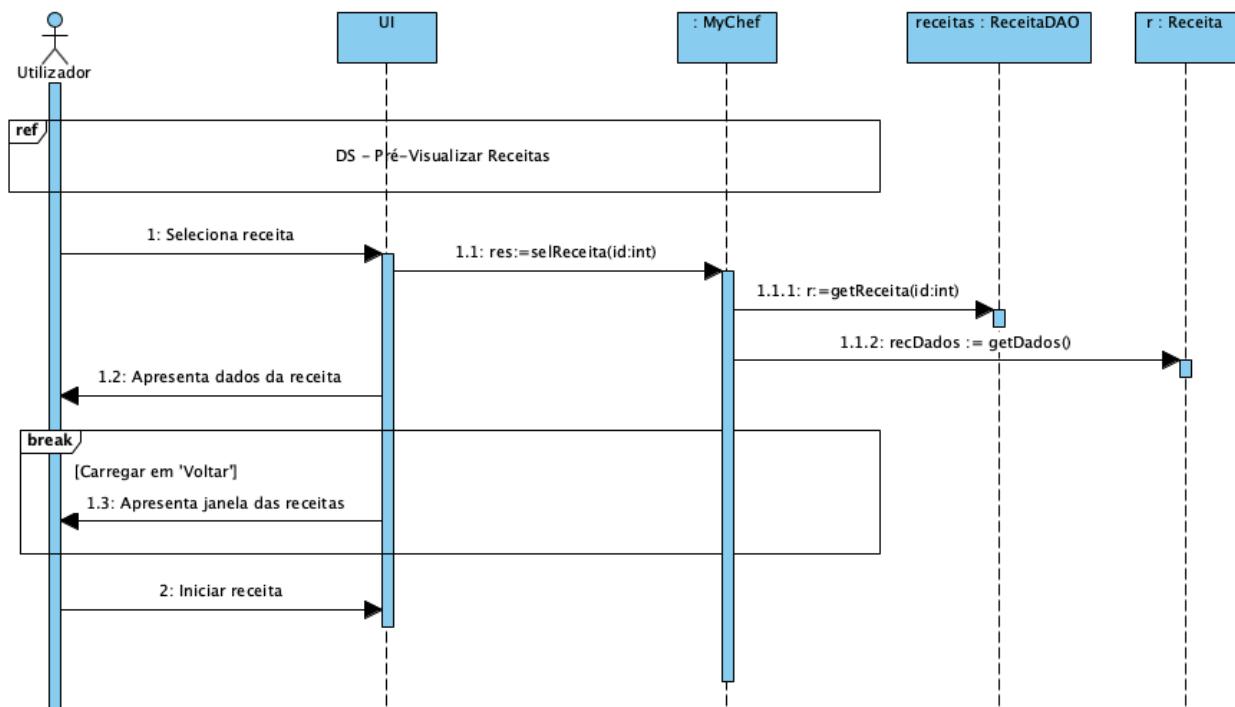


Figura 22 - DS Implementação Selecionar Receita

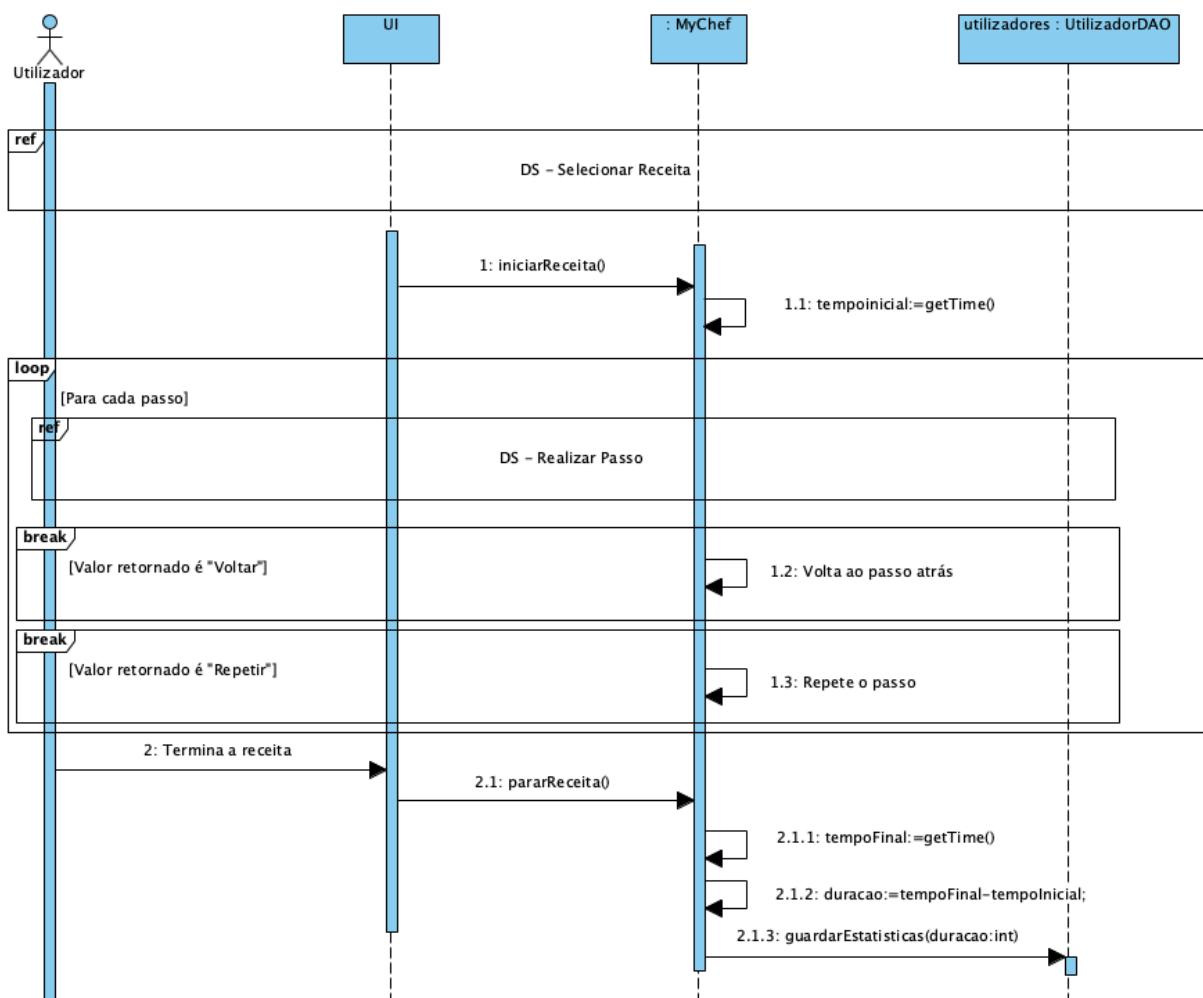


Figura 23 - DS Implementação Preparar Receita

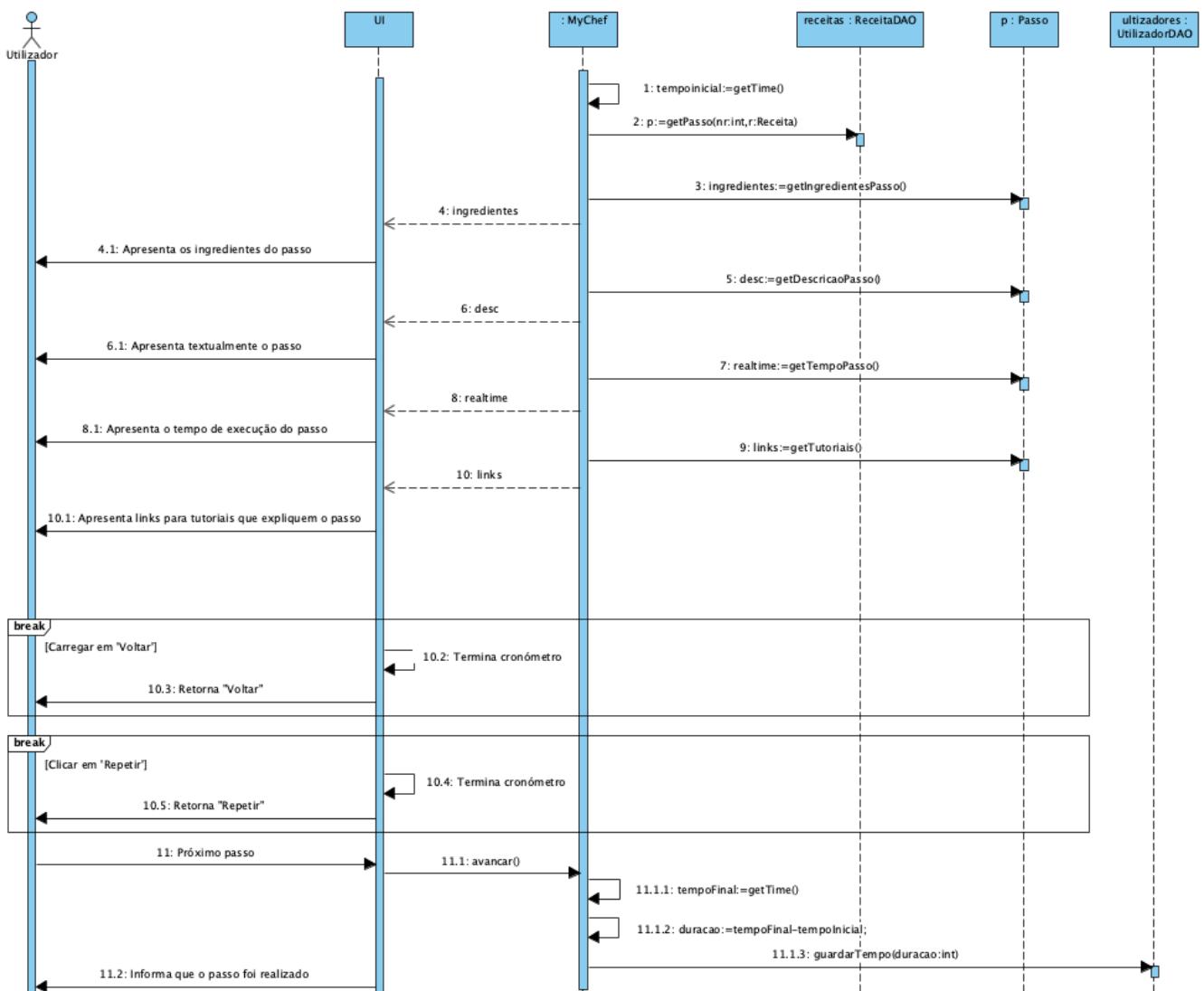


Figura 24 - DS Implementação Realizar Passo

3.3. Modelos Estruturais

Os modelos estruturais demonstram a organização de um sistema relativamente às componentes e às relações em que este se baseia. Este género de modelos são mais uma fase muito importante na modelação do sistema, porque é neste momento que é discutida e estabelecida a arquitetura do sistema que será construído.

Neste capítulo, vamos abordar os diagramas de classe com a intenção de modelar uma estrutura estática das classes do sistema de software.

3.3.1 Diagrama de Classe

Como já foi introduzido, um diagrama de classes é usado para desenvolver um sistema orientado a objetos que mostre as classes nele presentes e as associações entre essas mesmas classes.

É nesta parte da modelação que, no contexto do projeto, são identificados os objetos essenciais para o seu funcionamento, e é realizada a representação destes mesmos em classes.

O diagrama de classes abaixo apresentado, semanticamente, tem parecenças com o modelo de dados apresentado anteriormente. Tal não acontece por acaso, pois um diagrama de classes tenta mostrar as entidades, os atributos e as relações entre as próprias entidades apresentadas num modelo de dados, só que num contexto orientado a objetos

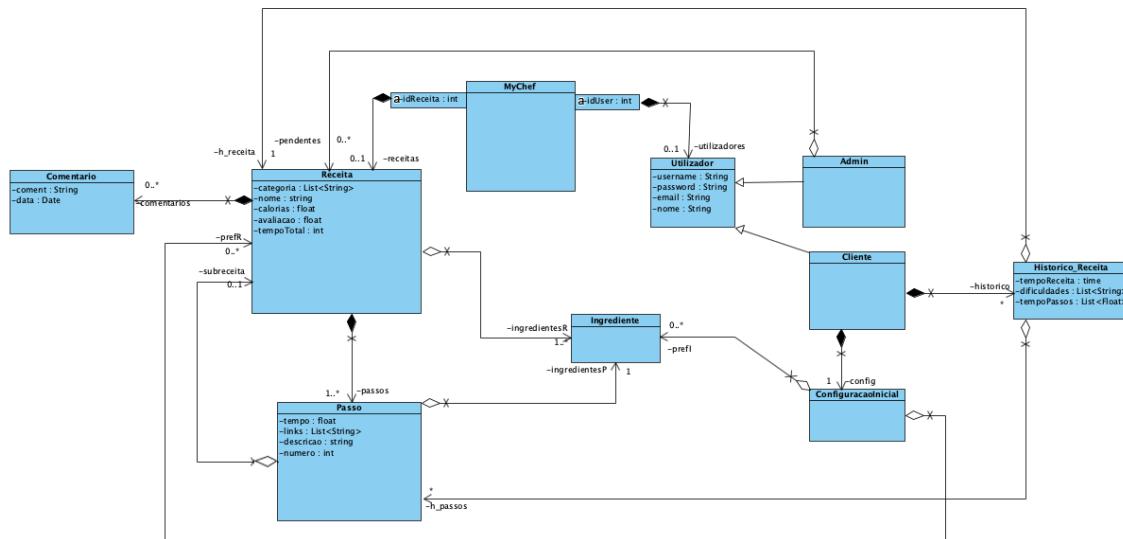


Figura 25 - Diagrama de Classe

3.3.2 Diagrama de Classe ORM

Para o diagrama de Classe ORM o grupo sentiu a necessidade de criar 3 classes novas: UtilizadorDAO, ReceitaDAO e IngredienteDAO.

A classe utilizadorDAO é necessária para poder ser feito o acesso a todos os utilizadores do sistema, sejam eles administradores ou clientes.

Já a classe ReceitaDAO é necessária para poder aceder a todas as receitas existentes no sistema bem como tudo o que a si está relacionado.

Por fim, achamos necessária a criação da classe IngredienteDAO pois é possível que existam ingredientes que não apareçam em receitas e assim nunca seriam acedidos. Com esta classe, é possível aceder a todos os ingredientes existentes.

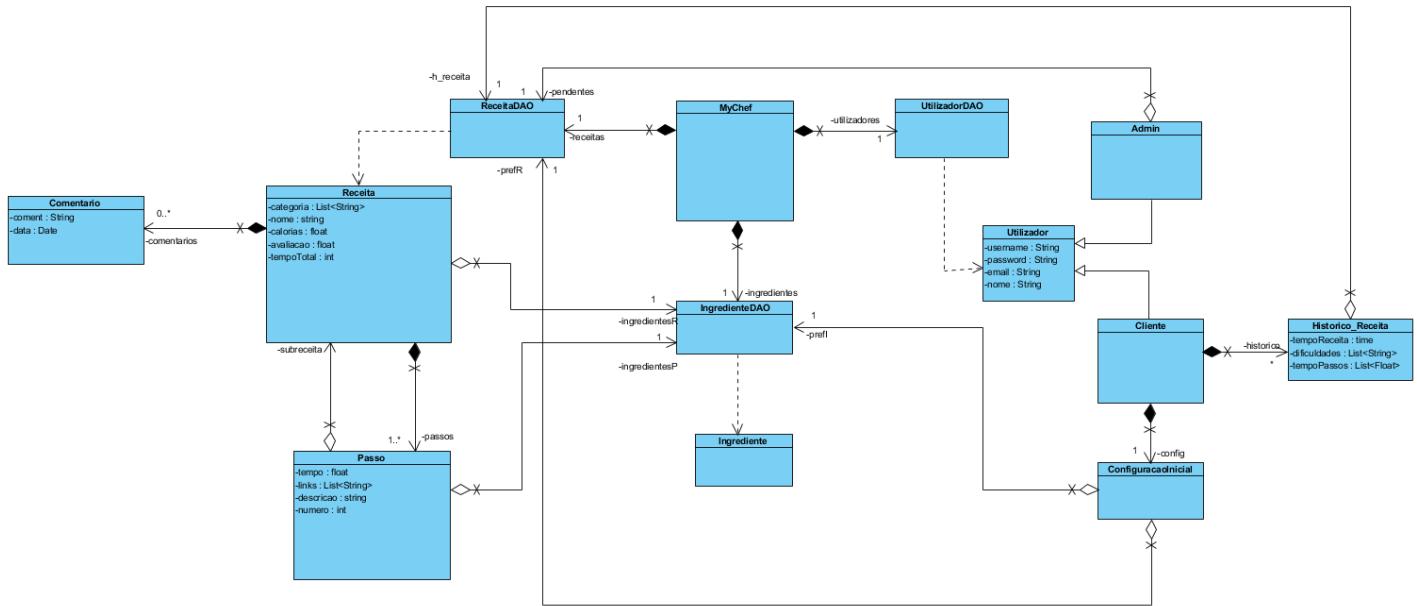


Figura 26 - Diagrama de Classe ORM

3.4. Modelos Comportamentais

Estes modelos têm a sua importância na modelação pois representam o comportamento do sistema quando este está em execução. Eles mostram o que acontece quando o sistema responde a um qualquer acontecimento que esteja sujeito.

São dois os tipos de acontecimentos aos quais o sistema está sujeito. Pode ocorrer chegada de dados ou então acontecimento de um evento, e nos dois casos o sistema tem que ter capacidade de os processar.

Neste caso vamos modelar o comportamento do sistema tendo em conta a prototipagem construída e apresentada no Capítulo 5.

De referir por último, que existem vários estados que poderiam ser modelados de forma mais específica pois não são de todo triviais. Optámos por modelar alguns casos que retratam diferentes abordagens na modelação mas que constituem uma imagem exemplificativa do método que usamos para modelar os estados do produto.

3.4.1 FeelItaly

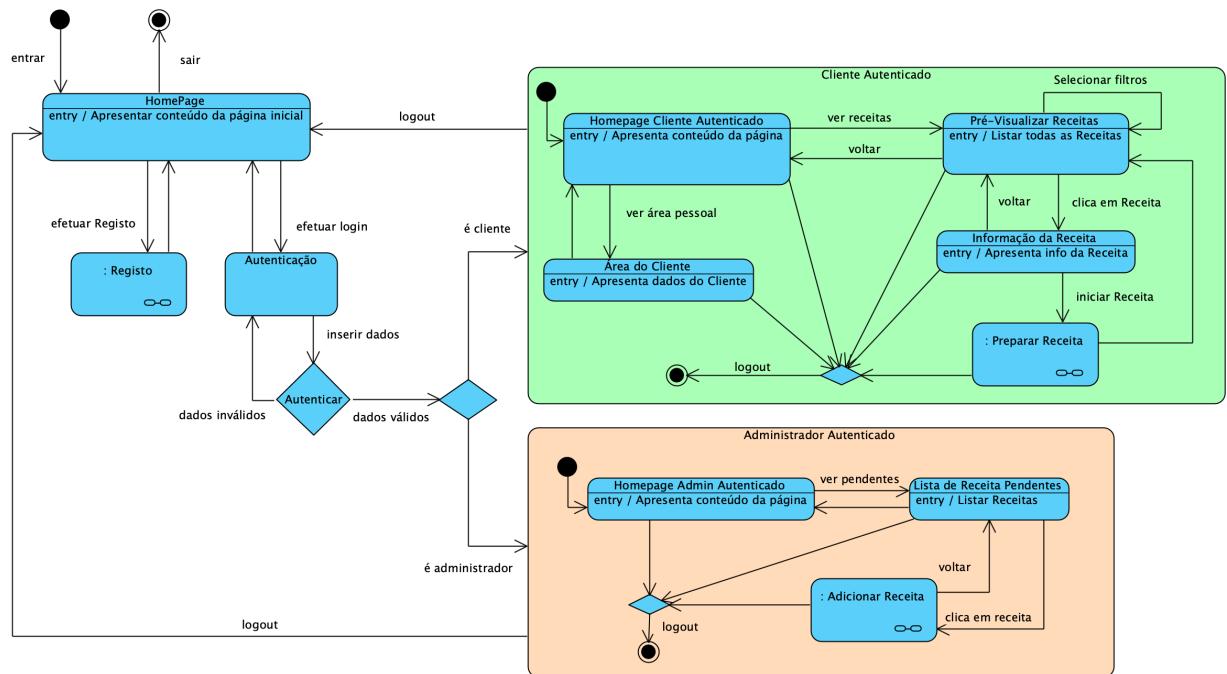


Figura 26 - Diagrama Geral do Sistema com as respectivas submáquinas

Este é o diagrama que exemplifica os estados no sistema. Nele estão contidos a maior parte dos estados e sub-estados que o sistema pode atingir.

3.4.2 Preparar Receita

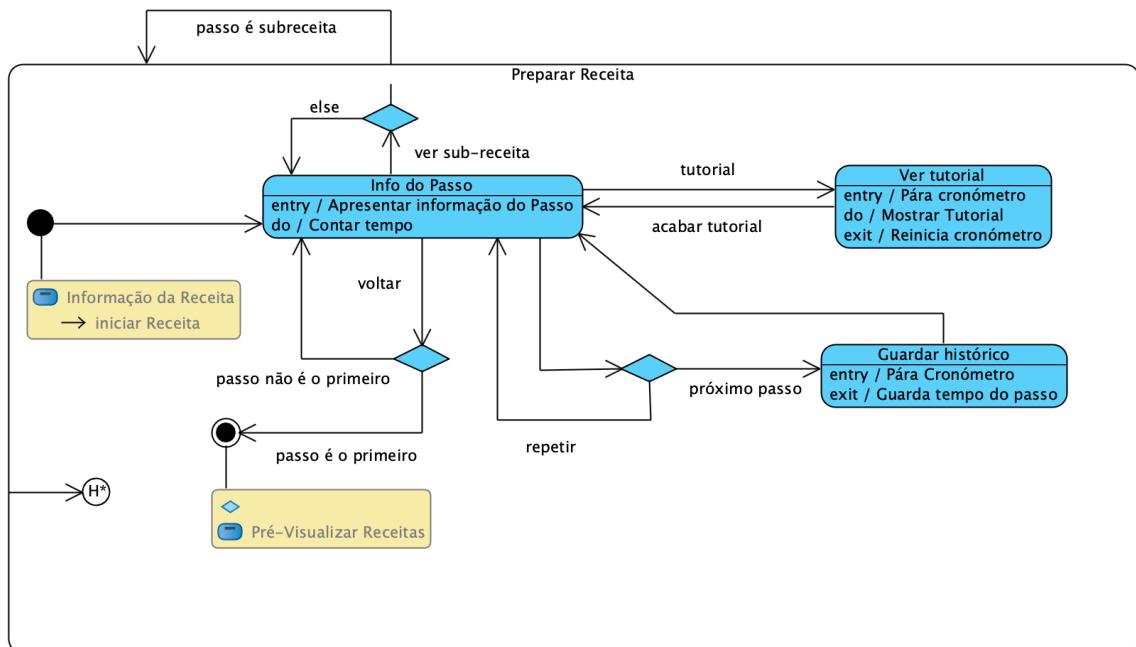


Figura 27 - Sub-Máquina de Estado 'Preparar Receita'

Sendo esta uma funcionalidade que o cliente pode executar, e muito provavelmente a função mais importante deste produto de software decidimos modelar com rigor os estados pelo que o sistema passa aquando da execução desta tarefa.

Recorremos ao uso dum histórico pois como é possível notar, um passo pode ser uma receita contida dentro de outra pelo é necessário manter um histórico para que o estado da preparação da receita principal não seja perdida.

3.4.3 Adicionar Receita

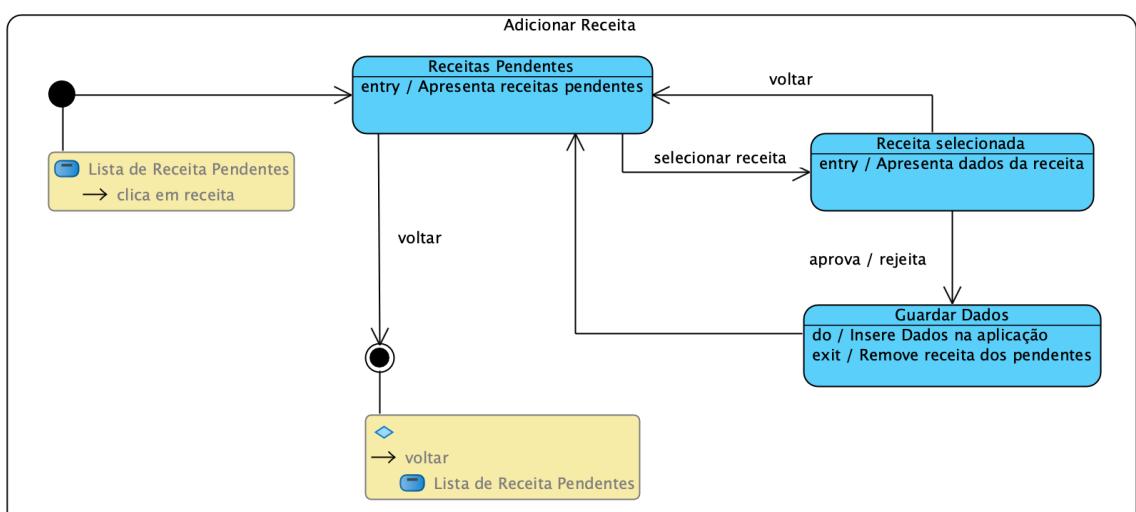


Figura 28 - Sub-Máquina de Estado 'Adicionar Receita'

Esta é uma funcionalidade apenas acedida pelos Administradores do sistema. Uma vez que cada utilizador pode adicionar uma receita nova ou uma variação duma receita já existente, para que estas não sejam inseridas de forma *ad hoc* é necessária uma aprovação dum Administrador.

Cada Receita que é adicionada vai parar a uma lista de Receitas pendentes para que seja posteriormente aprovado, ou rejeitada. A tarefa de aprovação ou rejeição é descrita no diagrama acima apresentado.

3.4.4 Registo

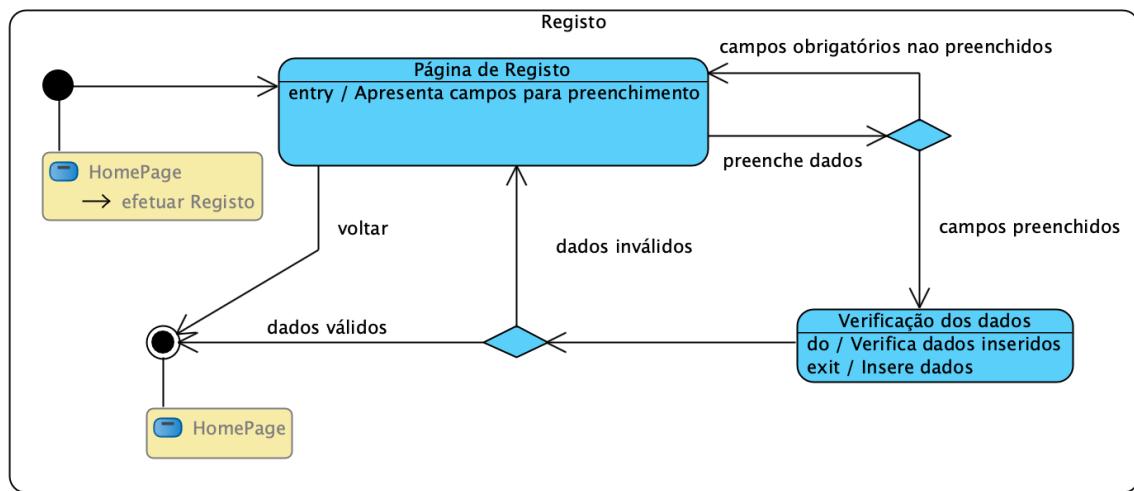


Figura 29 - Sub-Máquina de Estado 'Registo'

Esta é uma função que pode ser realizada por qualquer pessoa que se deseja tornar-se cliente da aplicação. Apesar de não ser um diagrama muito extenso decidimos apresentar a sua modelação para que seja tida como exemplo para casos de modelação semelhantes que surgiram neste projeto.

4. Base de Dados

4.1. Modelo Conceptual

Este subcapítulo serve para apresentar a base do nosso Modelo Lógico que acaba por ser o Modelo Conceptual idealizado pelo grupo. Vamos descrever as entidades envolvidas e os relacionamentos entre elas. Vamos também identificar os atributos das diferentes entidade e apresentar as suas características.

4.1.1 Identificação e caracterização das entidades

As entidades devem apenas identificar os principais objetos, ou seja, os que são considerados mais interessantes.

Tendo isto em conta foram definidas as seguintes entidades: Receita, Passo, Ingrediente, Utilizador, Categoria e Ação.

Caracterização:

- A entidade Receita identifica, em geral, uma Receita.
- A entidade Passo identifica, em geral, um Passo que está presente numa determinada receita.
- A entidade Ingrediente identifica, em geral, um Ingrediente que faz parte de um determinado passo.
- A entidade Utilizador identifica, em geral, um Utilizador que está registado no sistema.
- A entidade Categoria identifica, em geral, a categoria de uma determinada receita.
- A entidade Ação identifica, em geral, a ação realizada num determinado passo.

4.1.2 Identificação e caracterização dos relacionamentos

Para identificarmos relacionamentos é necessário encontrar os relacionamentos em si e os verbos correspondentes.

Os relacionamentos existentes são:

- Listar os passos **presentes** numa receita em específico.
- **Histórico** de um determinado utilizador relativamente às receitas e passos feitos.
- Listar os passos que **são** uma receita.
- Listar os passos que **contêm** um determinado ingrediente.
- Listar os passos que **contêm** uma determinada ação.
- Listar as categorias **presentes** numa determinada receita.

4.1.3 Domínio dos atributos

Associando cada atributo à sua respetiva entidade, foi desenvolvida a tabela seguinte de forma a ilustrar os diferentes atributos.

Entidade	Atributo	Descrição	Domínio	Nulo	Derivado	Multivalorado
Receita	idReceita (Chave Primária)	Número identificativo de cada receita	Inteiro	Não	Não	Não
	Nome	Nome da receita	String com 100 carateres	Não	Não	Não
	Nutricional	Informação nutricional da receita	String com 225 carateres	Não	Não	Não
	tempoTotal	Tempo total médio que demora a fazer a receita	Time	Não	Não	Não
	Avaliação	Avaliação média da receita	Inteiro	Não	Não	Não
	Comentário	Comentário da receita	String com 225 carateres	Não	Não	Sim
	Comentário_Data	Data de um comentário	String com 32 carateres	Não	Não	Sim

	idPasso (Chave Primária)	Número identificativo de cada passo	Inteiro	Não	Não	Não
Passo	tempoParcial	Tempo que demora a realizar o passo	Time	Não	Não	Não
	Unidade	Unidade de medida de massa	String com 32 caracteres	Sim	Não	Não
	Quantidade	Quantidade do ingrediente no passo	Inteiro	Não	Não	Não
	Extra	Informação adicional do passo	String com 225 caracteres	Sim	Não	Não

	idIngrediente (Chave Primária)	Número identificativo de cada ingrediente	Inteiro	Não	Não	Não
Ingrediente	Nome	Nome do ingrediente	String com 100 caracteres	Não	Não	Não
	Valor Nutricional	Valor Nutricional do Ingrediente	Float	Não	Não	Não

	Username (Chave Primária)	Username do Utilizador	String com 32 carateres	Não	Não	Não
Utilizador	Password	Password do Utilizador	String com 16 carateres	Não	Não	Não
	Email	Email do Utilizador	String com 225 carateres	Não	Não	Não
	Nome	Nome do Utilizador	String com 225 carateres	Não	Não	Não
Categoria	idCategoria (Chave Primária)	Número identificativo de cada categoria	Inteiro	Não	Não	Não
	Descrição	Descrição da Categoria	String com 32 carateres	Não	Não	Não
Ação	idAcao (Chave Primária)	Número identificativo da Ação	Inteiro	Não	Não	Não
	Descrição	Descrição da Ação	String com 2 carateres	Não	Não	Não

Tabela 2 - Domínio dos atributos das entidades

Relacionamento	Atributo	Descrição	Domínio	Nulo	Derivado	Multivalorado
Histórico	tempoPasso	Tempo que o utilizador demorou a realizar um passo	Time	Não	Não	Não
	data	Data da realização do passo	Date	Não	Não	Não
	nrPasso	Número do passo	Inteiro	Não	Não	Não

Receita_Passo	Número	Número do passo na receita	Inteiro	Não	Não	Não
---------------	--------	----------------------------	---------	-----	-----	-----

Tabela 3 - Domínio dos atributos dos relacionamentos

4.1.4 Apresentação do modelo conceptual

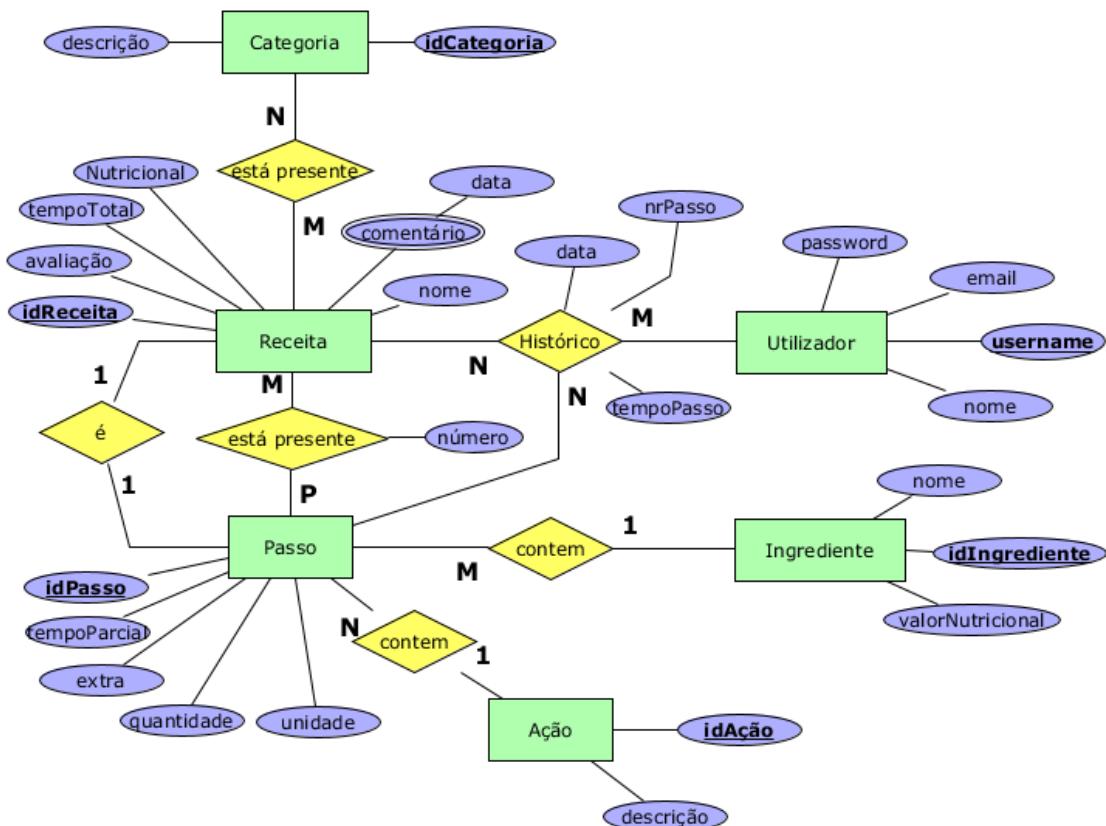


Figura 30 – Modelo Conceptual desenhado no “TerraER”

4.2. Modelo Lógico

Nesta secção apresentamos todos os passos para a construção do Modelo Lógico. Como base foi criado um modelo conceptual onde foram criadas todas as entidades, os relacionamentos entre elas e os atributos (tanto dos relacionamentos como das entidades). Após este processo, e com ajuda da ferramenta “Visual Paradigm”, desenhamos o modelo. De seguida, validamos o modelo através das regras de normalização e verificamos se este respondia a todos os requisitos referidos na **secção 2**. Por fim, verificamos a integridade do modelo e revemos este com o utilizador.

4.2.1 Construção e validação do modelo de dados lógico

Começamos por especificar os nomes das relações e os atributos desta. De seguida identificamos as chaves primárias e estrangeiras.

Após isso criamos os relacionamentos e podemos identificar três tipos diferentes no nosso modelo: **1:1**, **1:N** e **N:N**. No caso dos relacionamentos **1:1**, as relações deverão apenas estar associadas através de uma única entrada (no nosso caso existe uma entrada obrigatória dum lado e outro é opcional). Para os relacionamentos **1:N**, a chave primária referente à entidade “pai” (entidade do lado 1) deve estar presente na relação referente à entidade “filho” (entidade do lado N). Ambos não criam relações novas, ao contrário dos relacionamentos **N:N** que dão origem a uma nova relação que irá conter as chaves primárias de ambas as relações associadas e também os atributos (caso existam) do relacionamento.

A existência de um relacionamento ternário levou à criação de uma nova relação que contém as chaves estrangeiras das três relações envolvidas e os atributos (caso existam) do relacionamento.

Por último, devido à presença de vários atributos multivvalorados foram criadas novas relações para cada um destes atributos. O relacionamento entre a entidade “pai” (entidade que contém o atributo multivvalorado) e a entidade “filho” (entidade que representa o atributo multivvalorado) é do tipo **1:N**.

Também criamos as relações **Acao** e **Categoria** que servem para aumentar a eficiência e a organização dos dados. Isto acontece porque a informação referente a uma ação ou a uma categoria seria várias vezes repetida desnecessariamente se fosse um simples atributo, desta forma resolvemos o impasse através de uma forma otimizada.

De seguida apresentamos uma tabela com todas as relações bem como as suas chaves primárias, estrangeiras e atributos.

Receita (idReceita, nome, nutricional, tempoTotal, avaliação)
Chave primária: idReceita
Utilizador (username, password, email, nome)
Chave primária: username
Ingrediente (idIngrediente, valorNutricional, nome)
Chave primária: idIngrediente
Passo (idPasso, tempoParcial, unidade, quantidade, extra, subReceita, ingrediente, acao)
Chave primária: idPasso
Chave estrangeira: subReceita referencia Receita (idReceita)
ingrediente referencia Ingrediente (idIngrediente)
acao referencia Acao (idAcao)
Categoria (idCategoria, descricao)
Chave primária: idCategoria
Cat_Receita (categoria, receita)
Chave primária: categoria, receita
Chave estrangeira: receita referencia Receita (idReceita)
categoria referencia Categoria (idCategoria)
Comentario (idComentario, descricao, data, receita)
Chave primária: idComentario
Chave estrangeira: receita referencia Receita (idReceita)
Receita_Passo (receita, passo, numero)
Chave primária: passo, receita
Chave estrangeira: receita referencia Receita (idReceita)
passo referencia Passo (idPasso)
Tutoriais (link,passo)
Chave estrangeira: passo referencia Passo (idPasso)
Acao (idAcao, descricao)
Chave primária: idAcao
Historico (receita, utilizador, passo, tempoPasso, data, nrPasso)
Chave primária: passo, receita, utilizador
Chave estrangeira: receita referencia Receita (idReceita)
passo referencia Passo (idPasso)
utilizador referencia Utilizador (username)

Tabela 4 - Relações do modelo Lógico

4.2.2 Desenho do modelo lógico

Depois de obtermos as relações e os respetivos relacionamentos, criamos o modelo lógico. Para isso, utilizamos o *Visual Paradigm* para desenhar o modelo. Eis o resultado:

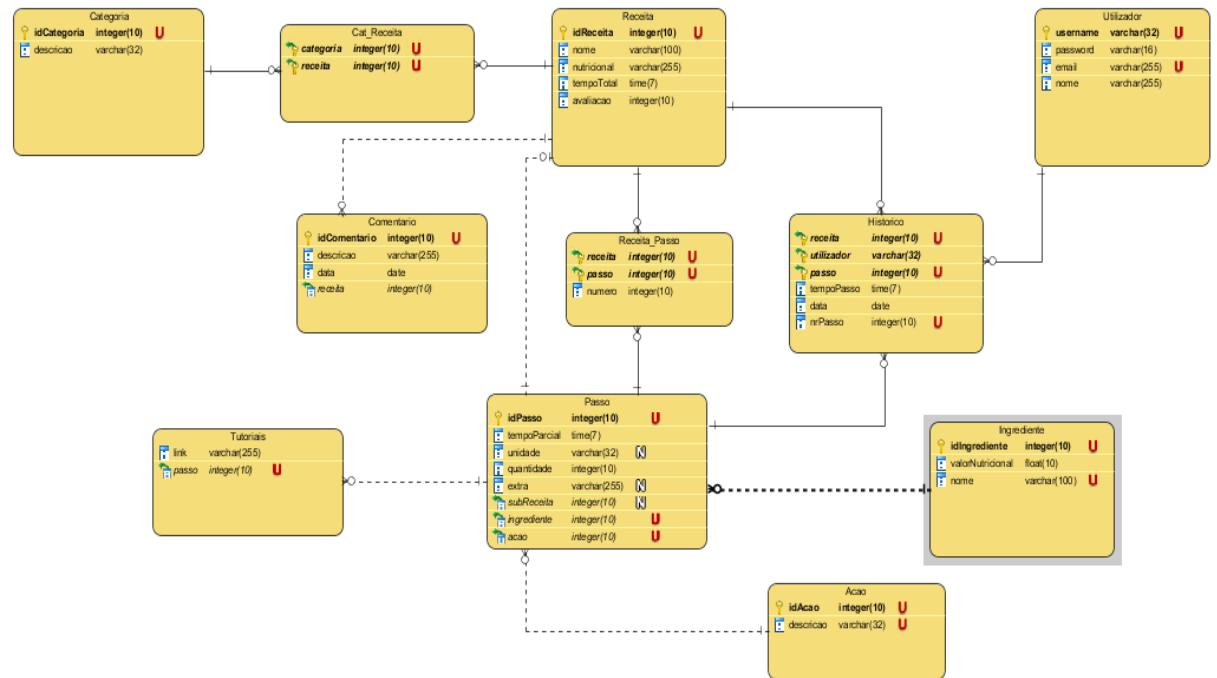


Figura 31 – Modelo Lógico desenhado no “Visual Paradigm”

4.2.3 Normalização e Integridade

Confirmamos que o nosso modelo cumpre pelo menos as 3 primeiras formas normais. Visto que todos os nossos atributos são atómicos, todos os atributos não chave dependem apenas da sua respetiva chave e não existem atributos não chave a depender de outros atributos não chave, isto verifica-se em todas as tabelas.

4.2.4 Verificação da Integridade das Restrições

Nesta subsecção apresentamos as restrições que desejamos que se cumpram de forma a proteger o nosso sistema de base de dados. Deste modo visamos que problemas como a inconsistência, imprecisão e dados incompletos sejam extinguidos. O objetivo desta análise de integridade de restrições baseia-se na necessidade de oferecer à empresa uma representação verdadeira dos seus requisitos de dados.

4.2.5 Dados Necessários

Existem no nosso modelo lógico 3 atributos que podem ser NULL, encontrando-se os três atributos na relação Passo. No caso dos atributos da relação Encomenda (subReceita,unidade,extra), estes podem ser nulos uma vez que o passo pode não ser uma sub-receita, o passo pode não ter unidade e não conter nenhum componente extra.

Todos os outros atributos que não são referidos não podem ser nulos. Desta forma no nosso modelo lógico todos estão assinalados com a opção NOT NULL.

4.2.6 Domínio das Restrições dos Atributos

O domínio de cada atributo foi escolhido especificamente para as necessidades de cada um, contudo as decisões que foram tomadas em relação a estes domínios foram já descritas na secção 4.1.3.

Para valores inteiros, os atributos da nossa base de dados possuem o tipo de dados INT, para atributos como o valorNutricional usamos o tipo de dados FLOAT.

Para atributos cujos valores são Strings, optamos por usar VARCHAR com o tamanho a variar consoante cada atributo.

Para os atributos que registam o tempo de confecção das receitas optamos por utilizar o tipo de dados Time.

Por último, para atributos que se baseiam em datas optamos por usar o tipo de dados DATE.

4.2.7 Cardinalidade

As restrições associadas ao relacionamento entre dados do nosso sistema de base de dados foram já introduzidas na secção 4.1.2.

O nosso modelo tem 2 relacionamentos N:M. Uma Receita pode ter várias categorias e uma receita pode ter vários passos, o que originou uma tabela para cada um destes relacionamentos.

Temos 4 relacionamentos 1:N, estes não originam novas tabelas apenas exigem que a chave primária da relação pai esteja presente na relação filho, como sendo chave estrangeira.

Temos também um relacionamento ternário "Histórico" que faz ligação entre o Utilizador na execução de uma Receita e dos seus respetivos passos.

Por fim, existe também um relacionamento 1:1 que faz ligação de Receitas para Passos, que existe devido a um passo ser uma sub-receita.

4.2.8 Integridade das Entidades

No nosso modelo lógico apresentado nenhuma relação pode ter uma chave primaria cujo valor seja nulo.

Assinalamos em todas as chaves primárias a opção NOT NULL para que esta restrição fosse cumprida.

4.2.9 Integridade Referencial

Como já foi abordado na secção 4.2.7 existem 4 relacionamentos tipo 1:N, e por isso é necessário ter atenção para que a integridade referencial não seja violada quando se alteram dados que fazem referência a outras tabelas, ou são referenciados por outras tabelas. De um modo mais simples é necessário ter cuidado quando se faz alguma operação que possa envolver referências, por exemplo através de chaves estrangeiras a outras tabelas.

Na operação de inserção de alguma entrada nas relações: Cat_Receita, Receita_Passo, Historico, é preciso ter atenção uma vez que estas são relações do tipo filho, porque não se deve inserir valores nestas tabelas se a referência à relação pai for nula. Apenas se deve proceder à inserção de valores quando existir uma relação pai válida.

Na operarão de remoção de entradas de relações do tipo pai, é necessário ter certos cuidados uma vez que as relações do tipo filho possuem referências para estas tabelas.

Remover uma entrada de uma relação do tipo pai violaria a integridade referencial uma vez que a chave estrangeira presente na relação filho passaria a ter um valor que representaria uma tabela que não existe. Estes casos podem enão ser tratados através de uma ação conhecida como CASCADE, que dita que sempre que haja uma remoção de uma entrada de uma relação pai, todas as entradas de relações do tipo filho são também removidas mantendo a integridade referencial.

No caso do nosso modelo lógico se removêssemos um Utilizador teríamos de eliminar todos os históricos desse utilizador.

4.2.10 Restrições gerais

Existem certos atributos das entidades, para além das chaves primárias, que conseguem identificar a entidade como sendo única. Esses atributos merecem um destaque especial uma vez que podem ser marcados com a opção UNIQUE.

Exemplos deste caso são o atributo username e email do Utilizador, uma vez que estas designações têm de ser únicas a cada Utilizador pois não seria coerente dois utilizadores terem o mesmo username ou o mesmo email.

Uma vez que existem atributos que são do tipo INT e FLOAT, é necessário torná-los positivos, pois no nosso sistema não existem atributos do tipo numérico que possam ser negativos, seria incoerente se tal situação ocorresse. Para marcar este facto, neste tipo de atributos ativamos a opção UNSIGNED.

Alguns atributos, discriminando por exemplo o Id de um Comentario, o id de um Ingrediente, o id de uma receita, o Id de uma categoria e o Id do Passo, não são inseridos diretamente por quem está a inserir dados deste género no sistema.

Não seria nada cômodo ter que saber qual é o id que está livre a seguir para um Comentario ou para uma Receita, logo a solução para estes casos é acionar a opção AUTO INCREMENT, por forma a que na inserção deste tipo de dados, estes atributos sejam gerados automaticamente.

4.2.11 Revisão do modelo Lógico com o Utilizador

Os utilizadores foram requeridos para rever o modelo lógico, de modo a confirmar se este representa fiavelmente os dados necessários para o bom funcionamento da ideia em causa.

Este modelo permite a filtragem de receitas (realizada pelos utilizadores), bem como o seu registo e manutenção (realizado pelos administradores).

Também é possível realizar receitas passo a passo, selecionar receitas e avaliar/comentar as receitas após a sua realização.

Por fim, é concebível guardar informações relativas ao utilizador tais como estatísticas temporais e dificuldades na preparação de determinadas receitas.

Concluímos, portanto, que o modelo apresentado cumpre os requisitos impostos pelos utilizadores.

5. Prototipagem da Interface

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

Inicialmente apresentamos a página inicial que disponibiliza ao utilizador a operação de registo e de login.

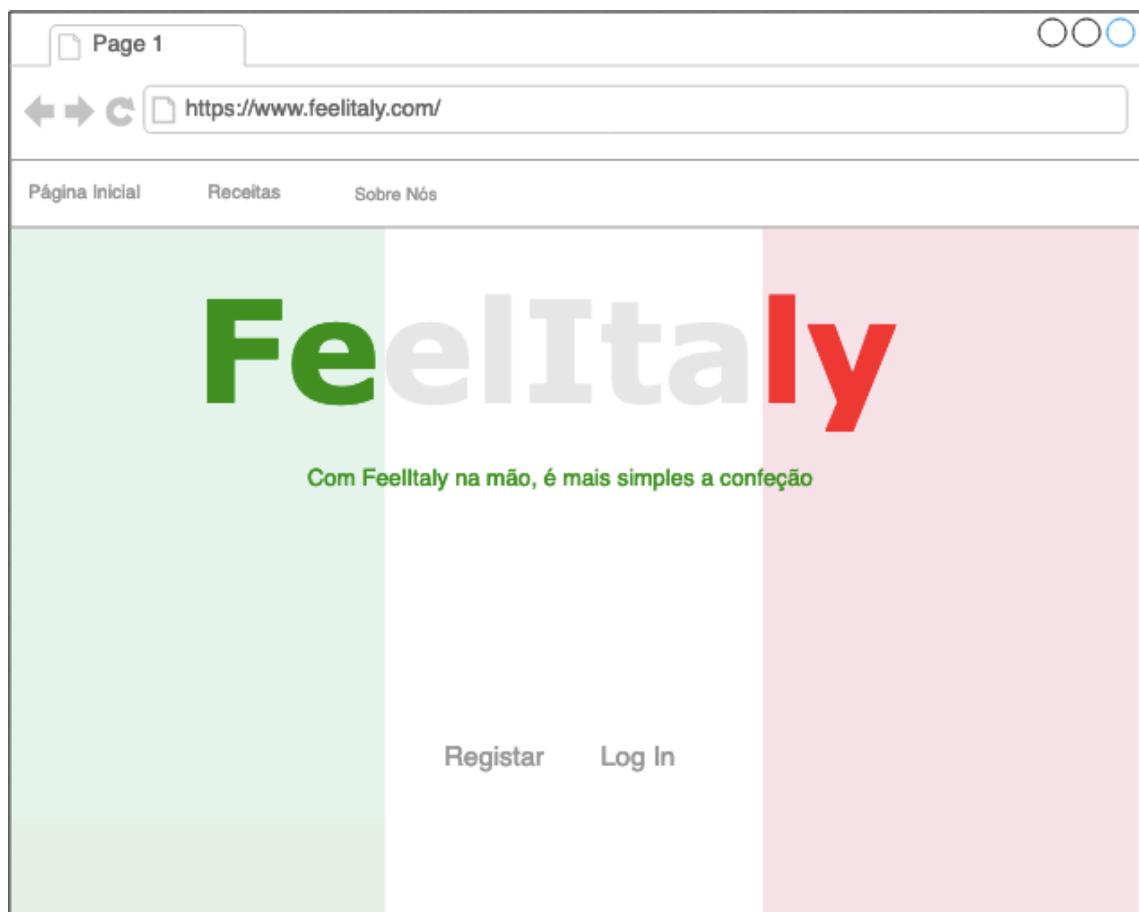


Figura 30 – Mockup Página Inicial

As imagens seguintes mostram os *mockups* que permitem a um utilizador proceder ao registo ou então ao login, caso este já se encontre registado.

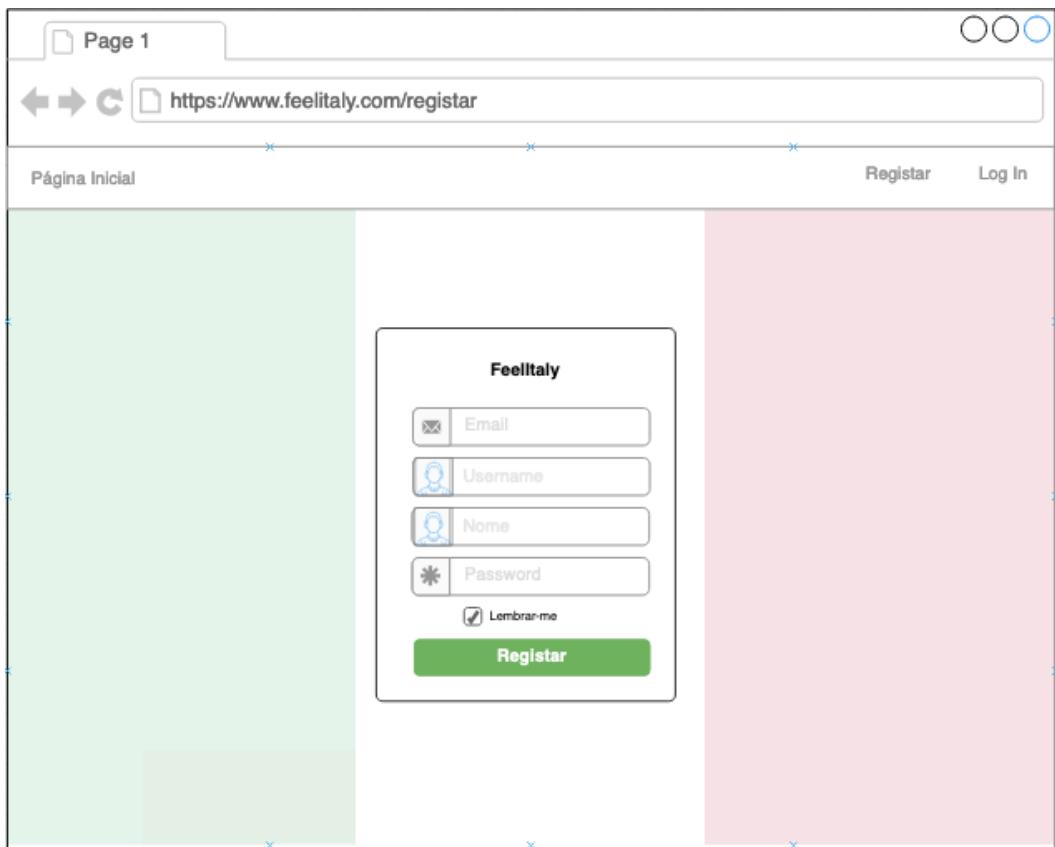
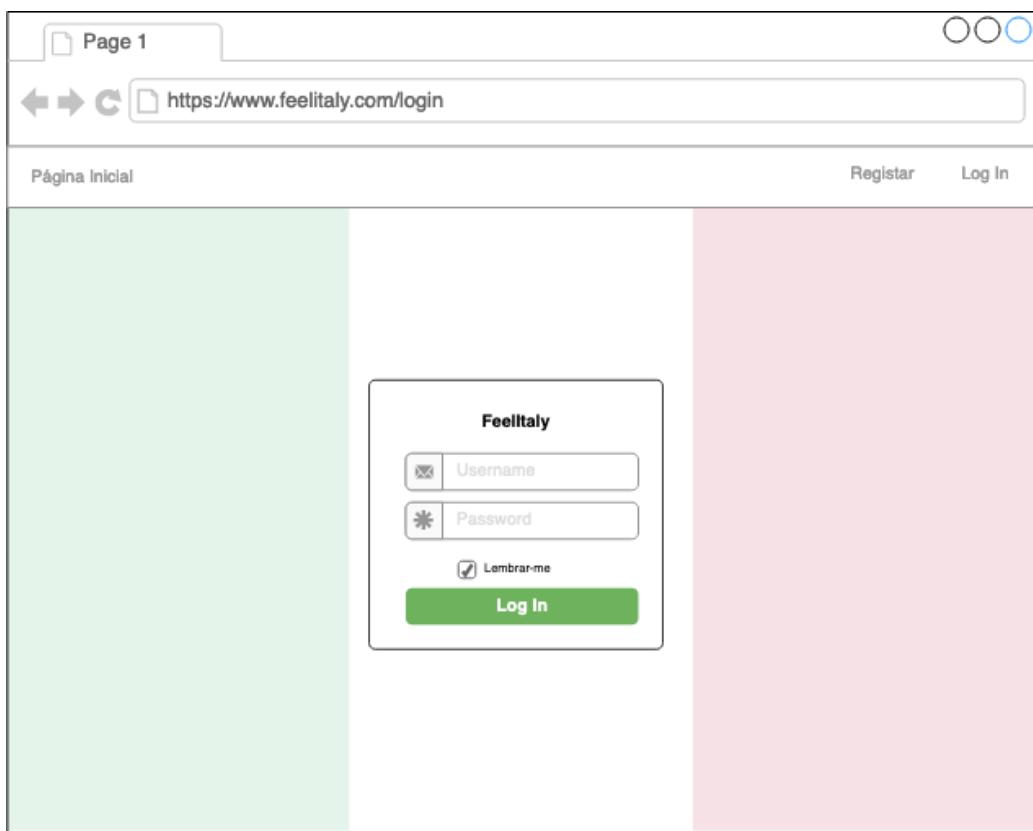


Figura 31 - Mockup Registrar



De seguida, é apresentada a interface proporcionada ao utilizador para escolher uma receita. Nesta são listadas as receitas presentes no site, e também quatro opções para filtrar a sua escolha, respetivamente, tipo de prato, ingredientes, tipo de refeição, e tempo de confeção.

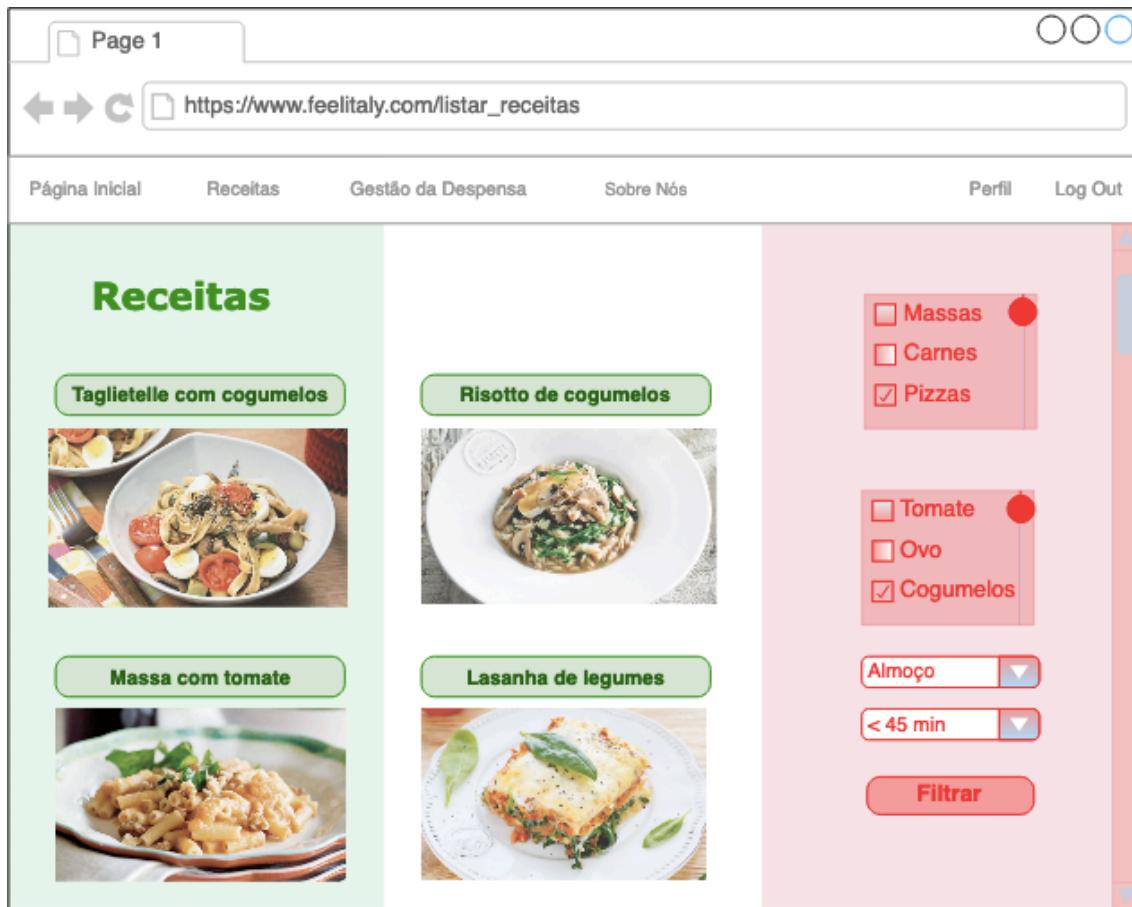


Figura 33 - Mockup Listar Receitas

Tendo em conta a escolha do utilizador, será apresentada uma das receitas presentes no nosso site. Na sua apresentação será providenciada a imagem relativa à receita em ponto maior, os ingredientes necessários para a sua confeção, a energia calórica total, e também a avaliação e os comentários relativos à receita.

O utilizador tem depois a opção de preparar a receita, caso clique no botão iniciar.

Na seguinte figura, é tido como exemplo o Risotto de Cogumelos.

The mockup shows a web browser window with the URL https://www.feelitaly.com/risotto_de_cogumelos. The page header includes 'Page 1', a search bar, and navigation icons. The main content area has a green header 'Risotto de Cogumelos' and a large image of the dish. Below the image, there's a section for comments and a green 'Iniciar' button. To the right, there's a pink sidebar with sections for ingredients, energy content, and an evaluation rating.

Risotto de Cogumelos

Comentários:
joaop21: Excelente!! Melhor risotto que já comi!
richard : Não gostei muito!

Iniciar

Ingredientes

- Cebola
- Cogumelos
- Azeite
- Arroz
- Vinho Branco
- Água
- Sal
- Queijo brie
- Rúcula
- Pimenta

Energia
323 Kcal

Avaliação
★★★★★

Figura 34 - Mockup Selecionar Receita

Ao clicar no botão Iniciar, o utilizador vai então começar o processo de preparar uma receita. A preparação duma receita é realizada com um só passo de cada vez no ecrã da aplicação. É também providenciado o ingrediente utilizado no passo e também um tutorial em modo link, que ajuda na realização do passo.

O utilizador pode, após realização do passo, prosseguir para o próximo passo clicando nesse botão, pode repetir o passo caso não ache que o tenha realizado corretamente, e pode também voltar ao passo anterior, com a intenção de o repetir.



Figura 35 - Mockup Preparar Receita

6. Conclusão e Trabalho Futuro

Durante a realização desta fase do trabalho fomos várias vezes confrontados com dificuldades relativamente a decidir qual a melhor forma de abordar os diferentes diagramas e modelos, especialmente os diagramas de classe e o modelo lógico. Relativamente ao restante o grupo trabalhou de uma forma coerente e fluída.

Em termos de trabalho futuro, iremos rever alguns detalhes das fases já realizadas mas, resumidamente, o grupo tem todas as bases necessárias para iniciar o processo de implementação da melhor forma possível.

Em suma, o grupo está satisfeito com todo o processo de especificação tendo sido bastante enriquecedor ouvir todos os pontos de vista existentes.

7. Bibliografia

- Ian Sommerville, Software Engineering, Ninth Edition.