



Universidade do Minho  
Escola de Engenharia  
Mestrado Integrado em Engenharia Informática

## Unidade Curricular de Laboratórios de Informática IV

Ano Letivo de 2018/2019

### CookieFy, Assistente de Culinária

#### **Grupo 14:**

- Flávio Martins
- Gonçalo Costeira
- José Ramos
- Rafael Silva
- Rui Costa

Junho, 2019

L|4 18-19

Data de Recepção	
Responsável	
Avaliação	
Observações	

## **Resumo**

Desenvolvimento de uma rede social para culinária com o intuito de permitir a partilha e a facilidade de acessos a novas receitas de pratos principais por parte do utilizador comum, ouvindo ao mesmo tempo a sua música favorita. Existirá ainda a funcionalidade de planeamento das suas refeições semanais, bem como a gestão e procura dos ingredientes necessários aos cozinhados já planeados. Por fim, mas não menos importante, os utilizadores poderão ser acompanhados por um assistente pessoal de cozinha, cuja função é ajudar/orientar o utilizador durante a confeção da receita, através de controlo de voz.

**Área de Aplicação:** Desenvolvimento de Programa de culinária com vertente de Rede Social e partilha de receitas, com relação ao mundo da música.

**Palavras-Chave:** Bases de Dados Relacionais, C#, Protocolos de Comunicação, WebBrowser, Multiplataformas, Redes Sociais.

# Índice

Resumo	3
Índice	4
Índice de Figuras	<b>Erro! Marcador não definido.</b>
1. Introdução	10
1.1 Contextualização	10
1.2 Motivação e Objetivos	11
2. Definição da identidade do sistema a desenvolver	12
3. Justificação, viabilidade e utilidade do sistema	13
4. Identificação dos Recursos Necessários	14
4.1 Recursos de Rede	14
4.2 Recursos Humanos	15
4.3 Recursos de Hardware	15
4.4 Recursos de Software	16
4.5 Recursos de Dados	16
5. Medidas de sucesso	17
6. Maqueta do sistema	18
<i>Figura 1 - Maquete do Sistema</i>	18
7. Plano de Desenvolvimento do Projeto	19
7.1 Diagrama de Gantt	19
<i>Figura 2 - Diagrama de Gantt</i>	19
7.2 Modelo em Cascata	20
<i>Figura 3 - Modelo Cascata</i>	20
8. Requisitos	21
Requisitos Não Funcionais	21
Requisitos Funcionais	22
9. Especificação do Software	25
<i>Figura 4 - Diagrama Usado como Guia no Desenvolvimento do Projeto</i>	26
9.1. Modelo de Domínio	27
<i>Figura 5 - Modelo de Domínio</i>	27
9.2. Diagramas de Use Case	28

<i>Figura 6 - Uses Cases</i>	28
<i>    Editar Perfil</i>	29
<i>    Figura 7 - Use Case Editar Perfil</i>	29
<i>    Confeccionar Receita</i>	30
<i>    Figura 8 - Use Case Confeccionar Receita</i>	30
<i>    Figura 9 - Use Case Gerir Música</i>	30
<i>    Consultar Receita</i>	31
<i>    Figura 10 - Use Case Consultar Receita</i>	31
<i>    Figura 11 - Use Case Juntar a Lista</i>	31
9.2.1.     Especificação de Use Cases	32
<i>    Gestão</i>	32
<i>    Figura 13 - Use Case Alterar Descrição</i>	33
<i>    Figura 14 - Use Case Alterar Nome</i>	33
<i>    Figura 15 - Use Case Alterar Password</i>	34
<i>    Figura 16 - Use Case Alterar Profissão</i>	34
<i>    Figura 17 - Use Case Consultar Perfil</i>	35
<i>    Figura 18 - Use Case Criar Conta</i>	35
<i>    Figura 19 - Use Case Login</i>	36
<i>    Figura 20 - Use Case Logout</i>	36
<i>    Figura 21 - Use Case Pesquisar Receitas</i>	37
<i>    Figura 22 - Use Case Pesquisar Users</i>	37
<i>    Figura 23 - Use Case Definir Plano de Receitas Semanal</i>	38
<i>    Figura 24 - Use Case Subscrever a Premium</i>	38
<i>    Figura 25 - Use Case Alterar Morada</i>	39
<i>    <i>Receita</i></i>	39
<i>    Figura 26 - Use Case Adicionar Receita à lista de receitas do User</i>	39
<i>    Figura 27 - Use Case Adicionar Receita à lista de receitas já existente</i>	40
<i>    Figura 28 - Use Case Adicionar Receita</i>	40
<i>    Figura 29 - Use Case Apagar Receitas</i>	41
<i>    Figura 30 - Use Case Avaliar Receitas</i>	41
<i>    Figura 31 - Use Case Comentar Receita</i>	42
<i>    Figura 32 - Use Case Comprar receita premium</i>	42
<i>    Figura 33 - Use Case Consultar Receita</i>	43

Figura 35 - Use Case Editar Receita	44
Figura 37 - Use Case Monetizar Receita	45
<i>Gerir Música</i>	45
Figura 39 - Use Case Começar Música	46
Figura 40 - Use Case Gerir Volume	46
Figura 41 - Use Case Música Anterior	47
<i>Confeção</i>	48
Figura 43 - Use Case Solicitar Ingredientes Alternativos	48
Figura 44 - Use Case Iniciar Temporizador	48
Figura 45 - Use Case Próximo Passo	49
Figura 46 - Use Case Repetir Passo Atual	49
Figura 47 - Use Case Repetir Passo Anterior	50
Figura 48 - Use Case Ver Tutorial	50
9.3.     Diagrama de Sequência	51
Gestão	51
Figura 49 - Diagrama de Sequência Adicionar CooockieCoins	51
Figura 50 - Diagrama de Sequência Alterar Descrição	52
Figura 51 - Diagrama de Sequência Alterar Foto	52
Figura 52 - Diagrama de Sequência Alterar Morada	53
Figura 53 - Diagrama de Sequência Alterar Nome	53
Figura 54 - Diagrama de Sequência Alterar Password	54
Figura 55 - Diagrama de Sequência Alterar Profissão	54
Figura 56 - Diagrama de Sequência Editar Perfil	55
Figura 57 - Diagrama de Sequência Login	55
Figura 58 - Diagrama de Sequência Logout	56
Figura 59 - Diagrama de Sequência Consultar Perfil	56
Figura 60 - Diagrama de Sequência Criar Conta	57
Figura 61 - Diagrama de Sequência Pesquisa User	57
Figura 62 - Diagrama de Sequência Subscrever Premium	58
Receita	58
Figura 63 - Diagrama de Sequência Adicionar Receita	58
Figura 64 - Diagrama de Sequência Adicionar Receita à Lista	59
Figura 65 - Diagrama de Sequência Receita	59

Figura 66 - Diagrama de Sequência Avaliar Receita	60
Figura 67 - Diagrama de Sequência Comentar Receita	60
Figura 68 - Diagrama de Sequência Consultar Receita Implementação	61
Figura 69 - Diagrama de Sequência Criar Lista Receitas	61
Figura 70 - Diagrama de Sequência Comprar Receita Premium	62
Figura 71 - Diagrama de Sequência Definir Plano de Receitas Semanal	62
Figura 72 - Diagrama de Sequência Editar Receita	63
Figura 73 - Diagrama de Sequência Eliminar Receita	63
Figura 74 - Diagrama de Sequência Juntar Receita à Lista	64
Gerir Música	64
Figura 75 - Diagrama de Sequência Avançar Música	64
Figura 76 - Diagrama de Sequência Começar Música	65
Figura 77 - Diagrama de Sequência Gerir Volume da Música	65
Figura 78 - Diagrama de Sequência Música Anterior	66
Figura 79 - Diagrama de Sequência Para Música	66
Confeção	67
Figura 80 - Diagrama de Sequência Ingredientes Alternativos	67
Figura 81 - Diagrama de Sequência Iniciar Temporizador	67
Figura 82 - Diagrama de Sequência Repetir Passo Atual	68
Figura 83 - Diagrama de Sequência Próximo Passo	68
Figura 84 - Diagrama de Sequência Reproduzir Tutorial	69
Figura 85 - Diagrama de Sequência Repetir Passo Anterior	69
9.4. Diagrama de Classes	70
9.4.1. Diagrama de Classes V1	70
Figura 86 - Diagrama de Classes V1.0	70
9.4.2. Diagrama de Classes V2	71
Figura 87 - Diagrama de Classes V2.0	71
10. Base de dados	72
10.1. Modelo Conceptual	72
10.1.1. Identificar os tipos de identidades	72
Figura 88 - Tabela - Tipo de Entidades	72
10.1.2. Identificar tipos de relacionamento	73
Figura 89 - Tabela - Tipo de Relacionamento	73

10.1.3.	Identificar e associar atributos com os tipos de entidades e relacionamentos	74
	Figura 90 - Tabela - Entidade/Atributos/Relacionamentos	75
10.1.4.	Atributos simples/compostos	76
10.1.5.	Atributos derivados	76
10.1.6.	Atributos multivalor	76
10.1.7.	Associação entre atributos e entidades	76
10.1.8.	Determinar domínio dos atributos	76
10.1.9.	Determinar chaves primárias, candidatas e alternativas	78
	Figura 91 - Modelo Conceptual	79
10.1.11.	Revisão do modelo de dados com o cliente	79
10.2.1.	Construção e validação do modelo de dados lógicos	80
10.2.1.1.	Entidades Fortes	80
	Figura 94- Conceptual para lógico (Entidade RecipeList)	82
	Figura 95 - Conceptual para lógico (Entidade Subscription)	82
	Figura 96 - Conceptual para lógico (Entidade Ingredient)	82
	Figura 98 - Conceptual para lógico (Entidade Tag)	83
	Figura 99 - Conceptual para lógico (Entidade Utensils)	83
10.2.1.2.	Relacionamentos binários 1:N	84
	Figura 100 - Exemplo de relação 1 : N	84
10.2.1.3.	Relacionamentos binários N: N	85
	Figura 101 - Exemplo de relação N : N	85
10.2.1.4.	Auto-Relacionamentos binários	86
	Figura 102 - Exemplo de Auto-Relação	86
10.2.2.	Desenho do Modelo Lógico	87
	Figura 103 - Modelo Lógico	87
10.2.3.	Validação do modelo através da normalização	88
10.2.4.	Análise do crescimento futuro	89
10.2.5.	Revisão do modelo lógico com o utilizador	89
11.	MockUps	90
	HomePage	90
	Figura 104 - MockUp HomePage	90
	Registrar	91
	Figura 105 - MockUp Register	91

Login	92
Figura 106 - MockUp Login	92
Procurar	93
Figura 107 - MockUp Procurar	94
Consultar Receita	95
Figura 108 - MockUp Consultar Receita	95
Dashboard	96
Figura 109 - MockUp DashBoard	96
Assistente CookieFy	97
Figura 110 - MockUp Assistente CookieFy	97
Visualizar Perfil	98
Figura 111 - MockUp Visualizar Perfil	98
12. Implementação	99
15. Conclusões e Trabalho Futuro	114

# 1. Introdução

## 1.1 Contextualização

Hoje ainda existe muita gente que procura aquela receita para um momento especial ou simplesmente um jantar em família ou entre amigos. Além disso, a maioria da população quando está a cozinhar gosta de ouvir música para acompanhar no seu cozinhado.

Assim, neste âmbito, a nossa empresa de Desenvolvimento de Software foi contactada pela Spotify para desenvolvermos uma aplicação capaz de juntar esses dois mundos. Pois a Spotify que já se encontra em actividade na indústria da música há alguns anos, verificou que grande parte dos seus utilizadores acedem a aplicação para ouvir música, na hora de confeção da sua refeição, tendo isto em conta, eles pensaram que proveito podiam tirar dessa situação, após um estudo detalhado da sua equipa de Marketing/Gestão, chegaram a uma conclusão que podiam juntar o mundo da música com o mundo da culinária, uma situação que nunca antes foi realizada, ou pensada.

Para tornar isso possível a Spotify contactou a nossa Empresa de Software com o intuito de tornar essa ideia possível, a ideia resumidamente consiste em criarmos uma interface de culinária que permite ao Utilizador criar e escolher receitas, e durante a sua confeção ouvir a sua música e ao mesmo tempo ser guiado pelo assistente denotado “CookieFy” numa série de passos que compõem uma receita, trazendo assim o Spotify ao mundo da culinária.

## **1.2 Motivação e Objetivos**

Após a proposta que recebemos sobre o desenvolvimento de uma aplicação para o apoio de confeção de receitas de culinária bem como o pedido de um assistente de cozinha que iriam mais tarde ser implementados como uma novas funcionalidades para o Spotify a nossa equipa iniciou um debate sobre os benefícios e riscos presentes em aceitar tal proposta e caso aceitasse-mos, quais seriam os objetivos que iríamos definir.

Assim após revermos esta oportunidade fantástica de participar num projeto com grande inovação num mercado ainda pouco explorado, que iria desafiar as nossas competências na área de programação e desenvolvimento de software, e que apresentará grande utilização visto tratar-se de um add-on ao Spotify chegarmos a decisão de ser um a proposta do nosso interesse.

No âmbito de criar um primeiro rascunho mental do projeto argumentamos e deparamos nos com os seguintes objetivos que pretendemos alcançar:

- Criação de uma aplicação acessível a todos os utilizadores e que seja de fácil utilização.
- Aplicação será capaz de acompanhar o processo de cozinha da receita, de criação de uma nova, bem como terá a presença de um assistente de cozinha que comunicará por controlos voz com o utilizador.
- Desenvolvimento de um modelo de aplicação que permita monetização de receitas inseridas por utilizadores e o aparecimento de Influencers.
- Integração total no Spotify que permita a reprodução de música enquanto o assistente de voz não é necessário e o utilizador se encontra a realizar tarefas culinárias.
- Integração de várias funcionalidades que permitam o controle de tempo, temperatura e de vários checkpoints que possam existir nas receitas.

## **2. Definição da identidade do sistema a desenvolver**

**Nome:** *CookieFy*

**Categoria:** Alimentação

Descrição: A aplicação permitirá ao utilizador registar-se na mesma e, após o registo, disponibilizará toda a informação sobre os serviços disponíveis para usufruir. Além disso, será possível realizar uma procura de receitas, bem como a compras de algumas especiais. Quando o utilizador tem acesso a uma receita pode cozinhá-la e para tal terá acesso a um assistente de cozinha pelo qual comunicará por comandos de voz. Para além do assistente também terá acesso a funcionalidades que ajudaram a controlar o processo (exemplo: temporizador, vídeos de procedimentos especiais, etc), bem como conexão a sua conta Spotify onde a aplicação se integra podendo aceder às suas playlists e reproduzir música.

### **Características:**

- Reconhecimento de texto e voz;
- Assistente de Cozinha;
- Criação de Receitas
- Avaliações de receitas;
- Opinião de utilizadores;
- Partilha em redes sociais;
- Conexão ao Spotify para reprodução de músicas.

### **3. Justificação, viabilidade e utilidade do sistema**

**Justificação:** A Spotify recorreu a dados estatísticos recolhidos pela sua aplicação e deparou-se com a informação que grande parte dos seus utilizadores acedem ao Spotify para ouvir música, na hora de confeção das suas refeições.

Assim chegaram à conclusão que poderia abrir uma nova oportunidade de negócio e após um estudo da sua equipa de Marketing/Gestão, chegaram decidiram podiam juntar o mundo da música com o mundo da culinária, numa nova aplicação que procuraria chamar novos utilizadores, bem como fornecer novas funcionalidades a3os que já utilizam o Spotify, e também gerar um novo âmbito de revenue.

**Viabilidade:** A já existente comunidade de pessoas que utiliza Spotify no seu dia a dia, as utilizadores que através dos dados apresentados pela aplicação ouvem música em hora de cozinhar as refeições e todas as pessoas que procuram uma interface que lhes permita e facilita aprimorar as suas qualidades culinárias, são os fatores que levaram o Spotify ao planeamento e presente execução desta proposta visto ter muitas caminhos de viabilidade e prometer ser algo novo num negócio ainda pouco explorado.

**Utilidade:** Qualquer pessoa com acesso à internet terá nas suas mãos a possibilidade de entrar no mundo da cozinha em conjunto com a comunidade do Spotify, e através desta aplicação poderá aprender/aprimorar a cozinhar, terá apoio de um assistente pessoal durante o processo de confeção da receita, e será capaz de adquirir receitas especiais, bem como criar as próprias receitas. Assim trata-se de uma aplicação para uso no dia a dia que estará ao acesso de toda a gente.

## **4. Identificação dos Recursos Necessários**

A nossa aplicação será desenvolvida em volta do ecossistema Microsoft(eg. visual Studio, SQL server, ...), tendo como objetivo criar um assistente de cozinha para utilizar em casa.

É direcionado para o uso por utilizadores não especializados em cozinha nas suas próprias casas, tendo que ter uma conexão à internet e, consoante os seus objetivos, aderir a um plano de subscrição adequado. Para que a utilização seja prática e intuitiva este sistema é controlado por voz de maneira a evitar inconvenientes que se tornaram incontornáveis sem esta interface.

### **4.1 Recursos de Rede**

Visto que o nosso projeto é web based, a conexão à internet é fundamental sendo que a conexão é obrigatória para que o utilizador possa efetuar qualquer operação com o sistema, como por exemplo consultar ou editar receitas para podermos verificar se este User é válido no sistema ou não.

Tendo em conta o cariz de rede social da nossa aplicação a ligação à web é obviamente fundamental para poder haver interação entre diferentes utilizadores.

## 4.2 Recursos Humanos

Atendendo que sem a existência de alguém para a criar e também de alguém para o qual a aplicação fosse desenhada esta não teria sentido sem os recursos humanos , nem sem uma clara definição destes uma vez que se trata de uma das parte mais essenciais do desenvolvimento de software, pois afinal sem pessoas para utilizar, implementar e/ou tirar proveito haveria mesmo algum sentido na criação desta aplicação..

Assim, os engenheiros de software, os administradores, os testers, têm um papel fundamental na conceptualização, criação e manutenção da aplicação, bem como no suporte futuro desta para que se mantenha ao corrente da procura.

Finalmente, não podemos deixar de falar no utilizador final que é aquele que vai usar e obter proveito da aplicação gerando lucro à equipa de trabalho que a desenvolveu.

## 4.3 Recursos de Hardware

Em termos de requisitos mínimos para o funcionamento da aplicação, a nível de *hardware* não existem necessidades específicas além do comum, isto é qualquer computador ou *SmartPhone* capaz de acesso a internet e que tenha um WebBrowser deve ser capaz de utilizar a aplicação, uma vez que as necessidades de processamento e memória não são muito elevadas e se encontram todas em Cloud.

Para uma integração da aplicação e aproveitamento de todas as funcionalidades, o computador ou *SmartPhone* deve possuir microfone para a reconhecimento de voz que sirva como interface aos controlos por voz. Contudo este requisito não é exclusivo uma vez que a aplicação responde a comandos vocais bem como textuais sendo possível utilizar a aplicação embora sem o proveito de uma das funcionalidades que a distingue.

## **4.4 Recursos de Software**

A aplicação foi desenvolvida com recurso a ferramentas da *Microsoft* (eg. *Visual Studio*, *SQL Server*, *entre outras*), mas temos como principal objetivo o funcionamento desta para todo o tipo de plataformas.

## **4.5 Recursos de Dados**

Sendo esta aplicação uma rede social todos os dados serão guardados no servidor.

Desta forma os dados serão recolhidos e atualizados online, deste modo é garantido que o utilizador terá acesso a dados atualizados acerca das suas subscrições e balanço de modo a conseguir subscrever mais conteúdos.

## **5. Medidas de sucesso**

Durante a fase de planeamento definimos algumas medidas de monitorização de modo a que posteriormente nos seja possível classificar/ avaliar o sucesso do projeto.

Como primeira medida decidimos que seria a satisfação, tanto do grupo de trabalho como do nosso cliente, relativamente ao resultado final do projeto. Esta etapa tem como objetivo o cumprimento de todos os requisitos levantados junto do cliente assim como algumas ideias adicionais que o grupo achou relevantes.

Outra medida será terminar o projeto dentro dos prazos estabelecidos com o cliente, cumprindo todas as exigências impostas, assim como os prazos estimados pelo grupo para cada fase do projeto.

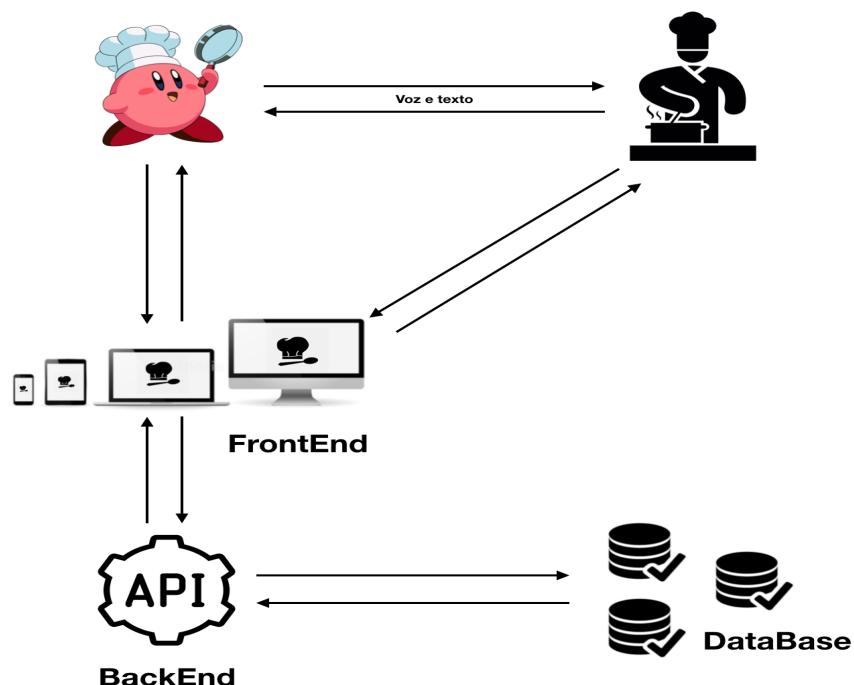
A última medida será a qualidade do projeto final. Esta avaliação será feita pelos utilizadores.

Esta rede deve ser acessível a todo o tipo de utilizadores que possam estar interessados, para tal optamos por implementar um sistema que incentiva os utilizadores a comunicar a sua experiência com a aplicação, criando a possibilidade de eles sugerirem possíveis alterações nas futuras atualizações e permitindo a classificação do sistema de modo a avaliar o nível de satisfação.

## 6. Maqueta do sistema

A camada de apresentação possui todas as ferramentas necessárias para a interação entre o utilizador e o sistema, incluindo o envolvimento culinário do Assistente de cozinha inteligente que desenvolvemos. Estas encontram-se implementadas na camada de negócio que trata os dados e respostas necessárias para uma execução fiável e eficiente destas funcionalidades bem como a comunicação entre o Front-End e a camada Back-End.

Assim dividimos a estrutura da aplicação em Front-End, Back-End e Business cada um com as respetivas funcionalidades que permitirão o bom funcionamento de modo ilustrado na maqueta abaixo:



**Figura 1 - Maquete do Sistema**

## 7. Plano de Desenvolvimento do Projeto

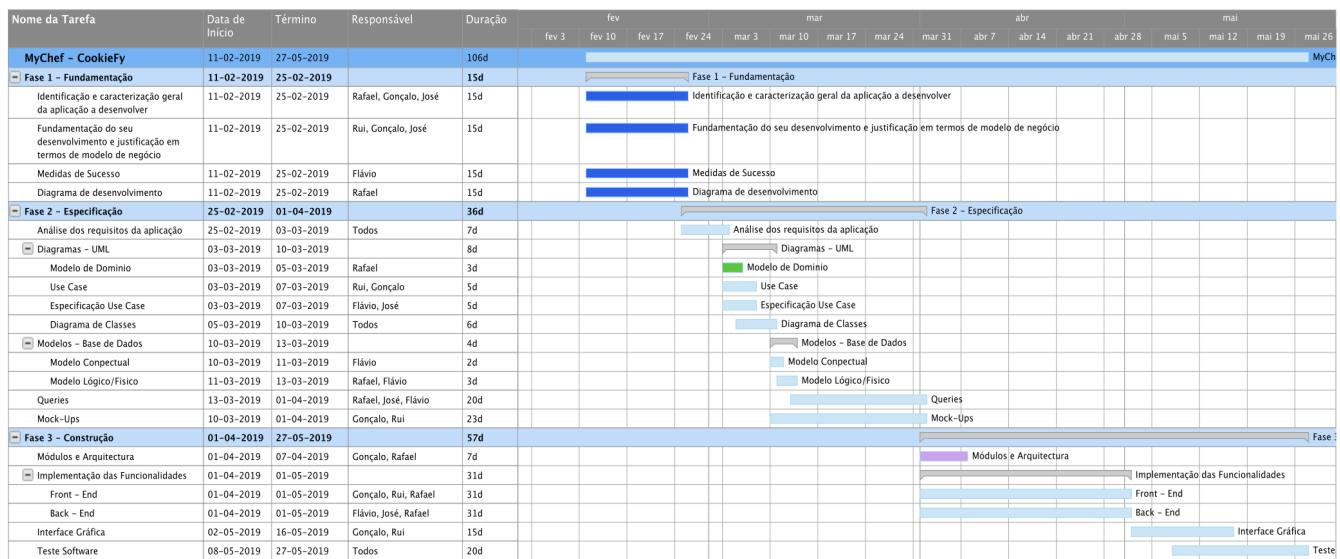
Para desenvolvermos um bom projeto e bem organizado, não nos podemos esquecer da importância do plano de desenvolvimento do mesmo, sendo distribuído o trabalho por cada membro da equipa de Software equitativamente, e ao longo de um intervalo de datas, sendo um método eficaz e bem estruturado, tal como previsto pela Engenharia de Software.

### 7.1 Diagrama de Gantt

Usando o Diagrama de Gantt abaixo, podemos distribuir as tarefas pelos diferentes membros da equipa de Software e atribuindo uma data de início e término.

Para realizar para tal tarefa, decidimos dividir o projeto de Software em três importantes fases.

Começando primeiro pela Fundamentação do mesmo projeto, de seguida pela Especificação, e por fim pela sua Construção, obtendo por fim com a junção destas etapas o Software que pretendemos criar, que neste caso é denominado “MyChef - CookieFy”.



**Figura 2 - Diagrama de Gantt**

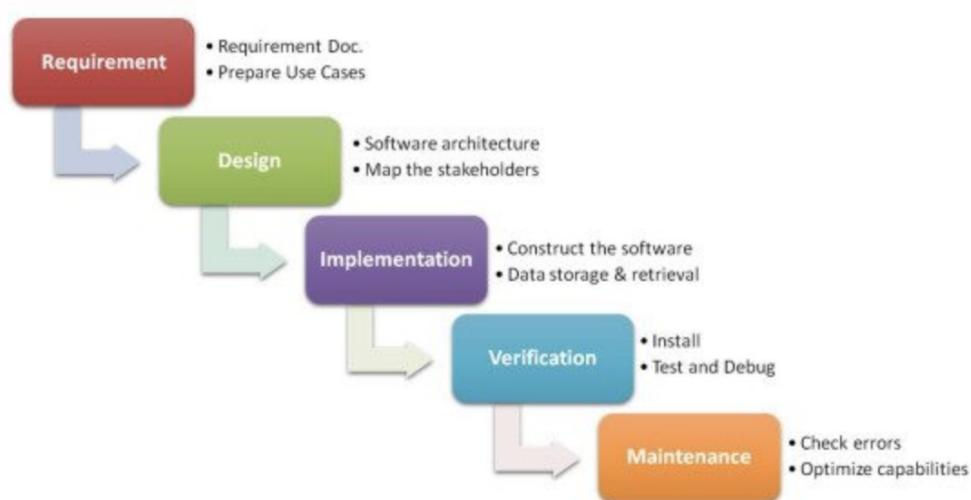
## 7.2 Modelo em Cascata

O desenvolvimento de um Software deve ser enfrentado com grande ambição e como uma grande responsabilidade pelas entidades presentes nele.

Este modelo tem como intenção mostrar as etapas pelo qual o desenvolvimento do Software deve ultrapassar, é conhecido como o modelo em cascata, pois o mesmo é uma sequência de uma etapa anterior, ou seja, que é um desenvolvimento sequencial, que é constituído pelas seguintes etapas:

- Análise de Requisitos.
- Design.
- Implementação.
- Verificação, ou fase de Testes.
- Manutenção.

O modelo poderá ser visto então como uma cascata na imagem em baixo:



**Figura 3 - Modelo Cascata**

## 8. Requisitos

### **Requisitos Não Funcionais**

#### Requisitos do Sistema

1. O sistema deverá suportar **receitas**. Estas serão adicionadas por utilizadores e são compostas por:
  - a. Todos os **ingredientes** necessários para a confeção da receita e ingredientes alternativos (opcional);
  - b. **Valores nutricionais** de todos os ingredientes e valor total das receitas, atualizando sempre que o utilizador mudar algum ingrediente;
  - c. Lista de **utensílios** necessários;
  - d. **Lista de passos** necessários para a confeção da receita, estes passos devem ser claros e sem ambiguidades;
  - e. **Sub receitas**, isto é, caso a receita tenha preparados (exemplo: uma lasanha tem um preparado de carne) deverá estar indicado antes do início da receita e durante (no passo necessário), as sub receitas podem conter um **link para receitas já existentes no sistema** e/ou **vídeos do Youtube**;
  - f. **Técnicas**, se uma receita tiver técnicas de cozinha mais elaboradas também deverá ser indicado antes da confeção da receita e durante as técnicas devem conter um link para **tutoriais do Youtube**.
2. A aplicação deverá ter a **carteira CookieFy**, onde os utilizadores poderão carregar com CookieCoins para depois comprar receitas;
3. A App necessita **acesso ao microfone e speakers** do dispositivo do utilizador, pois toda a comunicação entre o utilizador e o assistente será por voz;
4. A app tem de **ser compatível com qualquer dispositivo**;
5. O sistema tem de estar **sempre disponível** (exceto em manutenção);
6. **Lista de receitas**, a lista de receitas é uma espécie de arquivo onde o utilizador poderá agrupar n receitas, tal como uma playlist de músicas, cada lista de receitas deverá ter um **nome** e, claro, **receitas**.

## Requisitos do Utilizador

1. Um **utilizador** é composto por:
  - a. Nome;
  - b. Idade;
  - c. Profissão;
  - d. Morada;
  - e. Email;
  - f. Foto de perfil;
  - g. Descrição;
2. Os utilizadores deverão ser divididos em 4 categorias:
  - a. **Visitante**, este utilizador não terá qualquer conta logo só poderá **consultar receitas e utilizar o assistente**, irá ter **publicidade (lateral)** em todas as páginas que visitar no nosso sistema;
  - b. **Normal**, este utilizador terá conta, logo poderá utilizar todas as funcionalidades da aplicação apenas com algumas limitações:
    - i. Não poderá monetizar receitas;
    - ii. No planeador poderá apenas definir 1 refeição diária;
    - iii. Estará também sujeito a publicidade.
  - c. **Premium**, este utilizador terá conta e terá também de pagar uma subscrição mensal ou anual, não terá qualquer limitação nem publicidade em toda aplicação;
  - d. **Administrador**, este utilizador irá gerir a aplicação.
3. O utilizador tem de ter **acesso à internet**;
4. O utilizador tem de estar **autenticado** para usufruir de todas as funcionalidades da aplicação, caso não seja autenticado o utilizador será apenas visitante, a autenticação será feita através de **email e password**;

## Requisitos Funcionais

### Requisitos do Sistema

1. O sistema deverá disponibilizar a todos os clientes um **assistente de cozinha**. Este deverá ser capaz de:
  - a. **Acompanhar passo a passo** a execução da receita, isto é, o assistente deverá “traduzir” (**text to speech**) todos os passos indicados pelo utilizador que criou a receita;
  - b. **Aguardar** a conclusão de cada passo antes de avançar para o seguinte;
  - c. **Temporizar** o tempo de confeção da receita informando o utilizador quando o tempo previsto para o mesmo estiver concluído;
  - d. **Repetir** o passo presente caso o utilizador não tenha percebido ou tenha alguma dúvida por esclarecer;
  - e. **Avançar passo**, o sistema deverá disponibilizar ao utilizador a possibilidade de avançar para o próximo passo da receita;
  - f. **Recuar passo**;

- g. **Iniciar temporizador**, caso o passo atual da confeção de uma dada receita necessitar de um temporizador o assistente deverá iniciar o temporizador apenas quando o utilizador assim o requisitar;
- h. Direccionar o utilizador para sub receitas ou tutoriais no Youtube, quando exista algum associado ao passo em questão, caso o utilizador precise de ajuda para realizar o passo;
- i. **Avançar música, música anterior e pausar música** através de comandos voz o assistente deverá executar estas instruções na App do Spotify;
- j. **Reducir ou aumentar** o volume, o assistente deverá permitir ao utilizador reduzir ou aumentar o volume da música no Spotify;
- k. Sempre que o assistente estiver a **comunicar** com o utilizar e **vice-versa** o volume da música deve ser reduzido ou mesmo pausar a música.

## Requisitos do Utilizador

1. **Criar conta**, para criar conta o utilizador deverá preencher um formulário com os seguintes inputs:
  - a. Nome;
  - b. Idade;
  - c. Profissão;
  - d. Morada;
  - e. Email válido;
  - f. Foto de perfil;
  - g. Descrição;
  - h. Password.
2. **Criar receitas**, isto é, o utilizador deverá preencher um formulário bem definido, onde terá de indicar todos os ingredientes e utensílios, uma descrição da receita e todos os passos necessários, se existirem técnicas específicas ou sub receitas o utilizador deverá indicar na criação do passo e adicionar um url para um tutorial ou para um sub-receita;
3. **Editar receitas**, os utilizadores premium poderão editar as receitas subscritas, isto é, poderá editar ingredientes e quantidades, desta forma qualquer receita poderá ser ao gosto de cada utilizador;
4. **Monetizar receitas**, isto é o utilizador premium poderá colocar um preço sobre as receitas que criou, assim irá receber uma percentagem sempre que algum utilizador subscrever as suas receitas;
5. **Definir e visualizar o plano de refeições semanais**, será possível cada utilizador definir o seu menu semanal, isto é, poderá criar n refeições diárias e escolher receitas para confeccionar em cada refeição, no fim poderá retirar uma **lista de ingredientes** (com as respetivas quantidades), a **lista de utensílios necessários** e os **valores nutricionais por refeição**;
6. **Consultar receitas** (visualizar ou iniciar confeção com assistente de voz), todos os utilizadores poderão **visualizar as receitas**, isto é, poderá consultar os ingredientes, utensílios, valores nutricionais, descrição, sub receitas e técnicas utilizadas, por fim se o utilizador subscrever a receita poderá **iniciar o assistente** de voz e confeccionar a receita;

7. **Comprar receitas premium**, se algum utilizador quiser confeccionar receitas monetizadas terá de pagar por elas, o dinheiro será descontado da **carteira cookiefy** ou pago por **mbway**;
8. **Adicionar** dinheiro à carteira CookieFy, o utilizador poderá adicionar dinheiro na carteira para depois subscrever receitas monetizadas ou até pagar pacotes de subscrições;
9. **Consultar e editar** o perfil, cada utilizador poderá editar qualquer informação do seu perfil;
10. **Subscrever conta premium**, os utilizadores não premium poderão subscrever um pacote mensal ou anual que lhes irá conceder todos os privilégios de um utilizador premium;
11. **Comentar receitas**, qualquer utilizador poderá comentar receitas;
12. Criar **lista de receitas**, tal como no Spotify os **utilizadores premium** poderão criar listas de receitas;
13. **Avaliar receitas**, no fim da confeção de uma receita deverá ser pedido a todos os utilizadores para deixarem uma classificação de 0 a 5, esta classificação será usada para caracterizar as receitas, facilitando a escolha dos utilizadores;
14. Dar **follow** a outros utilizadores, qualquer utilizador poderá seguir outros utilizadores, desta forma os utilizadores terão mais conteúdo do seu interesse no feed;
15. **Apagar** receitas, qualquer utilizador poderá apagar as suas receitas desde que estas não sejam monetizadas;
16. **Pesquisar** receitas (filtros):
  - a. **Mais relevantes** grátis ou monetizadas, poderá pesquisar pelas receitas com melhor avaliação;
  - b. **Mais recentes**.
17. **Pesquisar** utilizadores por nome ou email, para os Chefs da aplicação deverá haver uma zona específica para procurar, poderá então **consultar** o perfil de outros utilizadores e as suas receitas.

## Requisitos Utilizador Administrador

1. **Bloquear** utilizadores, o administrador poderá bloquear utilizadores caso estes tenham algum comportamento abusivo, os bloqueios poderão ser de 1 dia, 1 semana, 1 mês, 1 ano ou permanente;
2. **Apagar comentários**;
3. **Apagar receitas**;

## **9. Especificação do Software**

Na primeira fase de Planeamento e Fundamentação do Software, tem como objetivo contextualizar o pedido requerido pela parte do Cliente. Como foi definido pelo nosso Cliente, a nossa aplicação trata-se de uma interface de culinária, com assistente pessoal.

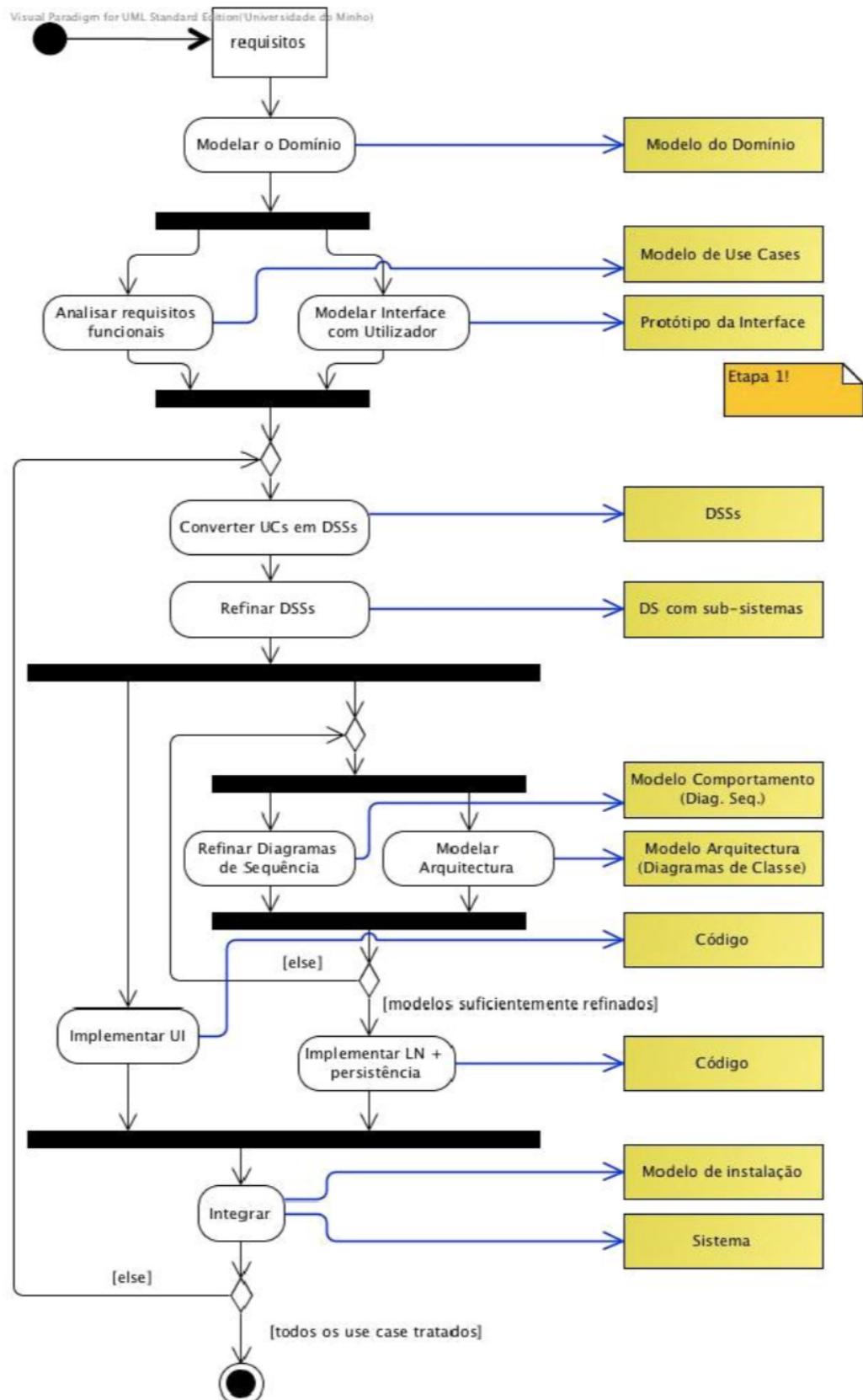
Nesta segunda fase, seguimos os métodos usados no Desenvolvimento de Software, tendo dedicado grande parte do tempo para as estudar e desenvolver diferentes estratégias.

Nesta secção iremos aplicar o conceito de Diagrama de Classes, Modelo de Domínio, Diagramas de Sequência, Use Cases e por fim a documentação necessária para validar estes modelos. Através dos modelos descritos em cima conseguimos apresentar uma boa estruturação, bem como modelação e especificação, para que uma equipa de programadores seja consiga construir a aplicação, como recurso auxiliar esta modelação.

Dentro dos nossos conhecimentos atuais fomos capazes de criar a nossa “aplicação” que demonstra perfeitamente com a aplicação terá de ser desenvolvida ao longo do tempo. Pensamos então que o que nós criamos é capaz de mostrar o que pretendemos, mostrando ao leitor o necessário e com informação direta.

Tornado assim mais fácil a compreensão do documento, e tornando mais fácil caso seja necessário intervir e alterar ou esclarecer alguma dúvida no que toca a modelação e as especificações.

De uma forma abstrata iremos mostrar o diagrama de atividades que tomamos como guia para nos ajudar nesta etapa no Desenvolvimento do Software. As últimas etapas não serão para esta fase, serão consideradas mais a frente quando o Software for desenvolvido. Esse diagrama encontra-se na figura em baixo.

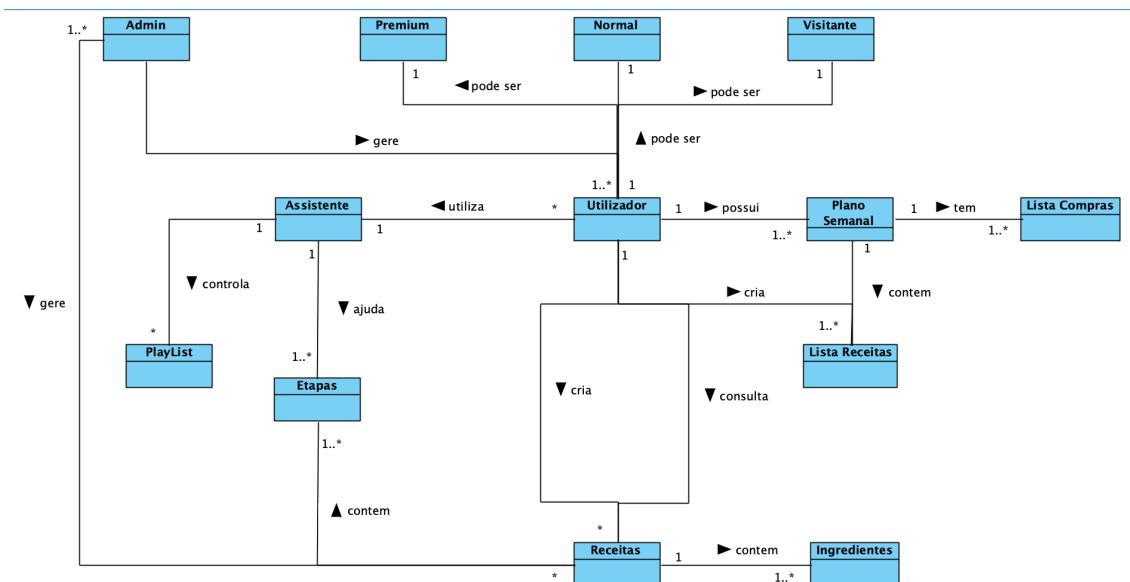


**Figura 4** - Diagrama Usado como Guia no Desenvolvimento do Projeto

## 9.1. Modelo de Domínio

Após a análise do caso de estudo e respetivo levantamento de requisitos, quer funcionais quer não funcionais, procedemos ao desenvolvimento do Modelo de Domínio. Este tem como objetivo capturar as entidades do problema e os relacionamentos entre elas e, assim, descreve um conjunto de características para os problemas aos quais a aplicação pretende dar solução.

De seguida, apresenta-se o respetivo modelo de domínio e é feita uma descrição deste.



**Figura 5 - Modelo de Domínio**

Como podemos ver no modelo de Domínio o Utilizador pode ser identificado como Premium, Normal, e Visitante, sendo que cada um tem um conjunto de permissões diferentes, uma ficha de utilizador e toda a informação que a sua contém tem isto é, Plano Semanal, Lista de Receitas e Lista de Compras que podem ser geradas na aplicação. O Utilizador tem também acesso ao assistente e cozinha que controla a playlist de música e ajuda nas várias etapas de confeção da receita.

## 9.2. Diagramas de Use Case

De forma a proceder à identificação dos Use Cases, tivemos que ter em atenção os seguintes elementos:

1. Identificar atores (quem utiliza o sistema).
2. Identificar Use Cases (o que se pode fazer no sistema).
3. Identificar associações (quem pode fazer o quê).

Após termos seguido os passos anteriores, chegamos aos Use Cases que serão devidamente apresentados ao longo desta secção. Neste caso, visto que ainda não atingimos o passo da especificação dos Use Cases, estes serão apresentados sob a forma de Diagrama.

Os utilizadores default tal como temos nos requisitos podem fazer praticamente tudo o que os utilizadores premium fazem apenas com algumas restrições que estão devidamente assinaladas. O visitante pode apenas consultar receitas, pesquisar receitas e confeccionar receitas.



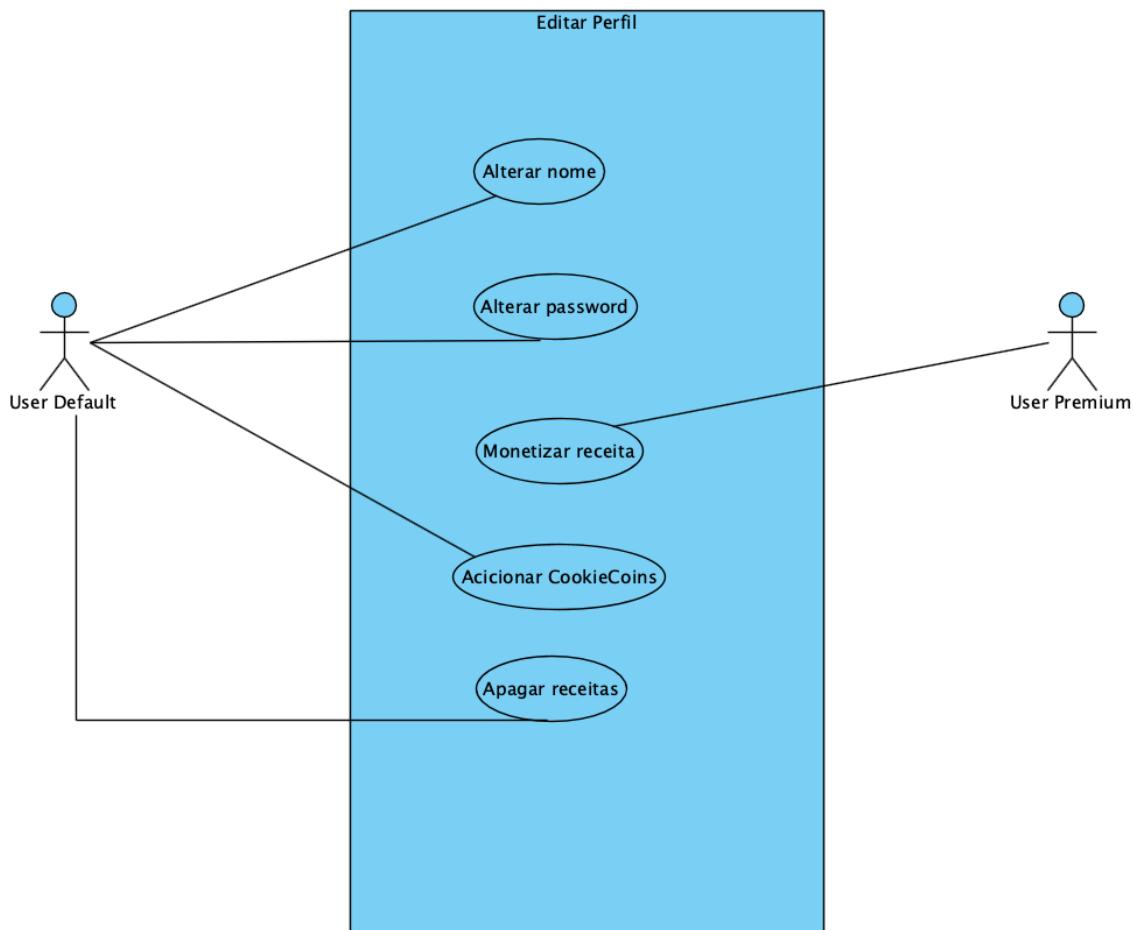
**Figura 6 - Uses Cases**

No nosso diagrama temos 3 subsistemas:

1. Editar Perfil;
2. Confeccionar Receitas;
3. Consultar Receitas.

### **Editar Perfil**

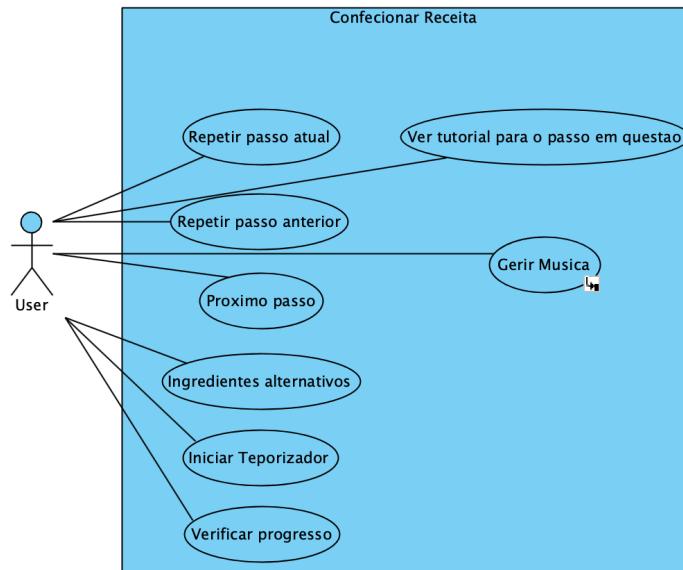
Tal como podemos ver no use case seguinte e como foi referido anteriormente nos requisitos apenas o utilizador premium pode monetizar receitas.



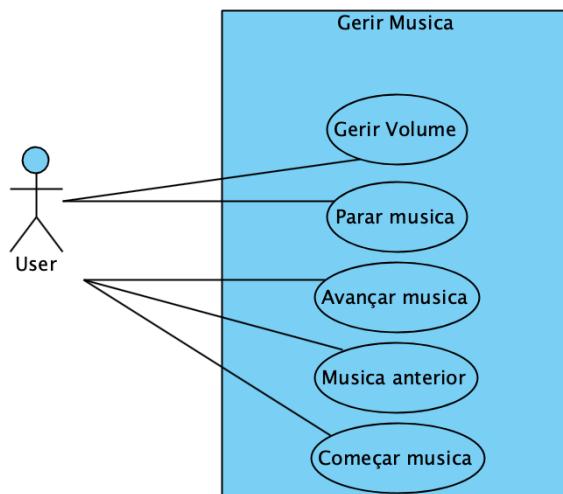
**Figura 7 - Use Case Editar Perfil**

## Confeccionar Receita

Qualquer utilizador pode confeccionar receitas, seja autenticado ou apenas visitante.



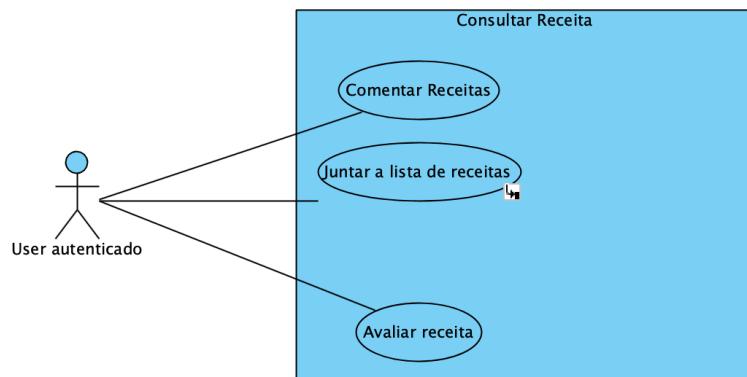
**Figura 8** - Use Case Confeccionar Receita



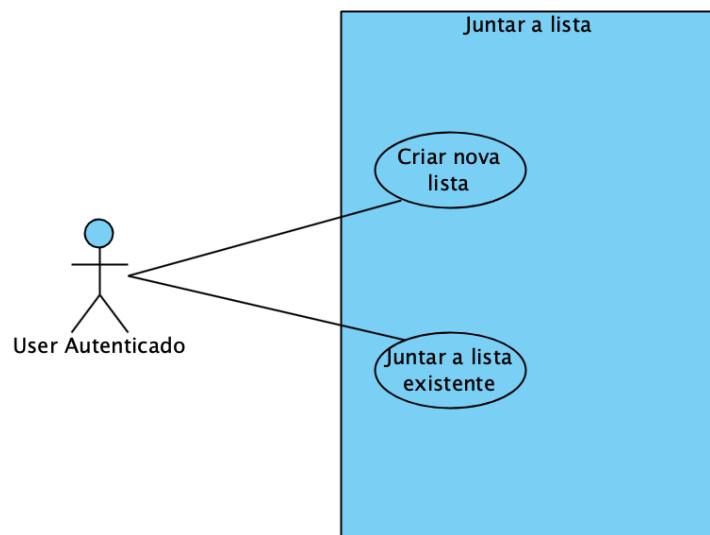
**Figura 9** - Use Case Gerir Música

## Consultar Receita

Tal como previsto qualquer utilizador pode consultar uma receita, porém apenas os utilizadores autenticados, ou seja, têm conta na aplicação, podem comentar receitas, adicionar uma lista a uma “playlist” e avaliar as receitas.



**Figura 10 - Use Case Consultar Receita**



**Figura 11 - Use Case Juntar a Lista**

### 9.2.1. Especificação de Use Cases

Nesta subsecção, apresentamos a especificação dos Use Cases apresentados na subsecção anterior, de modo a compreender melhor o que se pode fazer, no que toca ao utilizador interagir com o sistema.

#### Gestão

Use Case:	Adicionar CookieCoins	
Actor:	User	
Pré condição:	User autenticado	
Pós condição:	Quantia de CookieCoins atualizada	
Cenário Normal	Actor input	System response
	1. Seleciona a opção de adicionar CookieCoins	3. Mostra formulario de adição de coins
	4. Preenche formulário	5. Verifica campos 6. Regista alteração na quantidade de coins do utilizador
Comp.Alt. 1 (Passo 5) [Dados introduz. inválidos]		2.1 Informa da invalidade dos dados de transferencia introduzidos 2.2 Volta ao passo 3

**Figura 12 - Use Case Adicionar CookieCoins**

Use Case:	Alterar descrição	
Actor:	User	
Pré condição:	User autenticado	
Pós condição:	Registo de alteração de descrição efetuado	
Cenário Normal	Actor input	System response
	1. Selecciona a opção alterar descrição de user	2. Mostra menu de alteração de descrição
	3. Introduz nova descrição	4. Verifica validade dos dados introduzidos 5. Informa do sucesso na alteração da descrição
Comp. Alt. 1 (Passo 3) [dados introduz. invalidos]		2.1 Informa que a descrição introduzida não é válida (Campo vazio) 2.2 Volta ao passo 2

**Figura 13 - Use Case Alterar Descrição**

Use Case:	Alterar Nome	
Actor:	User	
Pré condição:	User autenticado	
Pós condição:	Registo de alteração de nome efetuado	
Cenário Normal	Actor input	System response
	1. Selecciona a opção alterar nome de user	2. Mostra menu de alteração de nome
	3. Introduz novo nome	3. Verifica validade do nome introduzido 4. Altera nome
Comp. Alt. 1 (Passo 3) [dados introduz. invalidos]		2.1 Informa que o nome introduzido não é válido (muito longo ou muito curto) 2.2 Volta ao passo 2

**Figura 14 - Use Case Alterar Nome**

Use Case:	Alterar password	
Actor:	User	
Pré condição:	User autenticado	
Pós condição:	Registo de alteração de password efetuado	
Cenário Normal	Actor input	System response
	1. Seleciona a opção alterar password de user	2. Mostra menu de alteração de password
	3. Introduz nova password	4. Veifica validade dos dados introduzidos 5. Informa do sucesso na alteração da password
Comp. Alt. 1 (Passo 3) [dados introduz. invalidos]		2.1 Informa que a password introduzida não é válida (Campo vazio) 2.2 Volta ao passo 2

**Figura 15 - Use Case Alterar Password**

Use Case:	Alterar Profissão	
Actor:	User	
Pré condição:	User autenticado	
Pós condição:	Registo de alteração de profissão efetuado	
Cenário Normal	Actor input	System response
	1. Seleciona a opção alterar profissão de user	2. Mostra menu de alteração de profissão
	3. Introduz nova profissão	4. Veifica validade dos dados introduzidos 5. Informa do sucesso na alteração da profissão
Comp. Alt. 1 (Passo 3) [dados introduz. invalidos]		2.1 Informa que a profissão introduzida não é válida (Campo vazio) 2.2 Volta ao passo 2

**Figura 16 - Use Case Alterar Profissão**

**Figura 17 - Use Case Consultar Perfil**

Use Case:	Criar Conta	
Actor:	User	
Pré condição:	Ligaçāo ao servidor estabelecida	
Pós condição:	Mais um utilizador no sistema	
Cenário Normal	Actor input	System response
	1. Seleciona a opção de criar conta	2. Mostra campos de criação de conta
	3. Preenche campos	4. Informa que conta foi criada com sucesso
Comp. Alt 1 (Passo 2) [E-mail já utilizado]		2.1 Informa que o email introduzido já se encontra no sistema 2.2 Volta a passo 2

**Figura 18 - Use Case Criar Conta**

Use Case:	Login	
Actor:	User	
Pré condição:	Ligaçāo ao servidor estabelecida	
Pós condição:	User autenticado	
Cenário Normal	Actor input	System response
	1. User seleciona opção de login	2. Mostra campos de login
	3. Preenche campos	4. Verifica credenciais de user
Comp. Alt. 1 (Passo 4) [Credenciais de user invalidas]		2.1 Informa que as credenciais introduzidas não são validas (username ou password) 2.2 Volta a passo 2

*Figura 19 - Use Case Login*

Use Case:	Logout	
Actor:	User	
Pré condição:	User autenticado	
Pós condição:	Menos um user online	
Cenário Normal	Actor input	System response
	1. User seleciona opção de logout	2. Mostra mensagem de logout

*Figura 20 - Use Case Logout*

Use Case:	Pesquisar receitas	
Actor:	User	
Pré condição:	User autenticado	
Pós condição:	Lista de receitas na pesquisa do user mostradas	
Cenário Normal	Actor input	System response
	1. Seleciona a opção de pesquisar receitas	3. Mostra campo de pesquisa
	4. Introduz dados para pesquisa	5. Procura dados no sistema consoante o input do user 6. Mostra dados obtidos pela pesquisa
Exception 1 (Passo 5) [Dados não encontrados]		2.1 Informa que não existem dados no sistema que correspondam à pesquisa efetuada

**Figura 21 - Use Case Pesquisar Receitas**

Use Case:	Pesquisar users	
Actor:	User	
Pré condição:	User autenticado	
Pós condição:	Lista de users na pesquisa do user mostradas	
Cenário Normal	Actor input	System response
	1. Seleciona a opção de pesquisar users	3. Mostra campo de pesquisa
	4. Introduz dados para pesquisa	5. Procura dados no sistema consoante o input do user 6. Mostra dados obtidos pela pesquisa
Exception 1 (Passo 5) [Dados não encontrados]		2.1 Informa que não existem dados no sistema que correspondam à pesquisa efetuada

**Figura 22 - Use Case Pesquisar Users**

Use Case:	Definir plano de receitas semanal	
Actor:	User	
Pré condição:	User autenticado	
Pós condição:	User com plano de receitas semanal definido	
Cenário Normal	Actor input	System response
	1. Seleciona a opção de criar plano semanal de receitas	
		3. Dispõe receitas para escolha
	4. Escolhe receitas e respetivos horários	
		4. Verifica permissões do user para as receitas escolhidas
		5. Verifica se user tem permissões para todos os horários escolhidos
		6. Cria plano de receitas semanal
Comp. Alt. 1 (Passo 4) [User sem acesso a uma receita selecionada]		2.1 Informa da falta de permissões de acesso a uma receita escolhida 2.2 Volta a passo 3
Comp. Alt. 2 (Passo 5) [User escolheu horário invalido]		2.1 Informa da falta de permissões de escolha de horário 2.2 Volta a passo 3

**Figura 23 - Use Case Definir Plano de Receitas Semanal**

Use Case:	Subscrever a premium	
Actor:	User	
Pré condição:	User autenticado	
Pós condição:	Registo de novo user premium adicionado ao sistema	
Cenário Normal	Actor input	System response
	1. Seleciona a opção de subscrever ao plano premium	
		3. Verifica se user tem CookieCoins suficientes
		4. Atribui estatuto de premium ao user
Exception 1 (Passo 3) [Saldo não suficiente]		2.1 Informa que o user não possui saldo de cookieCoins suficiente para a subscrição do serviço premium

**Figura 24 - Use Case Subscrever a Premium**

Use Case:	Alterar Morada	
Actor:	User	
Pré condição:	User autenticado	
Pós condição:	Registo de alteração de morada efetuado	
Cenário Normal	Actor input	System response
	1. Selecciona a opção alterar morada de user	2. Mostra menu de alteração de morada
	3. Introduz nova morada	4. Verifica validade dos dados introduzidos 5. Informa do sucesso na alteração da morada
Comp. Alt. 1 (Passo 3) [dados introduz. invalidos]		2.1 Informa que a morada introduzida não é válida (Campo vazio) 2.2 Volta ao passo 2

**Figura 25 - Use Case Alterar Morada**

## Receita

Use Case:	Adicionar receita à lista de receitas do user	
Actor:	User	
Pré condição:	User autenticado	
Pós condição:	Novo registo de receita adicionada à lista de receitas	
Cenário Normal	Actor input	System response
	1. Selecciona a opção de adicionar receita a lista de receitas	2. Verifica permissões do user para a adição da receita em questão 3. Apresenta opções de adição à lista de receitas(criar nova ou adicionar a lista já existente) 4. Informa que receita foi adicionada à lista com sucesso
Exception 1 (Passo 2) [User sem permissões]		2.1 Informa que o user não tem permissões para adicionar a receita escolhida à sua lista de receitas

**Figura 26 - Use Case Adicionar Receita à lista de receitas do User**

Use Case:	Adicionar receita a lista de receitas já existente	
Actor:	User	
Pré condição:	User autenticado	
Pós condição:	Receita adicionada a lista de receitas	
Cenário Normal	Actor input	System response
	1. Seleciona a opção de adicionar receita a lista já existente	2. Verifica se o user já tem listas de receitas criadas 3. Mostra listas de receitas disponíveis
	4. Escolhe lista de receitas	5. Adiciona receita a lista escolhida
Exception 1 (Passo 2) [User não tem listas criadas]		2.1 Informa que o user não tem listas de receitas criadas

**Figura 27 - Use Case *Adicionar Receita à lista de receitas já existente***

Use Case:	Adicionar receita	
Actor:	User	
Pré condição:	User autenticado	
Pós condição:	Lista de receitas atualizada	
Cenário Normal	Actor input	System response
	1. Seleciona a opção de adicionar receita	2. Verifica permissões do user para adicionar uma receita 3. Mostra menu de adição de receitas
	4. Preenche campos relativos à definição da receita	
	5. Confirma adição de receita	6. Verifica dados introduzidos 7. Regista alterações no sistema
Exception 1 (Passo 2) [User sem permissões para adicionar receita]		2.1 Mensagem com a informação de que o user não tem permissões para adicionar receitas
Comp.Alt 1 (Passo 6) [Dados introduzidos inválidos]		6.1 Informa que os dados introduzidos não são válidos (volta a passo 3)

**Figura 28 - Use Case *Adicionar Receita***

Use Case:	Apagar receitas	
Actor:	User	
Pré condição:	User autenticado	
Pós condição:	Menos um registo de receita no sistema	
Cenário Normal	Actor input	System response
	1. Selecciona a opção de apagar receitas	2. Mostra lista de receitas do user
	3. Selecciona receita a apagar	4. Informa que receita foi apagada

**Figura 29 - Use Case Apagar Receitas**

**Figura 30 - Use Case Avaliar Receitas**

Use Case:	Comentar Receita	
Actor:	User	
Pré condição:	User autenticado	
Pós condição:	Novo registo de comentário na receita em questão	
Cenário Normal	Actor input	System response
	1. Seleciona a opção de comentar receita	
		2. Mostra campo de comentário
	3. Preenche campo de comentário	
		4. Adiciona comentário

**Figura 31 - Use Case Comentar Receita**

Use Case:	Comprar receita premium	
Actor:	User	
Pré condição:	User consulta receita premium	
Pós condição:	Lista de receitas do user atualizada	
Cenário Normal	Actor input 1. Seleciona a opção de comprar a receita premium	System response 3. Verifica se o user tem saldo suficiente para comprar a receita escolhida 4. Confirma compra
Exception 1 (Passo 3) [User sem saldo suficiente]	2.1 Informa da falta de saldo.	

**Figura 32 - Use Case Comprar receita premium**

**Figura 33 - Use Case Consultar Receita**

Use Case:	Criar lista de receitas	
Actor:	User	
Pré condição:	User autenticado	
Pós condição:	Lista de receitas adicionada ao sistema	
Cenário Normal	Actor input	System response
	1. Seleciona a opção de criar nova lista de receitas	
		2. Mostra campos de criação de lista
	3. Preenche campos	
		4. Verifica dados introduzidos
		5. Adiciona receita à nova lista
		6. Mostra mensagem de sucesso ao criar lista de receitas
Comp. Alt. 1 (Passo 4) [dados introduz. invalidos]		2.1 Dados introduzidos invalidos (nome de lista de receitas vazio) 2.2 Volta a passo 2

**Figura 34 - Use Case Criar Lista de Receitas**

Use Case:	Editar receita	
Actor:	User	
Pré condição:	Receita disponível no sistema	
Pós condição:	Lista de receitas atualizada	
Cenário Normal	Actor input	System response
	1. Seleciona a opção de editar receita	2. Verifica permissões do user para editar a receita em questão 3. Mostra campos de receita que podem ser editados
	4. Preenche campos de edição	
	5. Conclui edição	6. Regista as alterações no sistema
Exception 1 (Passo 3) [Componente esncolhido invalido]		2.1 Informa que os dados introduzidos não são validos
Exception 2 (Passo 4) [Componente já existente]		2.1 Informa que já existe um componente igual no sistema

**Figura 35 - Use Case *Editar Receita***

Use Case:	Eliminar receita	
Actor:	User	
Pré condição:	Receita disponível no sistema	
Pós condição:	Lista de receitas atualizada	
Cenário Normal	Actor input	System response
	1. Seleciona a opção de eliminar receita	2. Verifica permissões do user para eliminar a receita em questão 3. Remove receita do sistema
Exception 1 (Passo 2) [User sem permissões para eliminar a receita]		2.1 Mensagem de erro com a informação de que o user não tem permissões para eliminar a receita em questão

**Figura 36 - Use Case *Eliminar Receita***

**Figura 37 - Use Case Monetizar Receita**

*Gerir Música*

Use Case:	Avancar musica	
Actor:	User	
Pré condição:	User a confeccionar uma receita a ouvir musica	
Pós condição:	Musica seguinte a reproduzir	
Cenário	Actor input	System response
Normal	1. Solicita o avanco de uma musica	2. Interpreta pedido 3. Avanca musica
Comp. Alt. 1 (Passo 2) [Pedido não foi recebido com clareza]		2.1 Informa que os dados não foram recebidos com clareza 2.2 Passa ao passo 1

**Figura 38 - Use Case Avançar Música**

Use Case:	Comecar musica	
Actor:	User	
Pré condição:	User a confeccionar uma receita	
Pós condição:	Musica a reproduzir	
Cenário Normal	Actor input	System response
	1. Solicita a reproducao musica	2. Interpreta pedido 3. Reproduz musica (Musica random playlist das mais tocadas)
Comp. Alt. 1 (Passo 2) [Pedido não foi recebido com clareza]		2.1 Informa que os dados não foram recebidos com clareza 2.2 Passa ao passo 1

**Figura 39 - Use Case Começar Música**

Use Case:	Gerir volume	
Actor:	User	
Pré condição:	User a confeccionar uma receita	
Pós condição:	Volume da musica alterado	
Cenário Normal	Actor input	System response
	1. Solicita uma mudanca no volume da musica	2. Interpreta pedido 3. Altera volume da musica (mais Alto ou mais baixo)
Comp. Alt. 1 (Passo 2) [Pedido não foi recebido com clareza]		2.1 Informa que os dados não foram recebidos com clareza 2.2 Passa ao passo 1

**Figura 40 - Use Case Gerir Volume**

Use Case:	Musica anterior	
Actor:	User	
Pré condição:	User a confeccionar uma receita a ouvir musica	
Pós condição:	Musica anterior a reproduzir	
Cenário Normal	Actor input	System response
	1. Solicita a reproducao cda musica anterior	2. Interpreta pedido 3.Reproduz musica anterior
Comp. Alt. 1 (Passo 2) [Pedido não foi recebido com clareza]		2.1 Informa que os dados não foram recebidos com clareza 2.2 Passa ao passo 1

**Figura 41 - Use Case Música Anterior**

Use Case:	Parar musica	
Actor:	User	
Pré condição:	User a confeccionar uma receita a ouvir musica	
Pós condição:	Musica parada	
Cenário Normal	Actor input	System response
	1. Solicita uma pausa na musica	2. Interpreta pedido 3. Para musica
Comp. Alt. 1 (Passo 2) [Pedido não foi recebido com clareza]		2.1 Informa que os dados não foram recebidos com clareza 2.2 Passa ao passo 1

**Figura 42 - Use Case Parar Música**

## Confeção

Use Case:	Solicitar ingredientes alternativos	
Actor:	User	
Pré condição:	User a efetuar uma receita	
Pós condição:	Lista de ingredientes alternativos apresentada	
Cenário Normal	Actor input	System response
	1. Seleciona a opção de solicitar ingredientes alternativos	2. Interpreta pedido 3. Verifica se existem ingredientes alternativos para este passo 4. Apresenta passo com ingredientes alternativos
Comp. Alt. 1 (Passo 2) [Pedido não foi recebido com clareza]		2.1 Informa que os dados não foram recebidos com clareza 2.2 Passa ao passo 1
Exception 1 (Passo 3) [Ingredientes alternativos não existentes]		2.1 Informa que não existem ingredientes alternativos para o passo em questão

**Figura 43 - Use Case *Solicitar Ingredientes Alternativos***

Use Case:	Iniciar temporizador	
Actor:	User	
Pré condição:	User a confeccionar uma receita	
Pós condição:	Temporizador iniciado	
Cenário Normal	Actor input	System response
	1. Solicita o inicio da contagem do Temporizador	2. Interpreta pedido 3. Inicia temporizador
Comp. Alt. 1 (Passo 2) [Pedido não foi recebido com clareza]		2.1 Informa que os dados não foram recebidos com clareza 2.2 Passa ao passo 1

**Figura 44 - Use Case *Iniciar Temporizador***

Use Case:	Próximo passo	
Actor:	User	
Pré condição:	User a confeccionar uma receita	
Pós condição:	Receita no passo seguinte	
Cenário Normal	Actor input	System response
	1. Solicita a passem ao passo seguinte	2. Interpreta pedido 3. Passa ao proximo passo
Comp. Alt. 1 (Passo 2) [Pedido não foi recebido com clareza]		2.1 Informa que os dados não foram recebidos com clareza 2.2 Passa ao passo 1

**Figura 45 - Use Case Próximo Passo**

Use Case:	Repetir passo atual	
Actor:	User	
Pré condição:	User a confeccionar uma receita	
Pós condição:	Passo da receita repetido	
Cenário Normal	Actor input	System response
	1. Solicita a repetição do passo atual	2. Interpreta pedido 3. Repete o passo atual
Comp. Alt. 1 (Passo 2) [Pedido não foi recebido com clareza]		2.1 Informa que os dados não foram recebidos com clareza 2.2 Passa ao passo 1

**Figura 46 - Use Case Repetir Passo Atual**

Use Case:	Repetir passo anterior	
Actor:	User	
Pré condição:	User a confeccionar uma receita	
Pós condição:	Passo anterior da receita repetido	
Cenário Normal	Actor input 1. Solicita a repetição do passo anterior	System response 2. Interpreta pedido 3. Repete o passo anterior
Comp. Alt. 1 (Passo 2) [Pedido não foi recebido com clareza]		2.1 Informa que os dados não foram recebidos com clareza 2.2 Passa ao passo 1

Figura 47 - Use Case Repetir Passo Anterior

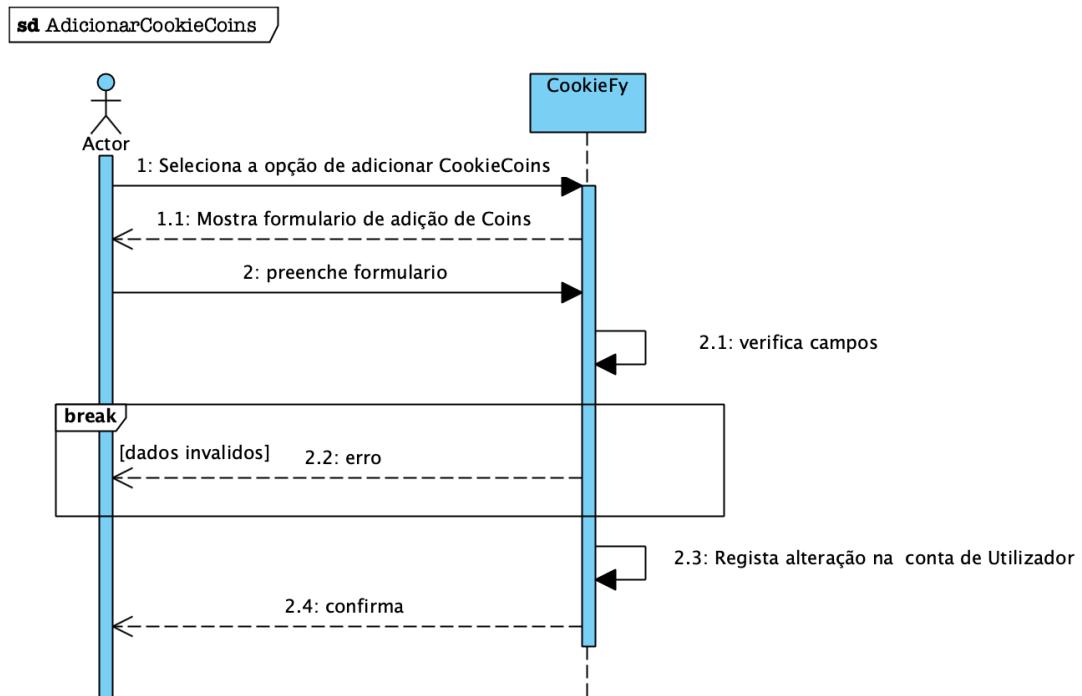
Use Case:	Ver Tutorial	
Actor:	User	
Pré condição:	User a confeccionar uma receita	
Pós condição:	Tutorial do passo reproduzido	
Cenário Normal	Actor input 1. Solicita a reprodução do tutorial do passo em questão	System response 2. Interpreta pedido 3. Reproduz tutorial
Comp. Alt. 1 (Passo 2) [Pedido não foi recebido com clareza]		2.1 Informa que os dados não foram recebidos com clareza 2.2 Passa ao passo 1

Figura 48 - Use Case Ver Tutorial

### 9.3. Diagrama de Sequência

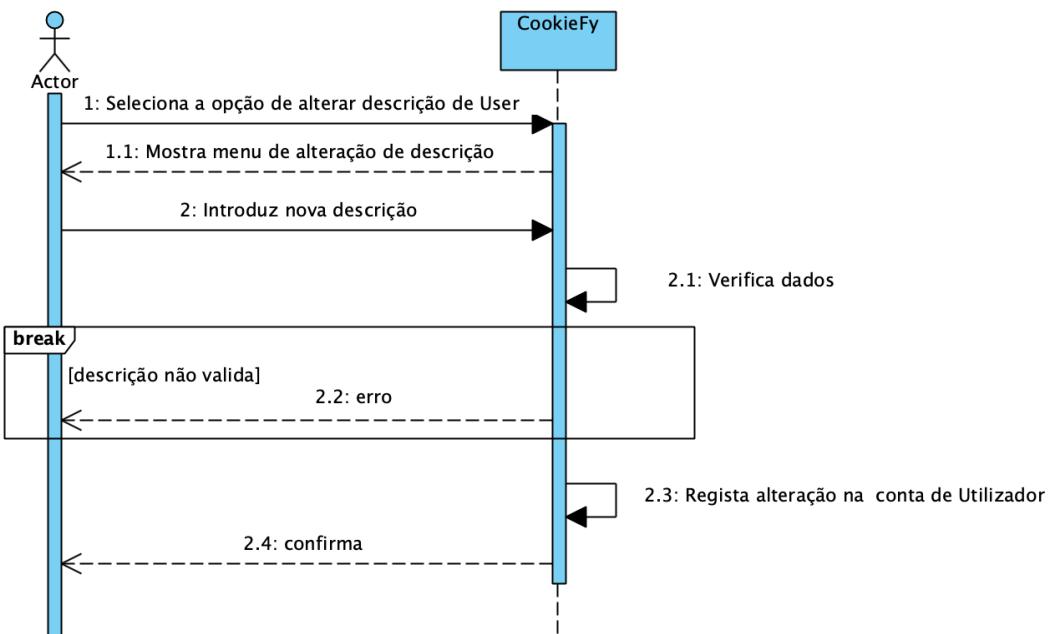
Nesta secção iremos apresentar todos os Diagramas de Sequência de Use Cases gerados diretamente a partir dos Use Cases. Estes diagramas são Diagramas de Sequência de Sistemas (DSS). Podemos assim observar todo o conjunto de operações relevantes da aplicação.

#### Gestão



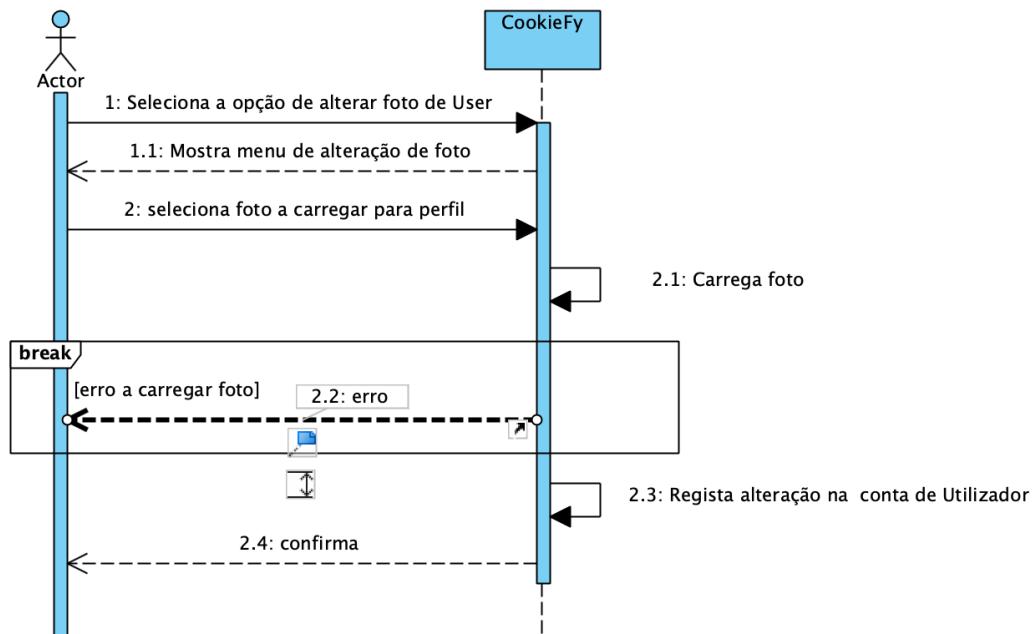
**Figura 49 - Diagrama de Sequência Adicionar CoockieCoins**

**sd Alterar Descrição**



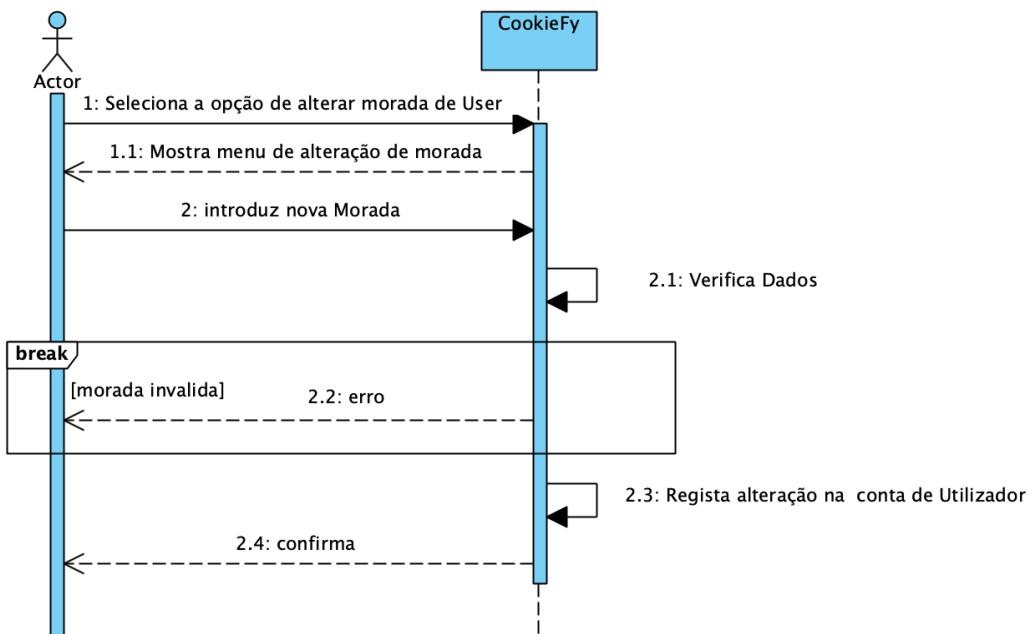
**Figura 50 - Diagrama de Sequência Alterar Descrição**

**sd Alterar Foto**



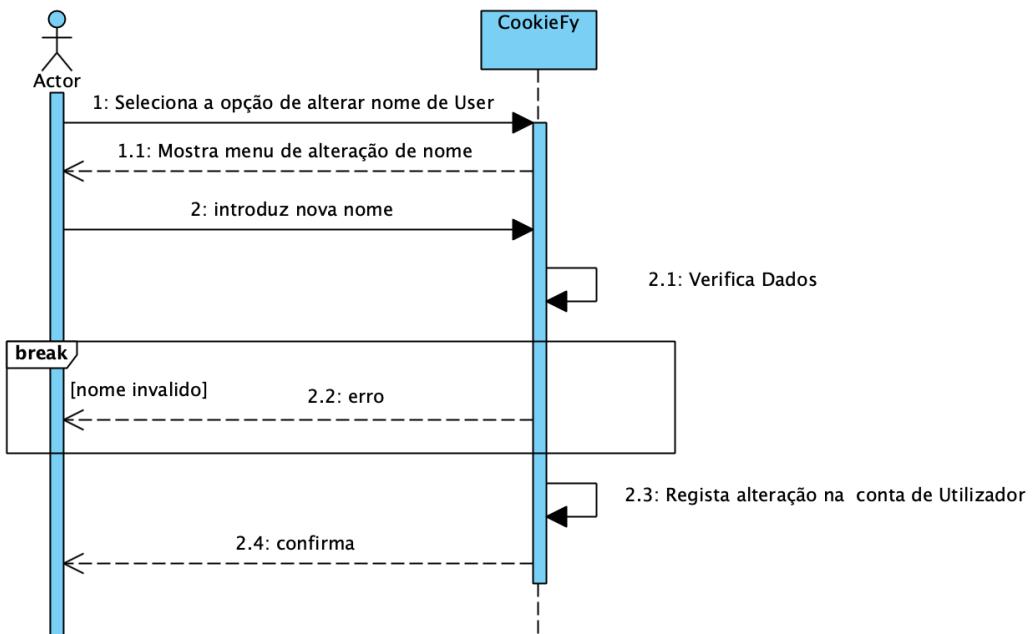
**Figura 51 - Diagrama de Sequência Alterar Foto**

**sd Alterar Morada**

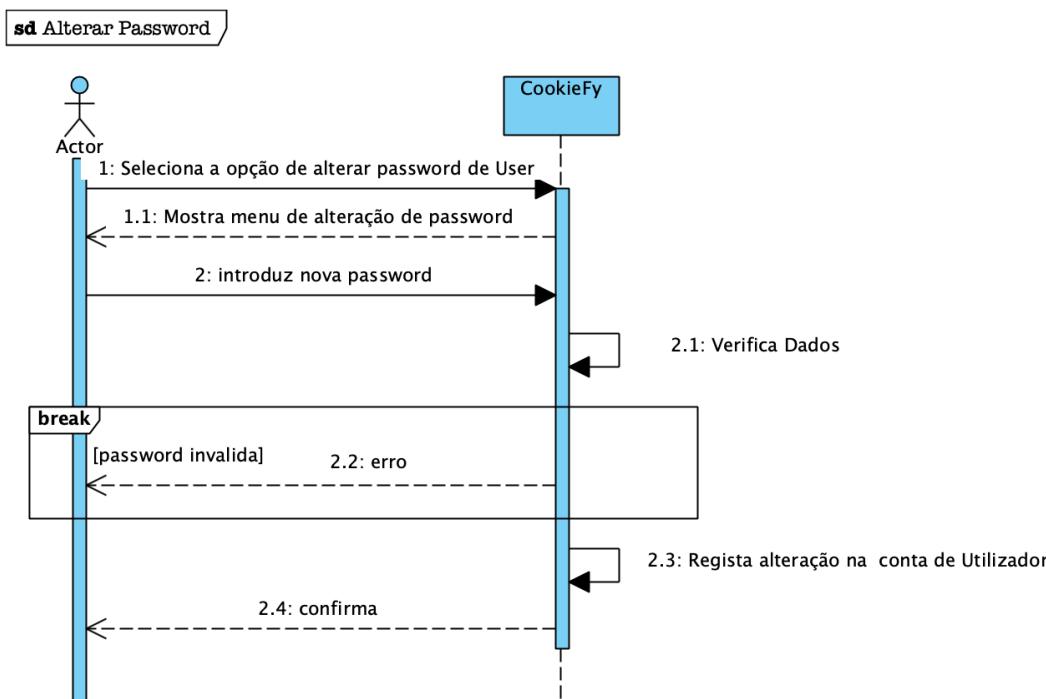
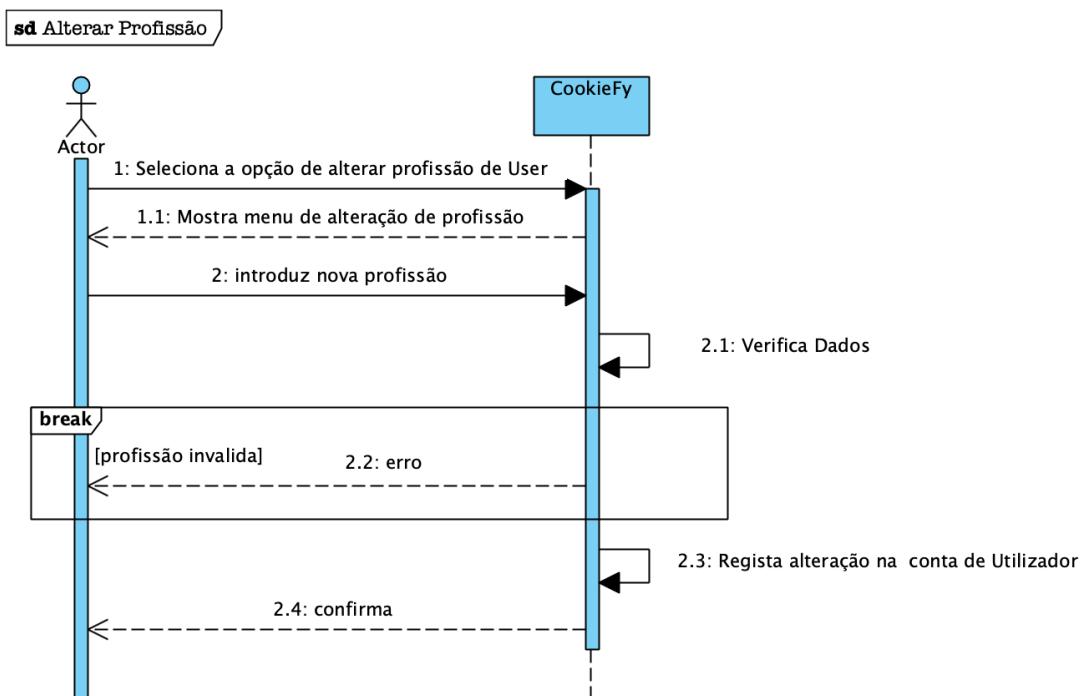


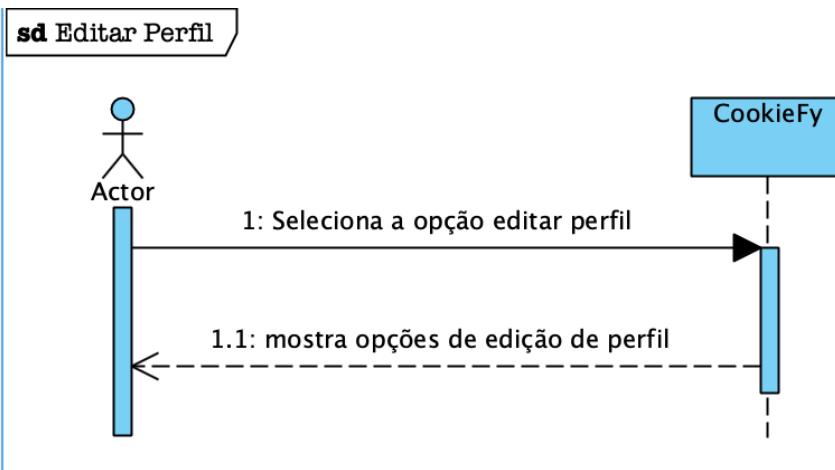
**Figura 52 - Diagrama de Sequência Alterar Morada**

**sd Alterar Nome**

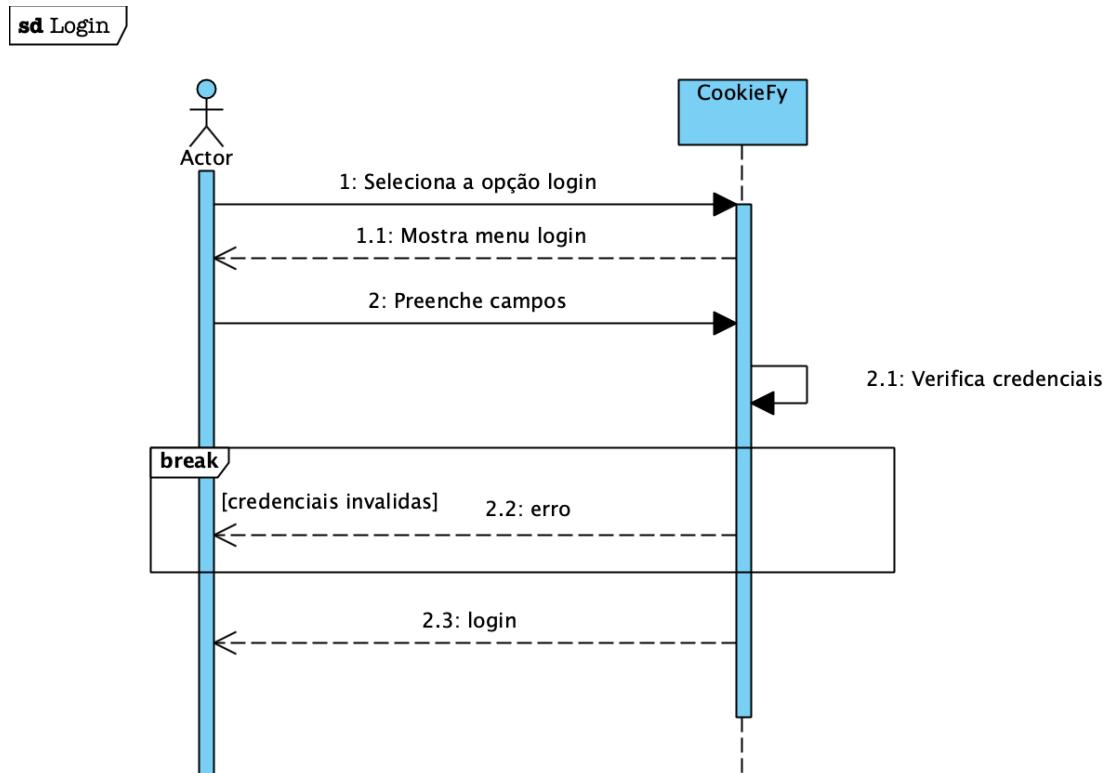


**Figura 53 - Diagrama de Sequência Alterar Nome**

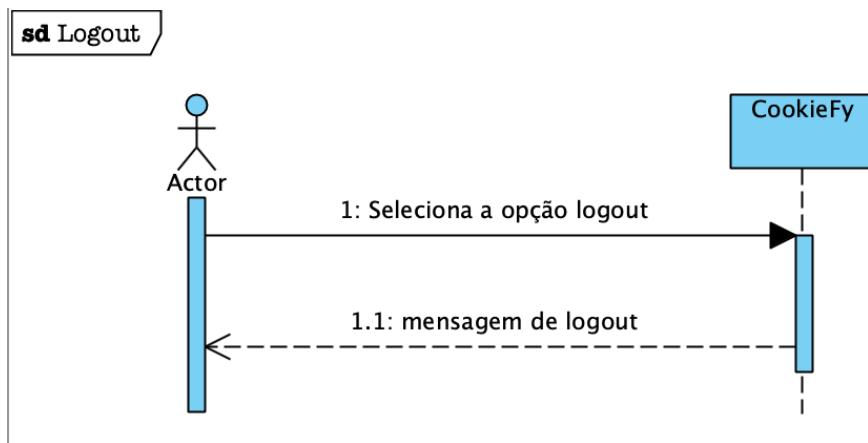
**Figura 54 - Diagrama de Sequência Alterar Password****Figura 55 - Diagrama de Sequência Alterar Profissão**



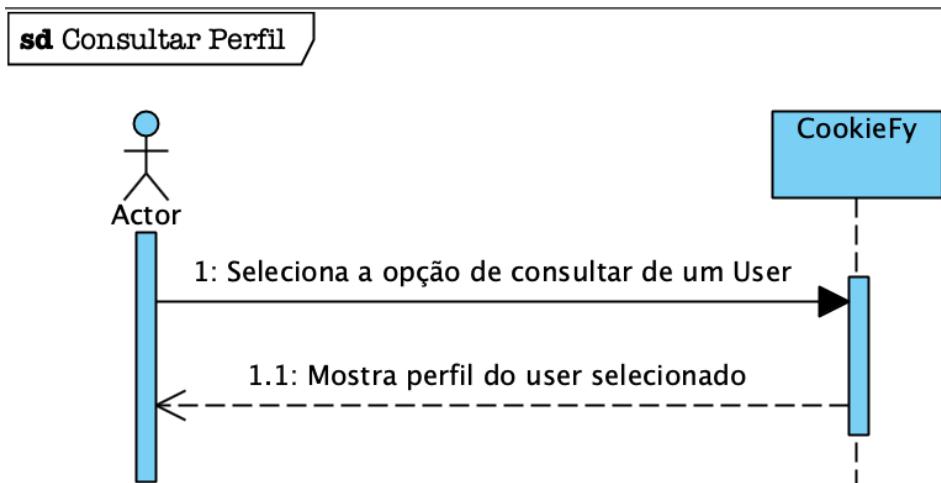
**Figura 56 - Diagrama de Sequência *Edita Perfil***



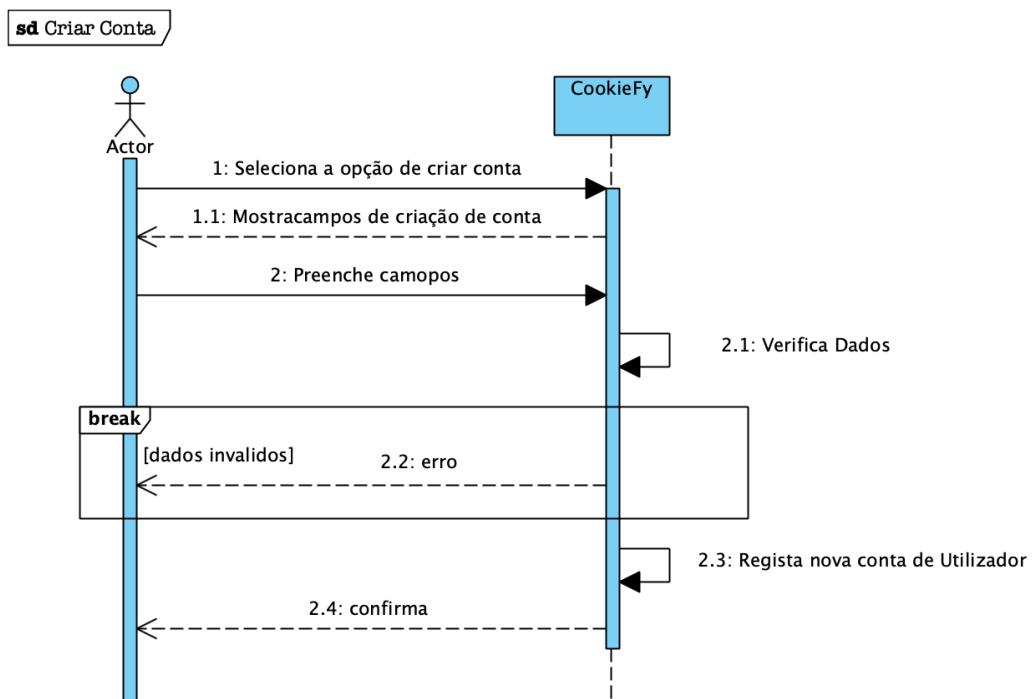
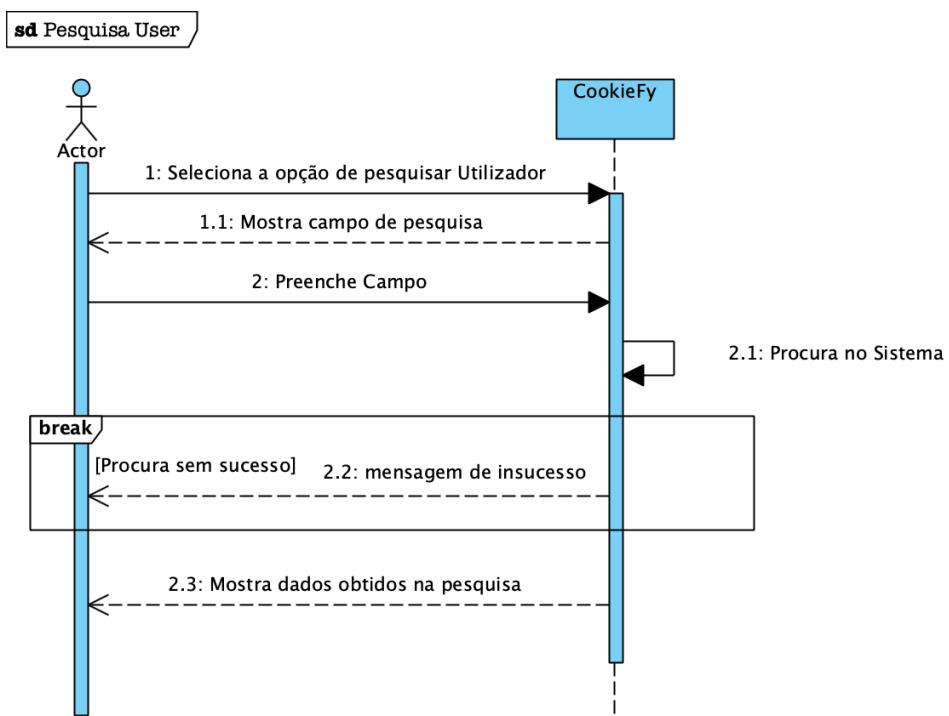
**Figura 57 - Diagrama de Sequência *Login***

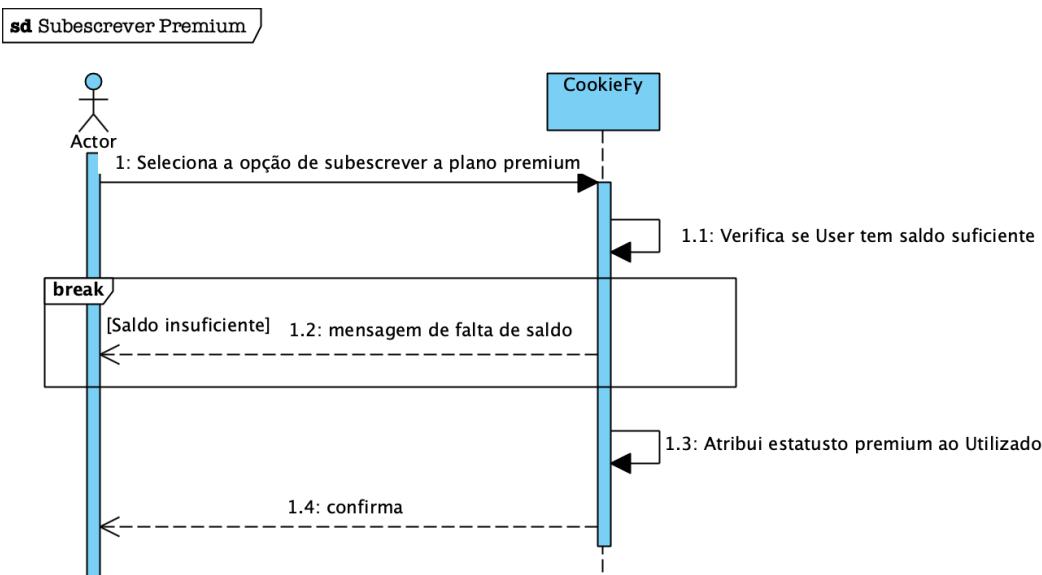


**Figura 58 - Diagrama de Sequência Logout**



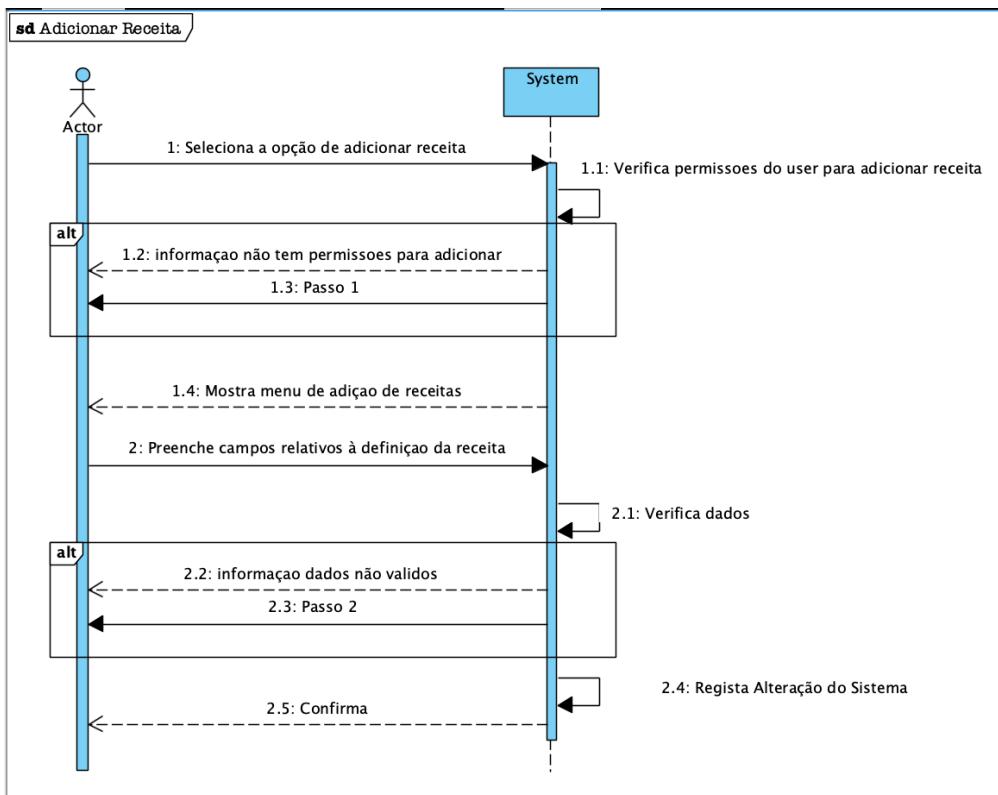
**Figura 59 - Diagrama de Sequência Consultar Perfil**

**Figura 60 - Diagrama de Sequência Criar Conta****Figura 61 - Diagrama de Sequência Pesquisa User**

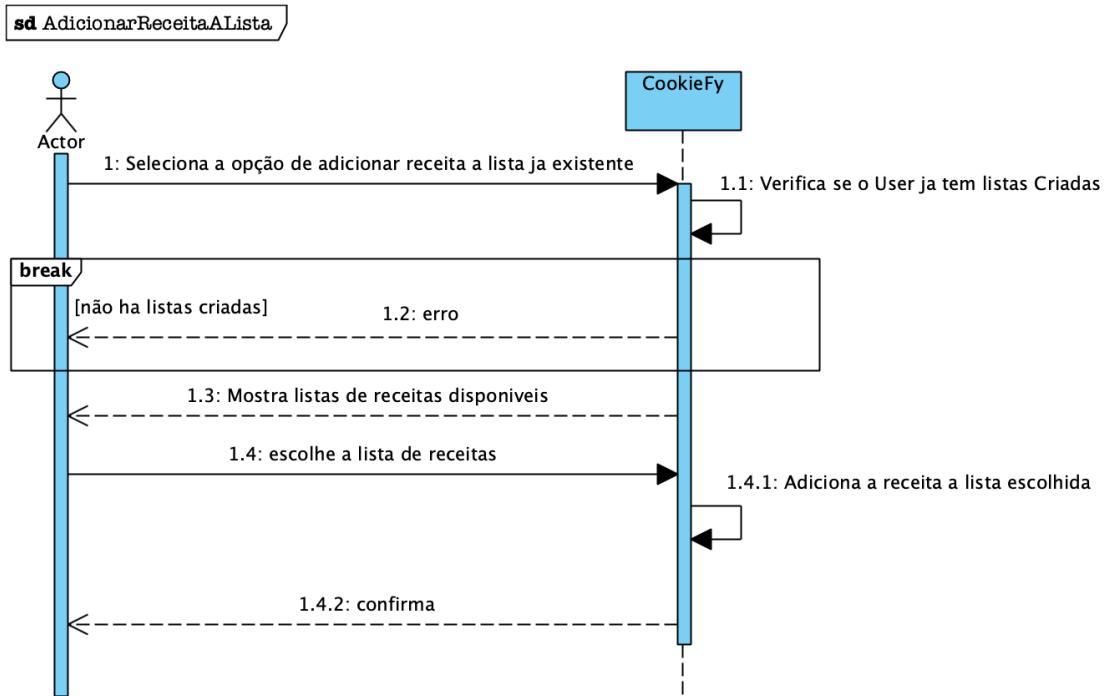


**Figura 62 - Diagrama de Sequência Subscrever Premium**

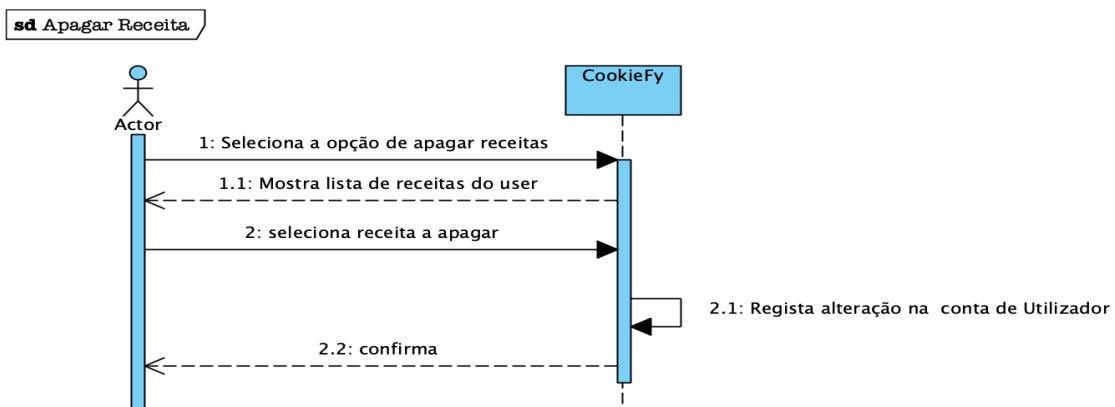
## Receita



**Figura 63 - Diagrama de Sequência Adicionar Receita**

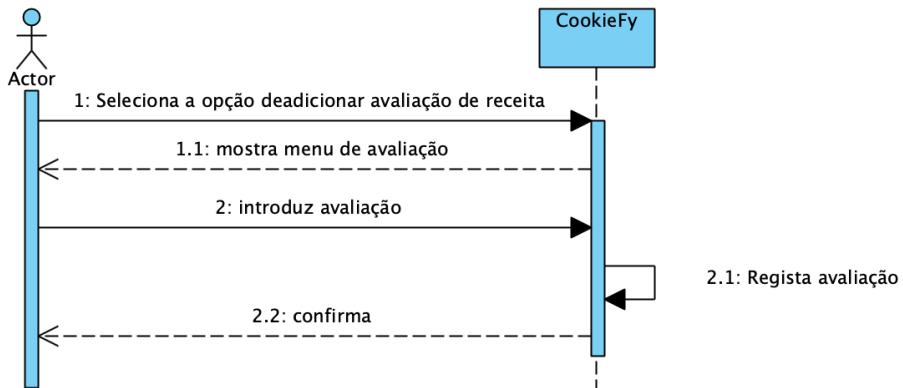


**Figura 64 - Diagrama de Sequência Adicionar Receita à Lista**



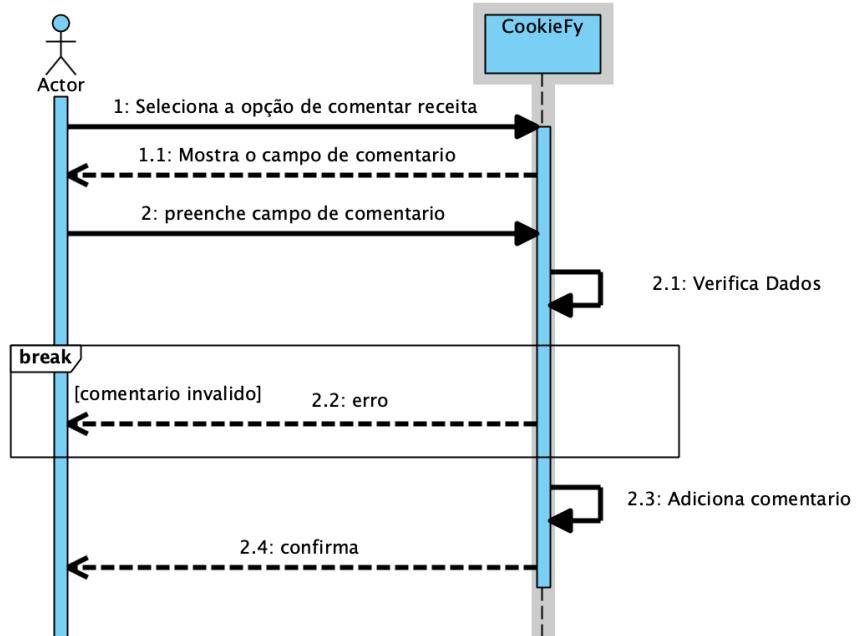
**Figura 65 - Diagrama de Sequência Receita**

**sd Avaliar receita**

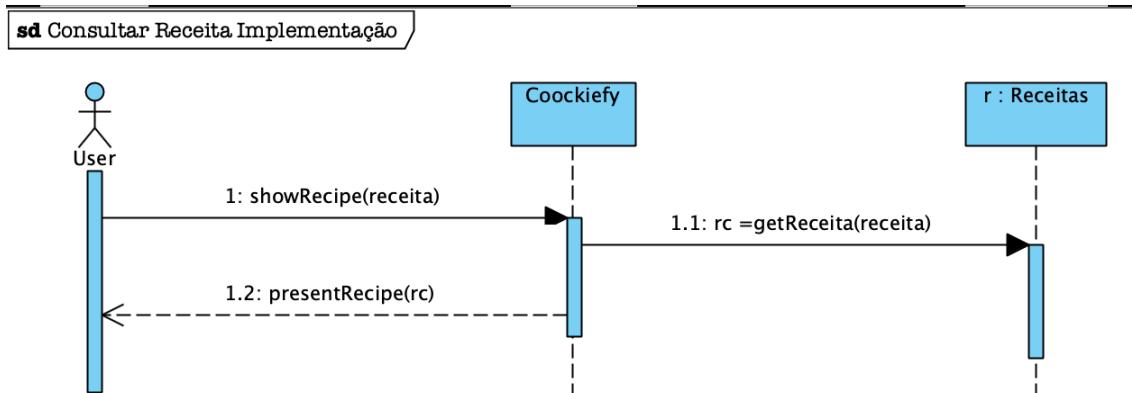


**Figura 66 - Diagrama de Sequência *Avaliar Receita***

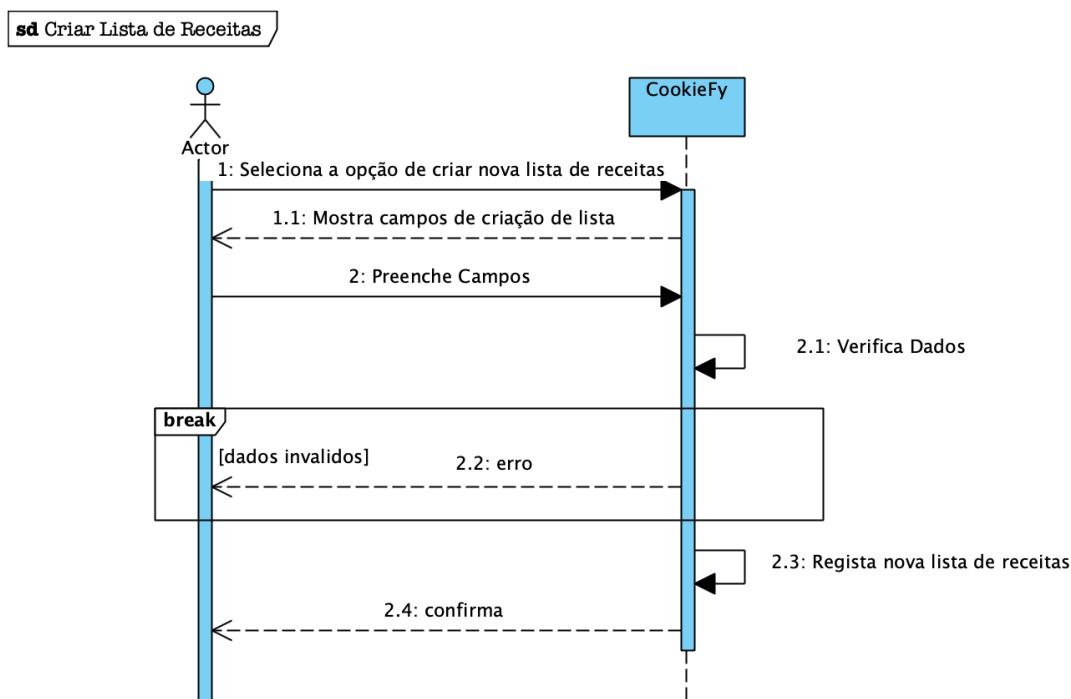
**sd Comentar Receita**



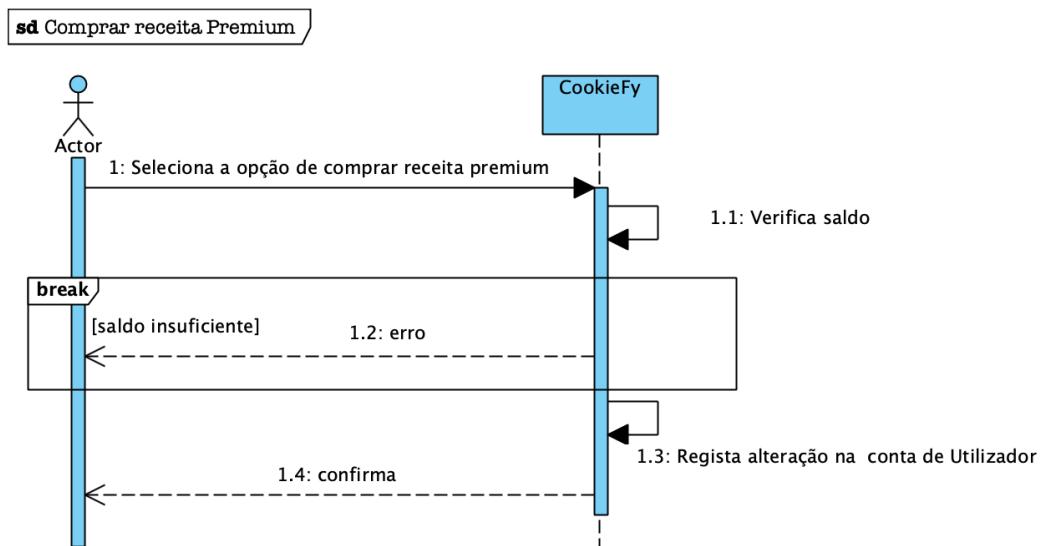
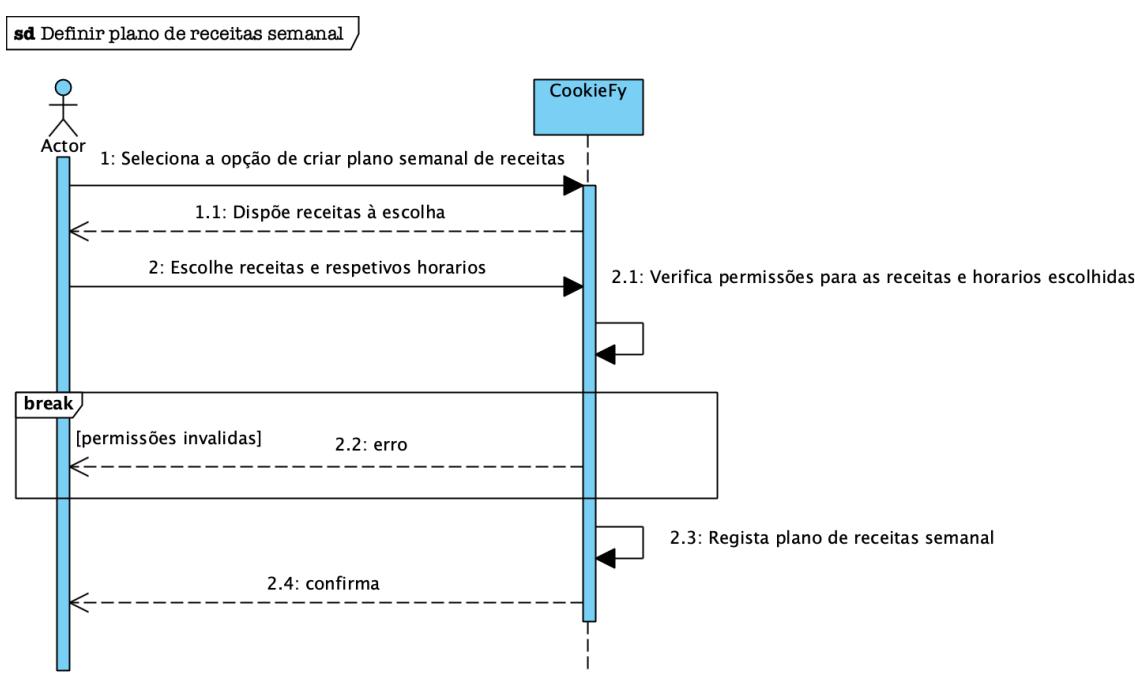
**Figura 67 - Diagrama de Sequência *Comentar Receita***

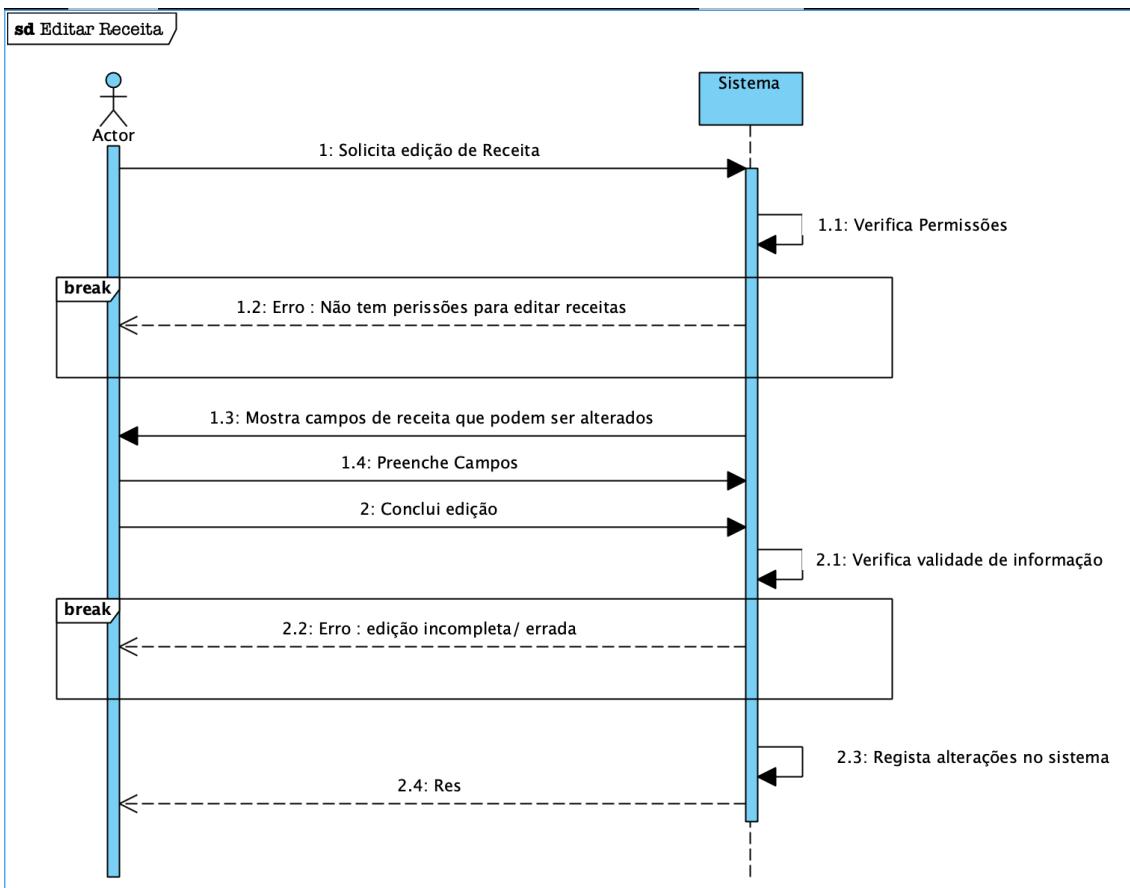
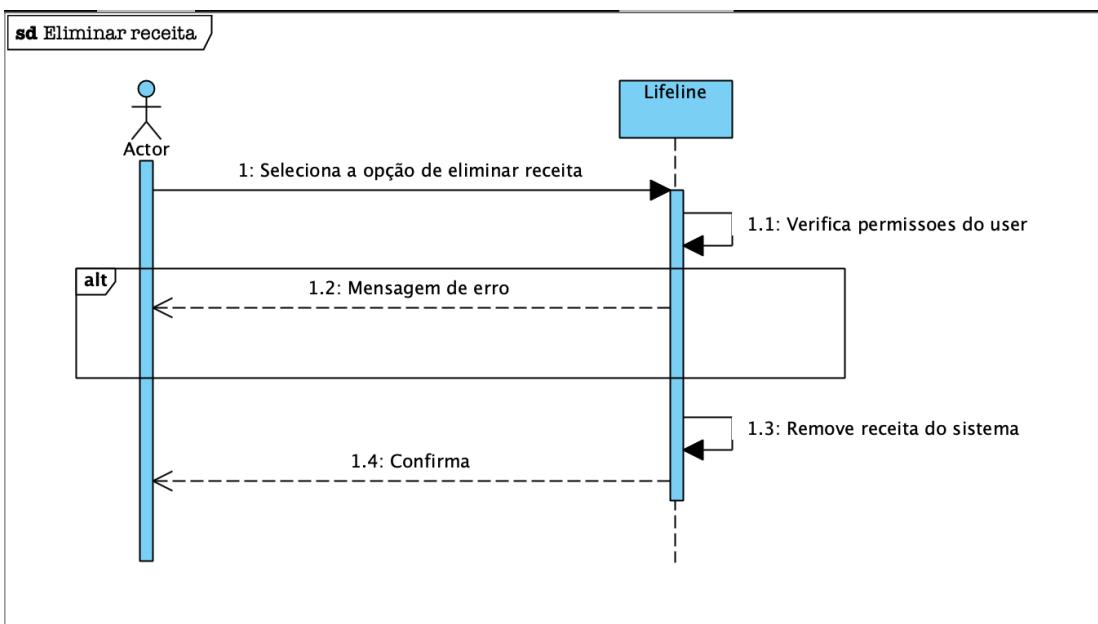


**Figura 68 - Diagrama de Sequência Consultar Receita Implementação**



**Figura 69 - Diagrama de Sequência Criar Lista Receitas**

**Figura 70 - Diagrama de Sequência Comprar Receita Premium****Figura 71 - Diagrama de Sequência Definir Plano de Receitas Semanal**

**Figura 72 - Diagrama de Sequência *Edita Receita*****Figura 73 - Diagrama de Sequência *Eliminar Receita***

**sd Juntar Receita à Lista**

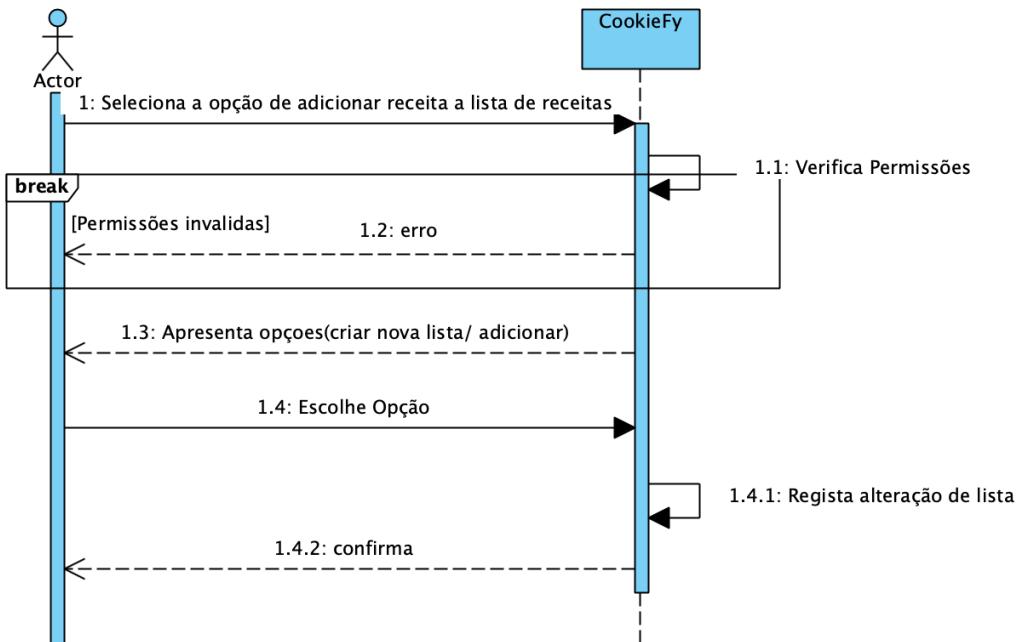


Figura 74 - Diagrama de Sequência Juntar Receita à Lista

## Gerir Música

**sd Avançar Musica**

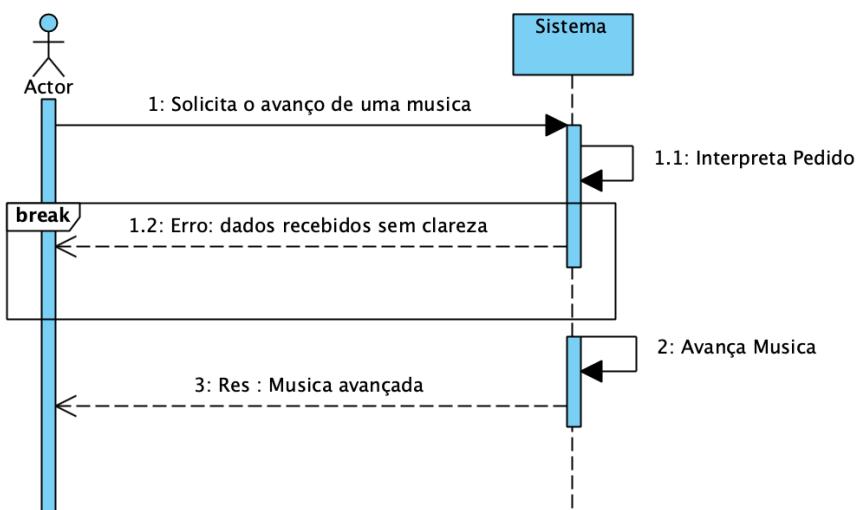
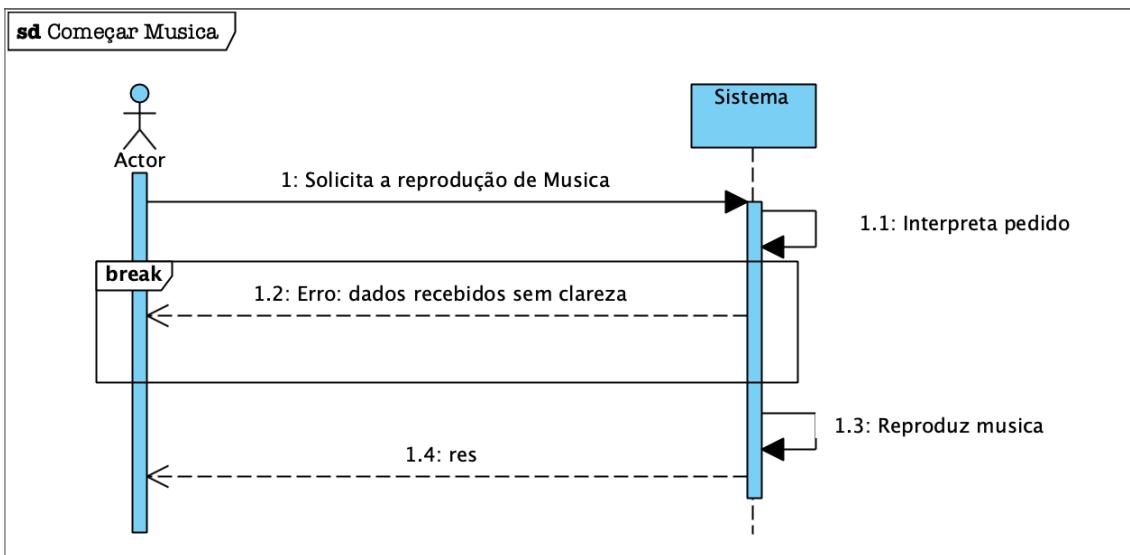
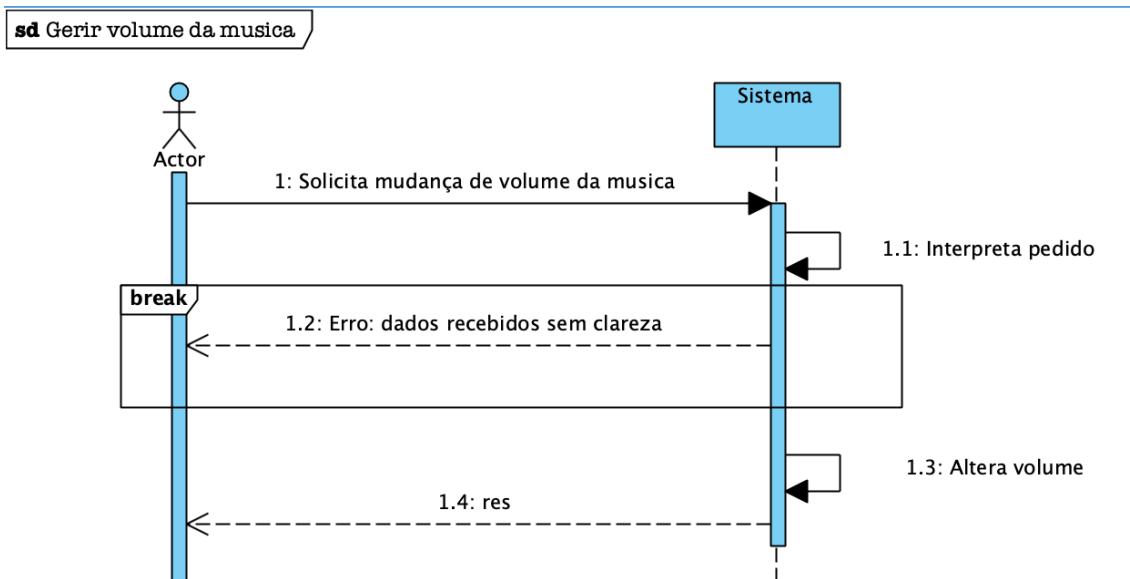


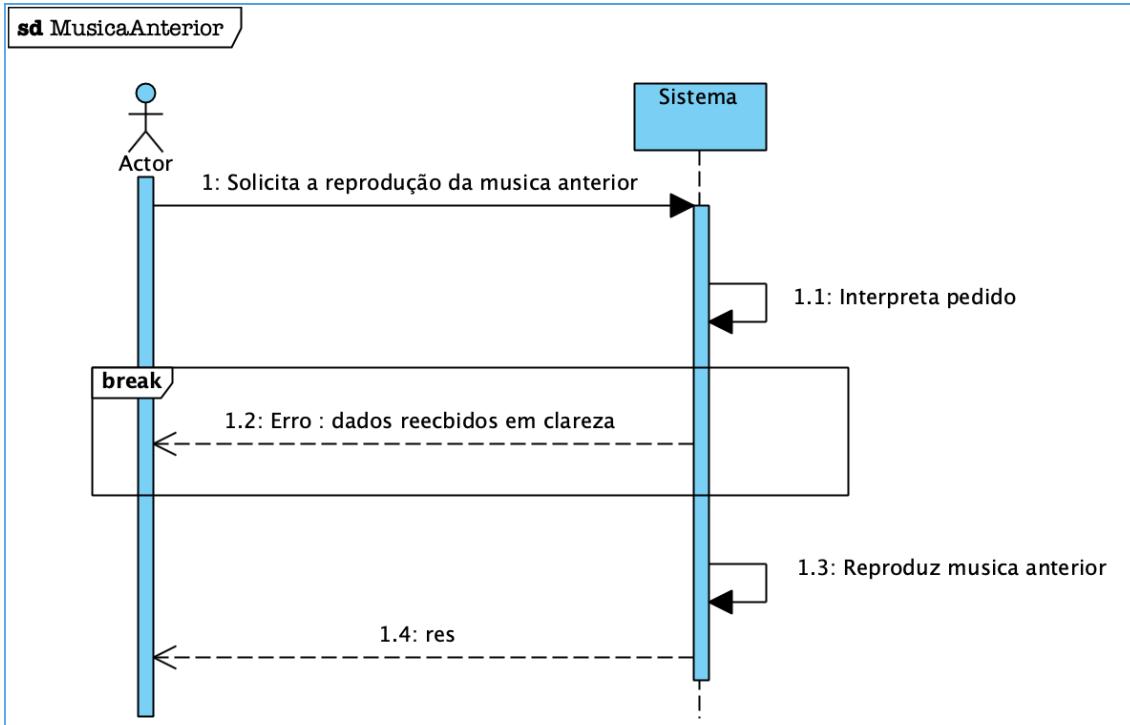
Figura 75 - Diagrama de Sequência Avançar Música



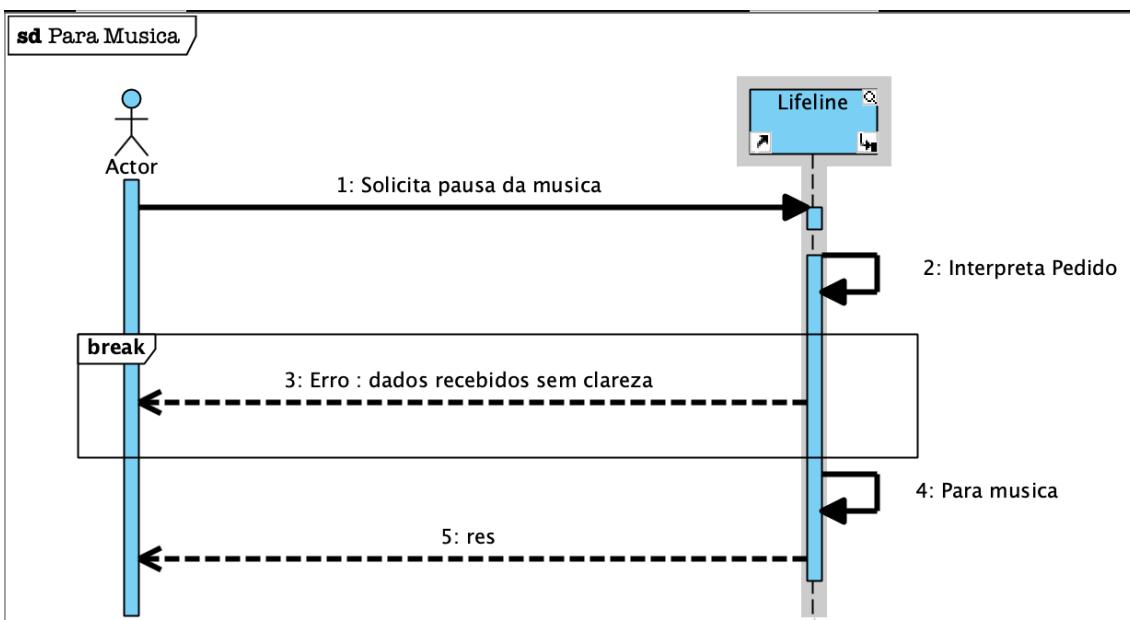
**Figura 76 - Diagrama de Sequência Começar Música**



**Figura 77 - Diagrama de Sequência Gerir Volume da Música**

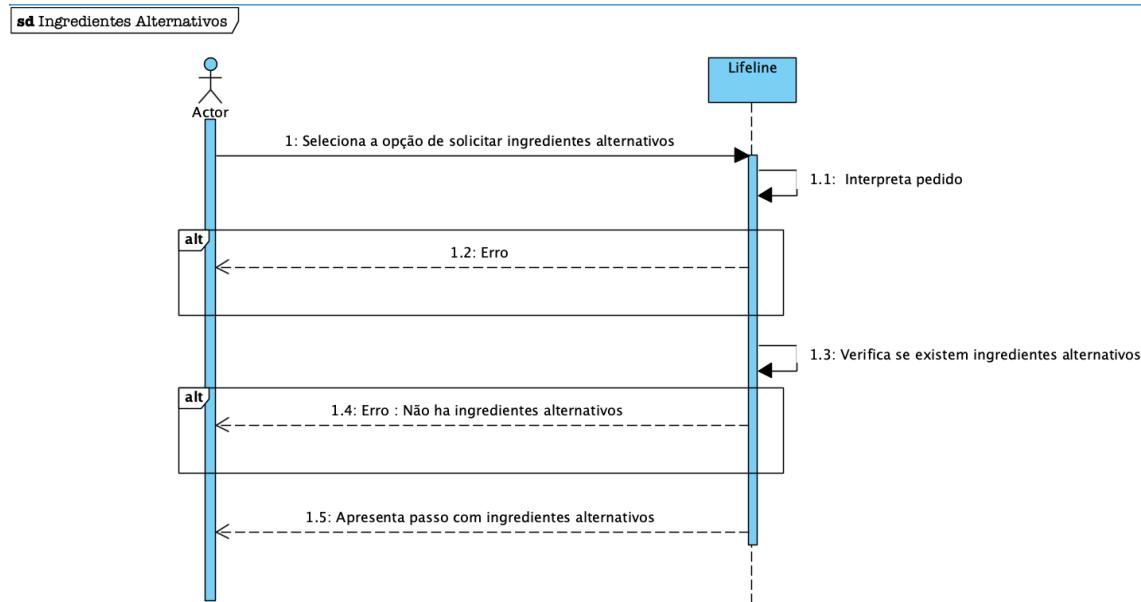


**Figura 78 - Diagrama de Sequência Música Anterior**

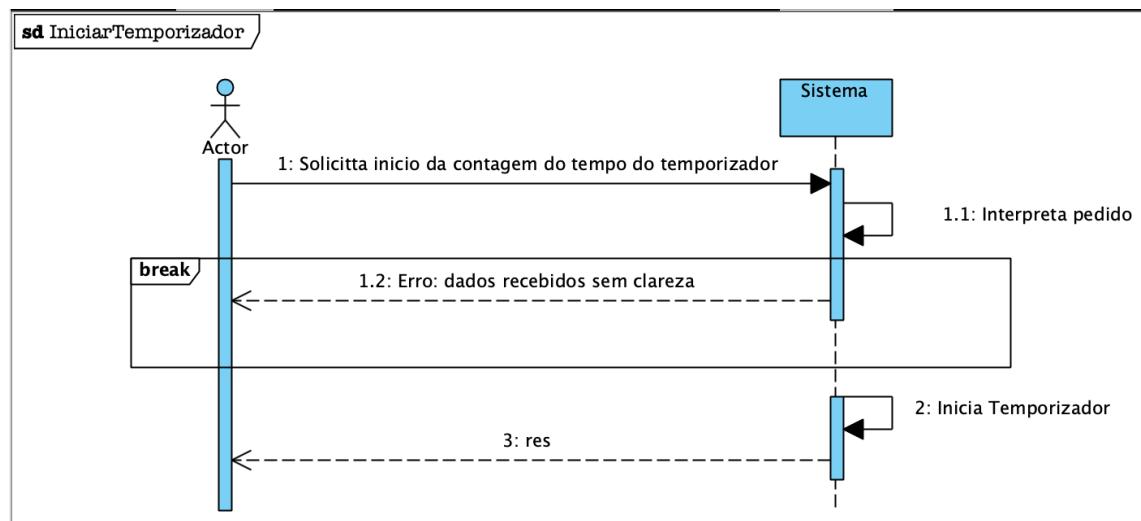


**Figura 79 - Diagrama de Sequência Para Música**

## Confeção

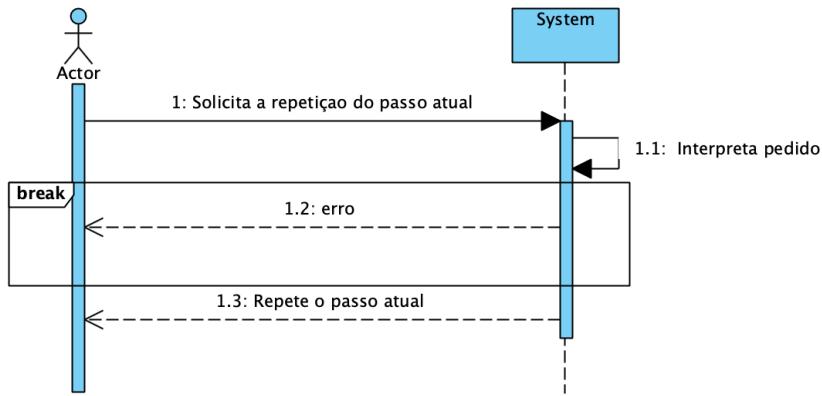


**Figura 80 - Diagrama de Sequência Ingredientes Alternativos**



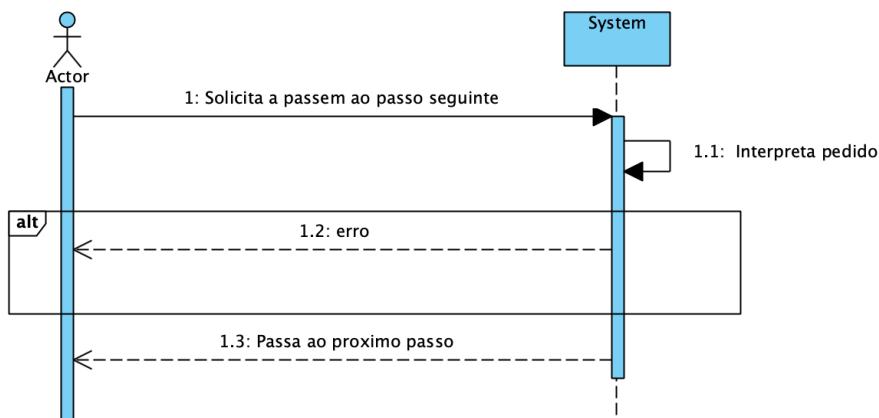
**Figura 81 - Diagrama de Sequência Iniciar Temporizador**

**sd Repetir passo Atual**



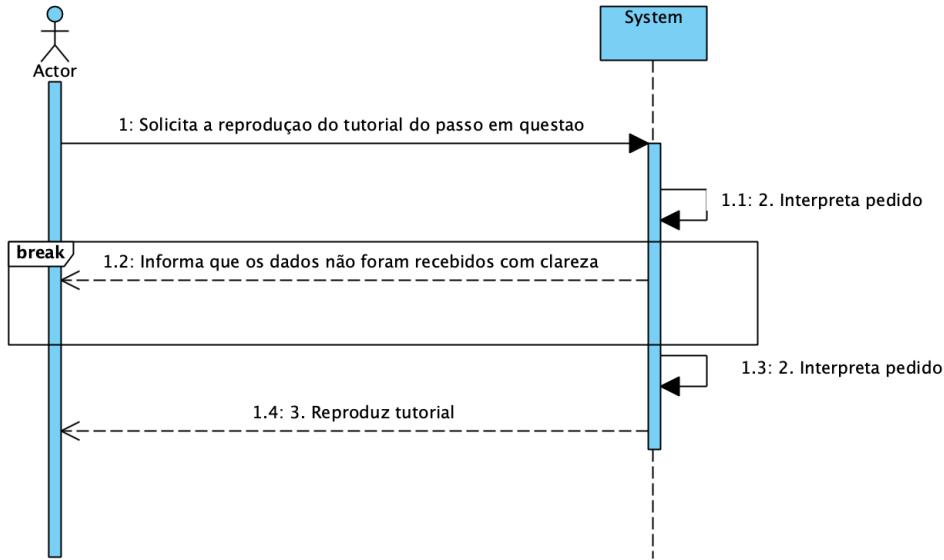
**Figura 82 - Diagrama de Sequência Repetir Passo Atual**

**sd PróximoPasso**



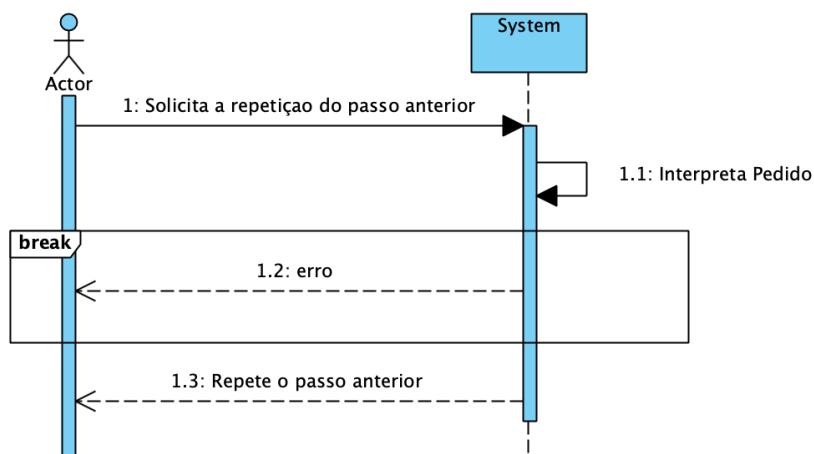
**Figura 83 - Diagrama de Sequência Próximo Passo**

**sd ReproduzirTutorial**



**Figura 84 - Diagrama de Sequência Reproduzir Tutorial**

**sd RepetirpassoAnterior**



**Figura 85 - Diagrama de Sequência Repetir Passo Anterior**

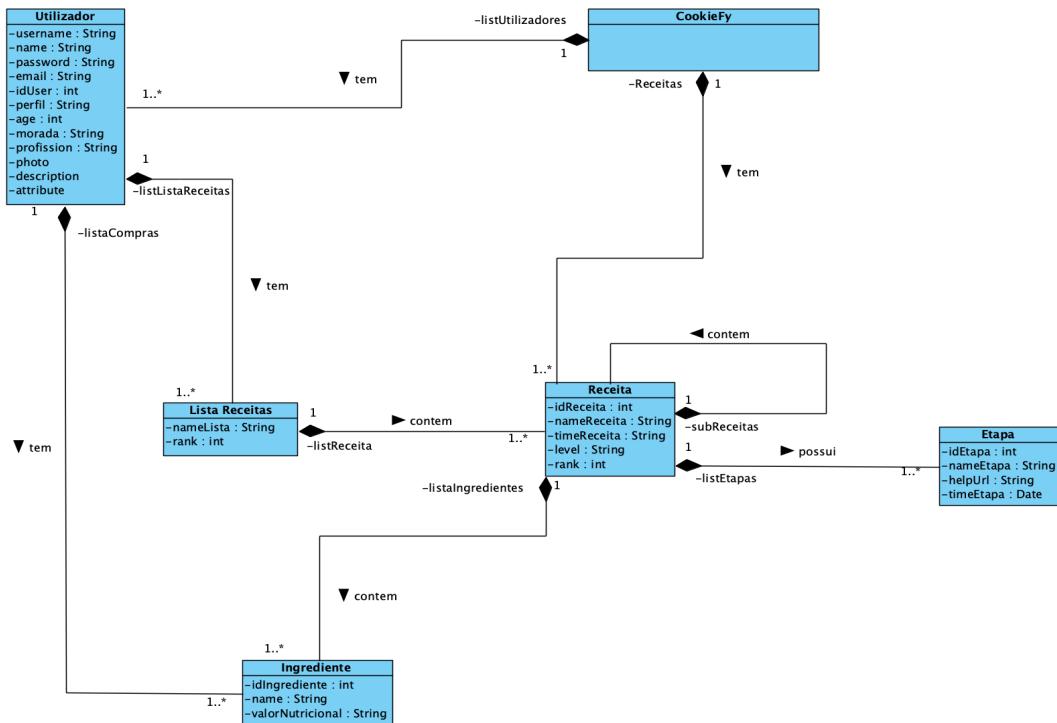
## 9.4. Diagrama de Classes

Nesta secção iremos apresentar todo o processo no desenvolvimento do Diagrama de Classes. Como ainda é a primeira vez que iremos desenvolver este diagrama para este projeto pensamos enumerar os diagramas de classes caso seja preciso fazer alguma alteração ao mesmo.

A partir do Modelo de Domínio apresentado na Figura 5, podemos identificar as principais entidades, e as potenciais classes do Projeto, iremos então aproveitar esse conhecimento e construir um Diagrama de Classes.

### 9.4.1. Diagrama de Classes V1

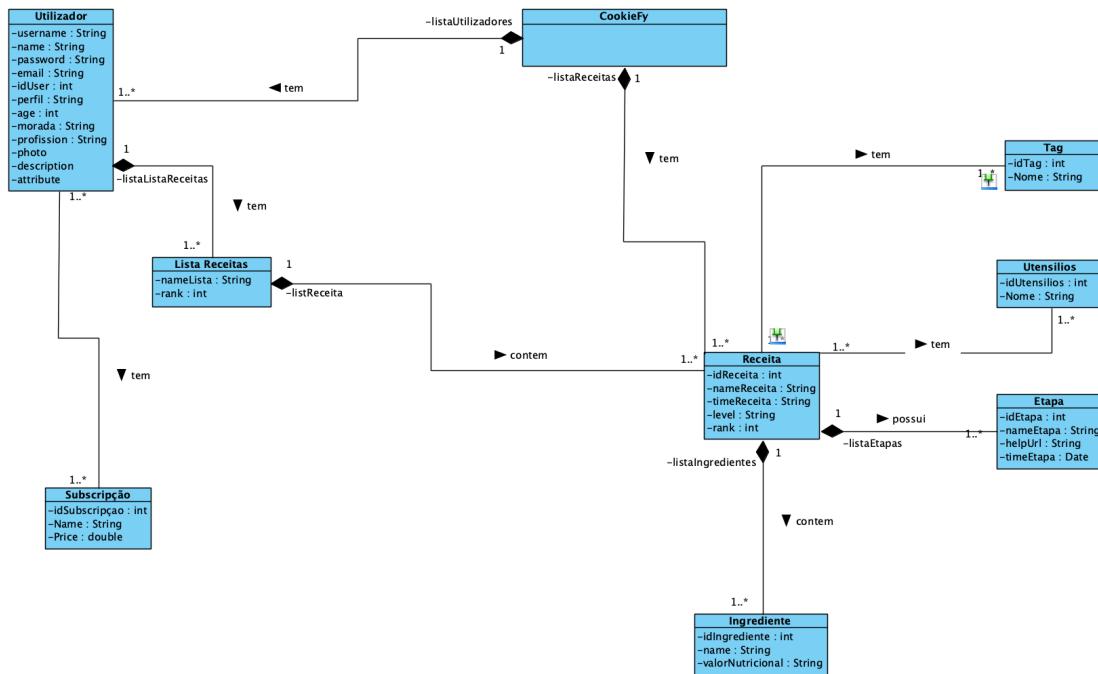
Nesta primeira versão do Diagrama de Classes foi feita uma filtragem ao Modelo de Domínio onde retiramos as potenciais classes. Colocamos as mais importantes e fizemos a relação entre elas. Neste Diagrama ainda falta acrescentar os métodos, visto que ainda é uma fase de desenvolvimento.



**Figura 86 - Diagrama de Classes V1.0**

## 9.4.2. Diagrama de Classes V2

Este diagrama de classes já é mais completo que o V1, pois já foi feita uma filtragem em conjunto com o modelo conceptual bem como da implementação que irá ser feita.



**Figura 87 - Diagrama de Classes V2.0**

## 10. Base de dados

### 10.1. Modelo Conceptual

#### 10.1.1. Identificar os tipos de identidades

Após leitura, interpretação e avaliação dos fatores importantes ao projeto identificamos os termos que seriam definidos como objetos chave. Entre estes selecionamos como mais importantes: *User*, *RecipeList*, *Recipe* e *Ingredients*.

Nome da Entidade	Descrição	Sinónimos	Ocorrências
<b>User</b>	Utilizador que pretende utilizar a aplicação, para visualizar ou partilhar receitas	Utilizador, Cliente, Chef	Cada utilizador tem uma subscrição, zero ou mais seguidores e comentários. Tem ainda zero ou mais receitas, planos e listas de receitas criadas ou seguidas.
<b>Subscription</b>	Tipo de subscrição do utilizador na aplicação	Subscrição	
<b>Recipe</b>	Receitas que irão ser o centro da aplicação	Receita	Cada receita tem uma lista de passos para a sua realização, os utensílios e ingredientes necessários, o seu tipo e zero ou mais tags
<b>RecipeList</b>	Listas de receitas	Lista de receitas	Cada lista de receitas tem uma ou mais receitas
<b>Step</b>	Passos necessários para confeccionar uma receita	Passo	Cada passo tem uma descrição, um tempo, uma posição que define a ordem do mesmo numa receita e pode ter um link para um tutorial
<b>Utensils</b>	Utensílios necessários para confeccionar uma receita	Utensílios	
<b>Ingredients</b>	Ingredientes necessários para confeccionar uma receita	Ingredientes	
<b>Tag</b>	Tag das receitas		
<b>Comment</b>	Comentários dos utilizadores a receitas	Comentarios	Cada comentário tem um id do utilizador que o realiza e da receita que está a comentar

**Figura 88 - Tabela - Tipo de Entidades**

### 10.1.2. Identificar tipos de relacionamento

Uma vez que definimos as entidades sobre as quais o sistema se elabora, torna-se essencial estabelecer os métodos de relação entre as mesmas. Deste modo, através de nova leitura e interpretação dos requisitos, identificamos os relacionamentos entre entidades bem como a respetiva multiplicidade, os quais apresentamos na tabela seguinte:

Nome da entidade	Multiplicidade	Relacionamento	Multiplicidade	Nome da entidade
User	1	Follow	N	User
User	N	Possui	N	Subscription
User	N	RecipePlan	N	Recipe
User	1	Contem	N	RecipeList
User	1	Possui	N	Recipe
RecipeList	N	RecipeListRecipe	N	Recipe
Recipe	N	RecipeIngredient	N	Ingredients
Recipe	N	RecipeUtensils	N	Utensils
Recipe	N	RecipeTag	N	Tag
Recipe	N	SubRecipe	N	Recipe
Recipe	1	Possui	N	Step

**Figura 89 - Tabela - Tipo de Relacionamento**

### 10.1.3. Identificar e associar atributos com os tipos de entidades e relacionamentos

Devido há existência um conjunto de informações importantes a reter pertinentes a cada uma das entidades surge a seguinte tabela na qual evidenciamos todos os atributos retidos em cada uma das entidades, que completam os dados essenciais à implementação destas e do projeto.

Nome da entidade	Atributo	Descrição	Tipo	Null	Multivalor
<b>User</b>	idUser	Identificador	Valor Inteiro Positivo	Não	Não
	Name	Nome	Caracteres	Não	Não
	Proffession	Profrissão	Caracteres	Não	Não
	Country	País	Caracteres	Não	Não
	City	Cidade	Caracteres	Não	Não
	Balance	Balanço	Valor decimal positivo	Não	Não
	Email	Email	Caracteres	Não	Não
	Password	Password	Caracteres	Não	Não
	ImgPath	Caminho para a imagem de perfil	Caracteres	Sim	Não
<b>Subscription</b>	Description	Descrição	Caracteres	Não	Não
	idSubscription	Identificador	Valor inteiro positivo	Não	Não
	Name	Nome	Caracteres	Não	Não
	Price	Preço	Valor decimal positivo	Não	Não

<b>RecipeList</b>	<b>idRecipeList</b>	Identificador	Valor inteiro positivo	Não	Não
	<b>Name</b>	Nome	Caracteres	Não	Não
	<b>Classifications</b>	Classificações	Valor inteiro positivo	Não	Não
	<b>Rank</b>	Avaliação	Valor decimal positivo	Não	Não
<b>Recipe</b>	<b>idRecipe</b>	Identificador	Valor inteiro positivo	Não	Não
	<b>Title</b>	Título	Caracteres	Não	Não
	<b>Description</b>	Descrição	Caracteres	Não	Não
	<b>Rank</b>	Avaliação	Valor decimal positivo	Não	Não
	<b>Classifications</b>	Classificações	Valor inteiro positivo	Não	Não
	<b>Date</b>	Data	DATETIME	Não	Não
	<b>Update_at</b>	Ultima actualização	DATETIME	Não	Não
	<b>Time</b>	Tempo de preparação	TIME	Não	Não
	<b>Tag</b>	Identificador	Valor inteiro positivo	Não	Não
<b>Ingredients</b>	<b>idIngredients</b>	Identificador	Valor inteiro positivo	Não	Não
	<b>Name</b>	Nome	Caracteres	Não	Não
<b>Utensils</b>	<b>idUtensils</b>	Identificador	Valor inteiro positivo	Não	Não
	<b>Name</b>	Nome	Caracteres	Não	Não
<b>Step</b>	<b>idStep</b>	Identificador	Valor inteiro positivo	Não	Não
	<b>Position</b>	Posição	Valor inteiro positivo	Não	Não
	<b>Time</b>	Tempo do passo	TIME	Não	Não
	<b>Text</b>	Texto	Caracteres	Não	Não
	<b>Link</b>	Link para tutorial	Caracteres	Sim	Não
<b>Comment</b>	<b>idComment</b>	Identificador	Valor Inteiro Positivo	Não	Não

**Figura 90 - Tabela - Entidade/Atributos/Relacionamentos**

#### 10.1.4. Atributos simples/compostos

Através de avaliação dos requisitos, concluímos que não existe a necessidade de implementarmos atributos compostos.

#### 10.1.5. Atributos derivados

Em seguida à identificação dos atributos, foi possível inferir que não haveria também necessidade de atributos derivados.

#### 10.1.6. Atributos multivalor

Do mesmo modo referido nos pontos anteriores em que concluímos não serem necessários, é possível concluir também que não existem atributos multivalor necessários ao bom desenvolvimento e funcionamento da base de dados.

#### 10.1.7. Associação entre atributos e entidades

Apresentamos, de seguida, a relação de atributos e respetivas entidades que decidimos implementar.

#### 10.1.8. Determinar domínio dos atributos

Neste ponto vamos descrever os domínios dos atributos das entidades principais ao projeto. Onde explicaremos o tipo e os valores que os atributos tomam, tal como a razão para essas decisões.

##### User

- IdUser: INT , identifica o Utilizador de modo a diferenciar entre todos os users por questões de eficiência e consistência.
- Name: STRING, onde estará contido o nome do cliente.
- Profession: STRING , identifica a profissão de origem do User .
- Country: STRING , identifica o país de origem do User .
- City: STRING , identifica a cidade de origem do User .
- Balance: FLOAT, guarda o valor do saldo do User na aplicação.
- Email: STRING, que irá conter o email de registo do cliente.
- Password: STRING, que conterá a password do cliente. Uma vez que não é necessário elevado segurança, esta informação ficará guardada na mesma Base de Dados que todos os outros dados.
- ImgPath: VARBINARY que guardará a imagem do utilizador. É guardado diretamente na BD pois a aplicação é de pequena dimensão e diminui a carga de implementação.

- Description: VARCHAR, breve descrição que o User escolhe colocar sobre si.

#### Subscription

- IdSubscription: INT , identificador de cada subscrição
- Name : STRING , nome da subscrição
- Price: FLOAT , preço da subscrição

#### Recipe

- IdRecipe: INT, identificador de cada receita
- Title: VARCHAR , nome da receita
- Description: VARCHAR , breve descrição da receita
- Rank: INT , rank médio das avaliações da receita
- Classifications: INT , numero de avaliações realizadas pelos users
- Date: DATETIME , data a que foi criada
- Update\_at: DATETIME , data do update mais recente da receita
- Time: TIME , tempo médio de fazer a receita

#### RecipeList

- IdRecipeList: INT , identificador de cada uma das listas de receitas.
- Rank : INT , rank médio das avaliações da lista de receitas
- Classifications: INT , numero de avaliações realizadas pelos users
- Name: STRING , nome da receita

#### Ingredient

- IdIngredients : INT , identificador de cada ingrediente
- Name : STRING , nome do ingrediente

#### Tag

- IdTag : INT , identificador de cada Tag
- Tag : STRING , nome da Tag

#### Utensils

- IdSubscription : INT , identificador de cada Utensílio
- Name : STRING , nome do Utensílio

#### Step

- IdStep : INT , identificador de cada Step
- Position : INT , numero da posição do step na receita
- Time: TIME , tempo médio de realização do step
- Help\_URL: STRING, link para vídeo de ajuda a um método de cozinha

### **10.1.9. Determinar chaves primárias, candidatas e alternativas**

Para identificarmos correta e unicamente cada uma das distintas ocorrências das entidades, recorremos a atributos chave, que servem de métodos de diferenciação das entidades.

Durante o processo de desenvolvimento optamos por chaves primárias:

- User: IdUser
- Recipe: IdRecipe
- RecipeList: IDRecipeList
- Subscription: IdSubscription
- RecipeList: IdStep
- Ingredients: IdIngredients
- Utensils: IdUtensils
- Tag: IdTag

### 10.1.10. Desenho do diagrama ER

Apresentamos, o desenho de diagrama E-R (Entidade-Relacionamento) tal que possamos representar conceptualmente as relações entre as entidades presentes na base de dados.

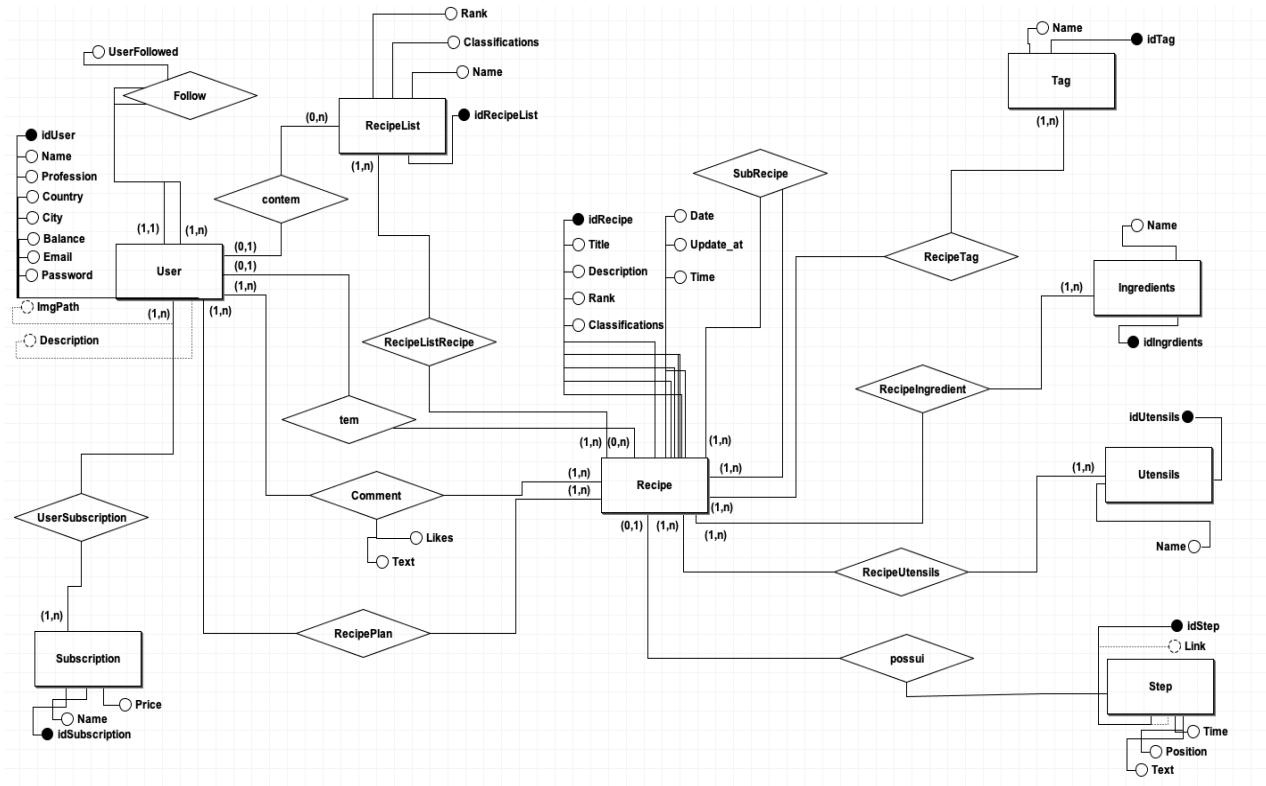


Figura 91 - Modelo Conceptual

### 10.1.11. Revisão do modelo de dados com o cliente

Após a conclusão do modelo conceptual, voltamos a revê-lo e depois dispusemo-lo para avaliação do cliente. Nenhum problema foi levantado sendo o modelo de dados aceite, tal que seguimos para a etapa seguinte do desenvolvimento do projeto.

## **10.2. Modelo Lógico**

### **10.2.1. Construção e validação do modelo de dados lógicos**

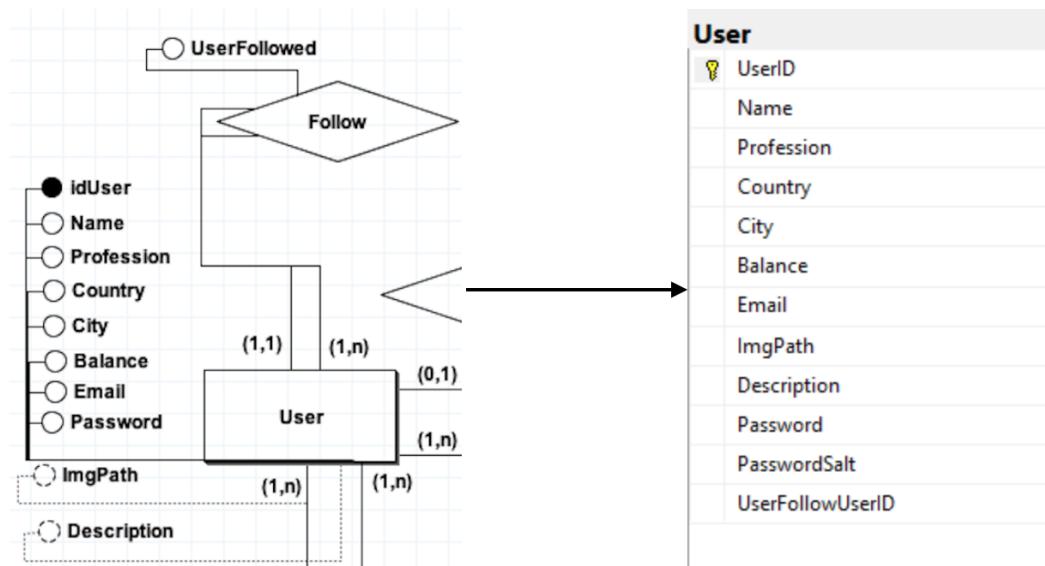
Nesta parte reproduzir-se-á o modelo lógico com base no modelo conceitual anteriormente estabelecido. Por isso, derivam-se os relacionamentos no modelo lógico criando tabelas ou relações a fim de se representar as entidades, os atributos e os relacionamentos que foram identificados.

#### **10.2.1.1. Entidades Fortes**

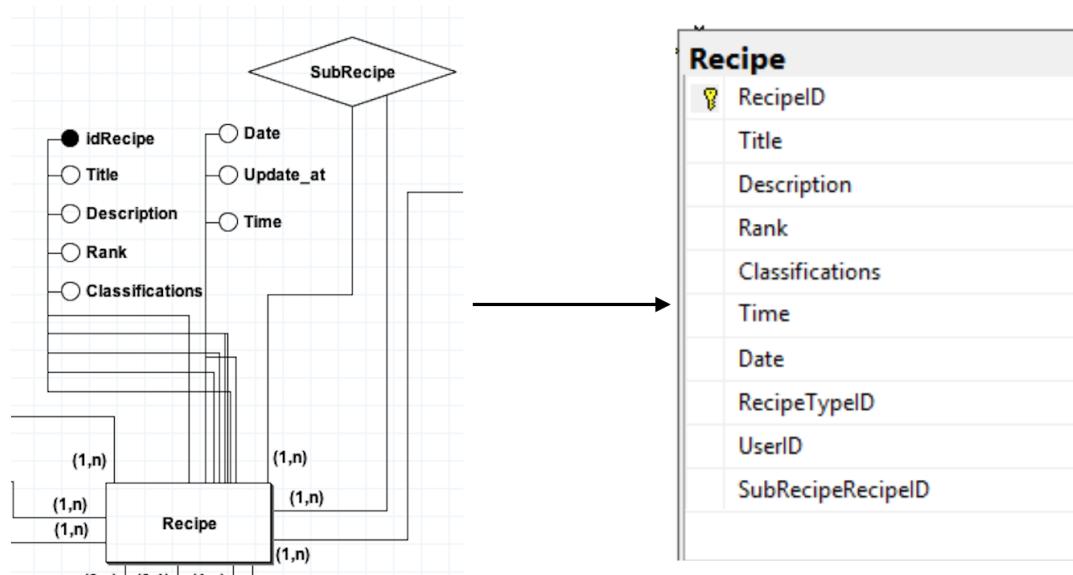
Na modelação conceitual realizada estão presentes as seguintes entidades:

- User
- Recipe
- Type
- Step
- Utensils
- Ingredients
- Tag
- RecipeList
- Subscription

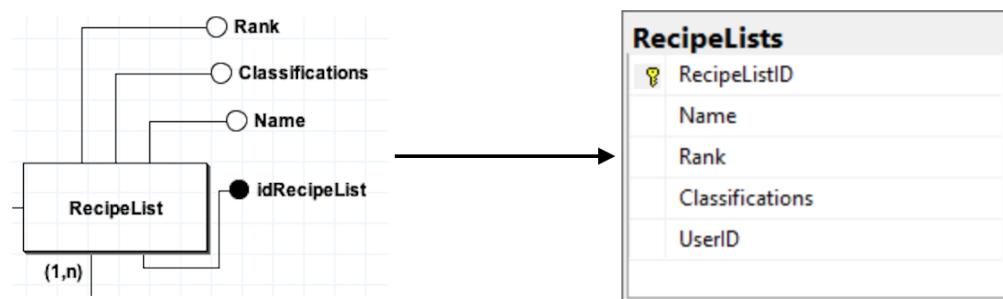
Como tal, por serem independentes umas das outras, são então também entidades fortes. Por isso, para cada uma delas, foi criada uma tabela no modelo lógico, com os seus atributos.



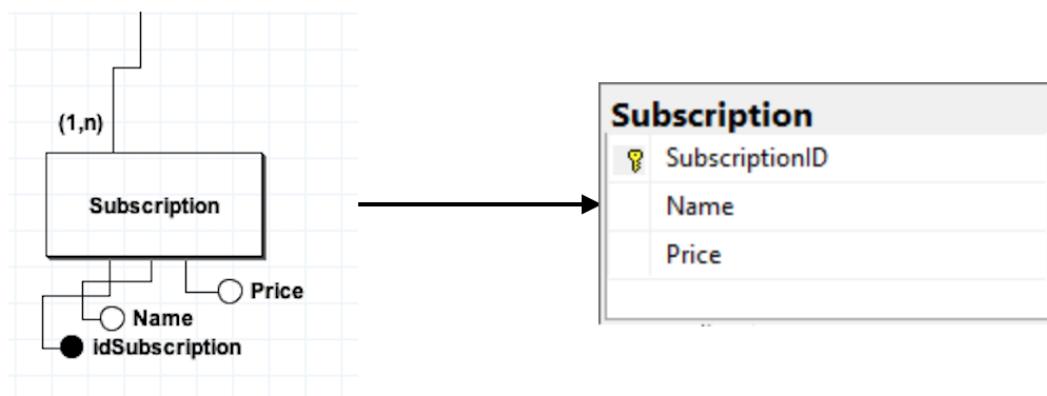
**Figura 92 - Conceptual para lógico (Entidade User)**



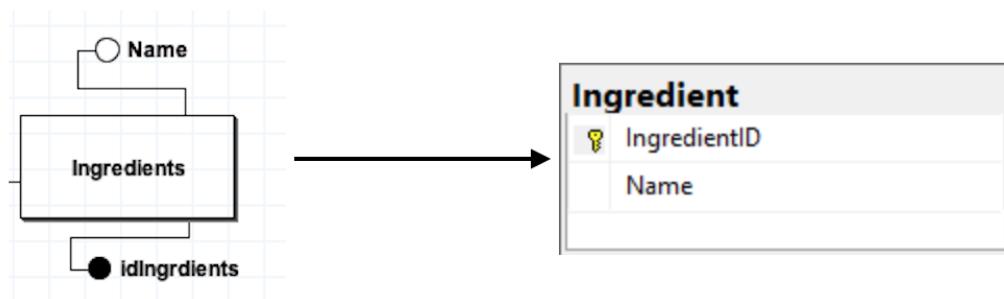
**Figura 93 - Conceptual para lógico (Entidade Recipe)**



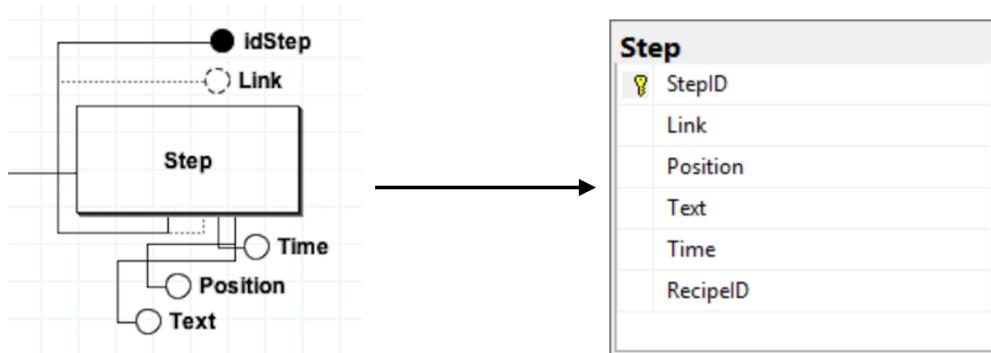
**Figura 94-** Conceptual para lógico (Entidade RecipeList)



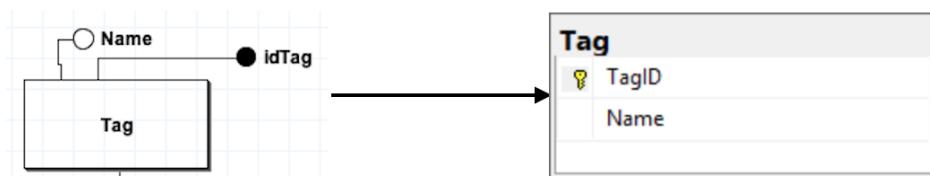
**Figura 95 -** Conceptual para lógico (Entidade Subscription)



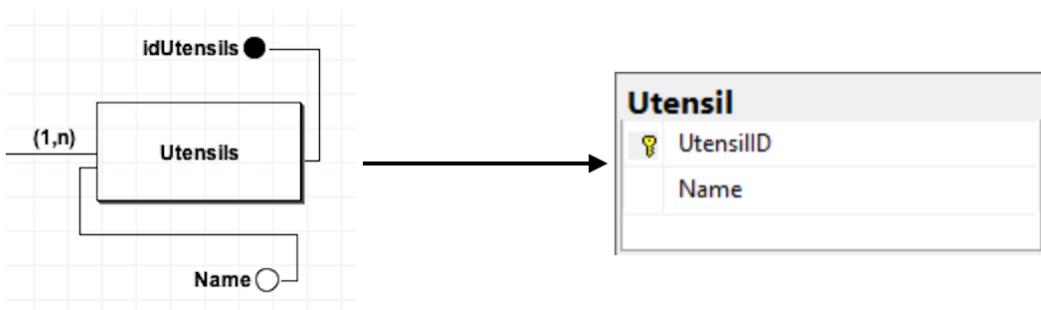
**Figura 96 -** Conceptual para lógico (Entidade Ingredient)



**Figura 97 - Conceptual para lógico (Entidade Step)**



**Figura 98 - Conceptual para lógico (Entidade Tag)**

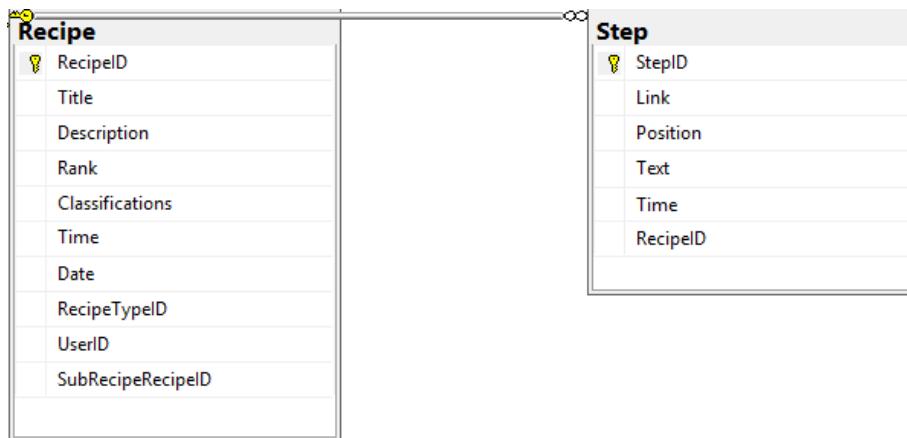


**Figura 99 - Conceptual para lógico (Entidade Utensils)**

### 10.2.1.2. Relacionamentos binários 1:N

No modelo conceitual desenhado, estão representadas 4 relações binárias 1:N. Que são as relações:

- User e Recipe: receitas criadas pelo utilizador;
- User e RecipeList: as listas de receitas com várias receitas de um utilizador;
- Recipe e Step: Passos a realizar para fazer a receita;

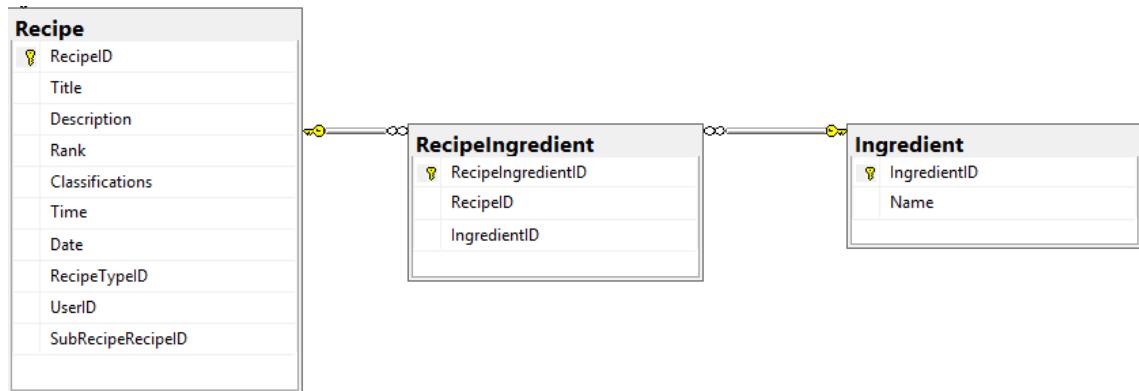


**Figura 100 - Exemplo de relação 1 : N**

### 10.2.1.3. Relacionamentos binários N: N

No modelo conceitual desenhado, estão representadas 6 relações binárias N: N. Que são as relações entre A e B, e entre C e D. Que são as relações:

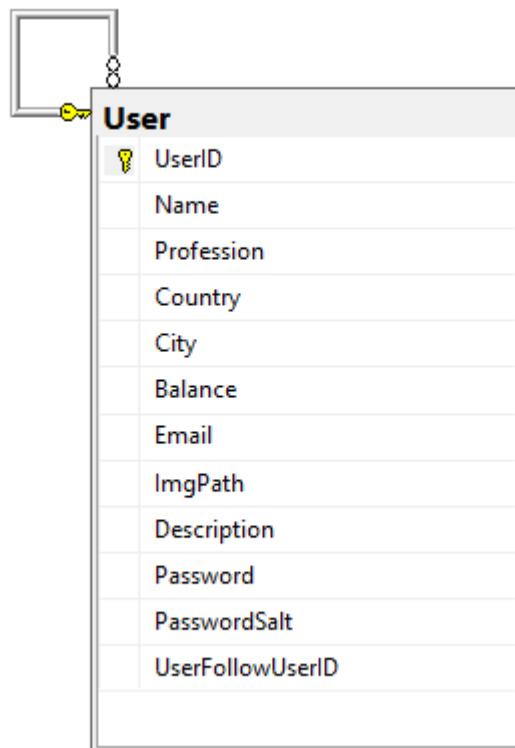
- User e Subscription: Trata todas as possíveis subscrições que um User pode submeter;
- User e Recipe: Esta relação trata o planos diários de receitas criados para os utilizadores usarem;
- User e Recipe : Esta relação trata os comentários a uma dada receita;
- Recipe e Tag : Cada receita possui uma lista de tags através das quais podem ser identificadas;
- Recipe e Ingredients : trata ingredientes presentes nas várias receitas, bem como as receitas que os utilizam.
- Recipe e Utensils : trata utensílios necessários nas várias receitas, bem como as receitas que os utilizam.



**Figura 101 - Exemplo de relação N : N**

#### 10.2.1.4. Auto-Relacionamentos binários

- User e User - que representa a lista de utilizadores que o seguem e trata uma ligação N : N.
- Recipe e Recipe - que trata a lista de sub receitas contidas na receita principal.



*Figura 102 - Exemplo de Auto-Relação*

## 10.2.2. Desenho do Modelo Lógico

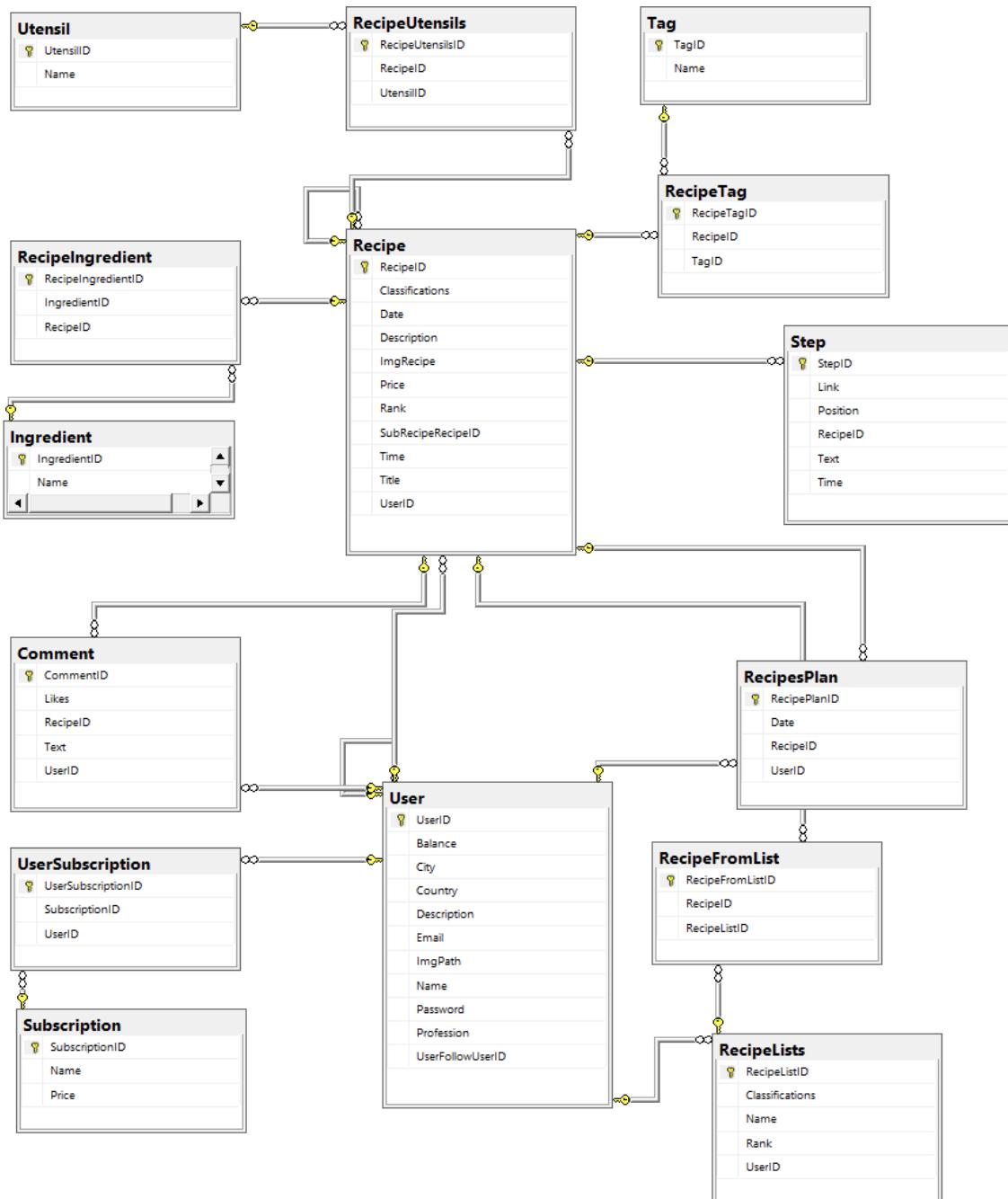


Figura 103 - Modelo Lógico

### 10.2.3. Validação do modelo através da normalização

Neste momento, e com vista a eliminar redundâncias, urgindo aumentar o desempenho e aumentar a integridade dos dados, procede-se à normalização dos mesmos, até à terceira forma normal, inclusive.

- **1<sup>a</sup>Forma Normal**

De modo a uma tabela estar de acordo com a 1<sup>a</sup>FN, é necessário que a intersecção de cada linha e coluna possua um, e apenas um valor.

Ora, de acordo com uma análise ao modelo atualmente construído, é possível constatar que não existem entidades com atributos repetidos e multi-valorados, ou seja, têm um e um só valor. Assim todas as tabelas estão na 1<sup>a</sup>FN.

- **2<sup>a</sup>Forma Normal**

Como anteriormente foi provado, o modelo encontra-se de acordo com a 1<sup>a</sup>FN, que é o primeiro requisito para estar de acordo com a 2<sup>a</sup>FN. A outra condição necessárias é que todos os atributos que não são chaves primárias, sejam completamente dependentes da chave primaria.

Algo que, através da observação das tabelas existentes, é também possível confirmar. Existem, no entanto, atributos que poderiam por si mesmos identificar uma tabela, como é o caso dos emails, no entanto foram anteriormente considerados como chaves candidatas, o que as torna também dependentes das primárias escolhidas.

- **3<sup>a</sup>Forma Normal**

Finalmente, para uma tabela se encontrar na 3aFN, é necessário que cumpra também a 2aFN. E que também não possua alguma dependência transitiva, ou seja, atributos não chave, que dependam de outros atributos não chave. Algo que não acontece em nenhuma das tabelas existentes no modelo lógico construído.

#### **10.2.4. Análise do crescimento futuro**

Embora a atual base de dados tenha sido criada para satisfazer as necessidades anteriormente mencionadas, é bastante flexível para trabalhar com alterações que sejam pretendidas.

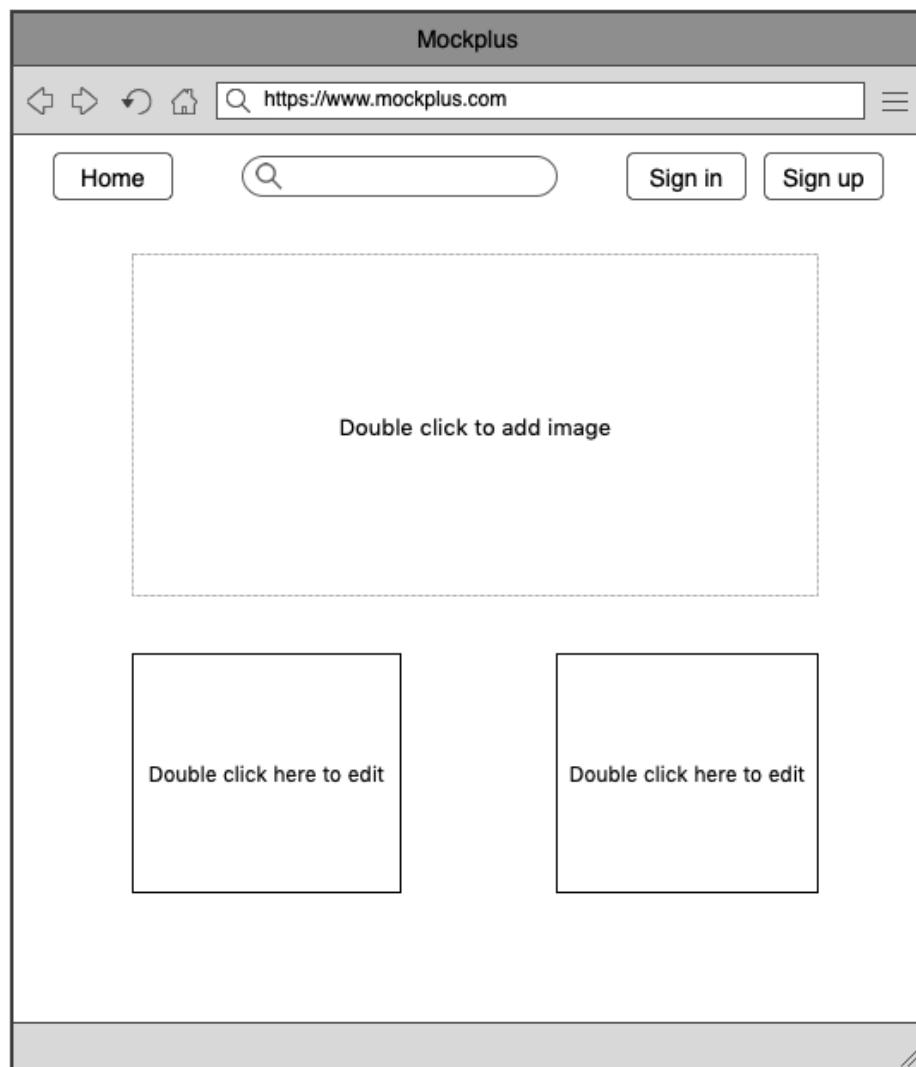
#### **10.2.5. Revisão do modelo lógico com o utilizador**

Após a realização do modelo lógico, este foi revisto pelo utilizador com o intuito de assegurar que este o considera uma representação fidedigna do que é pretendido. Nenhum problema foi detetado e por isso o modelo de dados foi aceite.

## 11. MockUps

Todos os mockups estão sujeitos a algumas alterações.

### HomePage



**Figura 104** - MockUp HomePage

**Registrar**



A wireframe mockup of a web-based registration form titled "Register". The form includes fields for UserName, Password, Confirm password, Job, Address, and Description. It features standard browser navigation icons at the top and "Confirm" and "Cancel" buttons at the bottom.

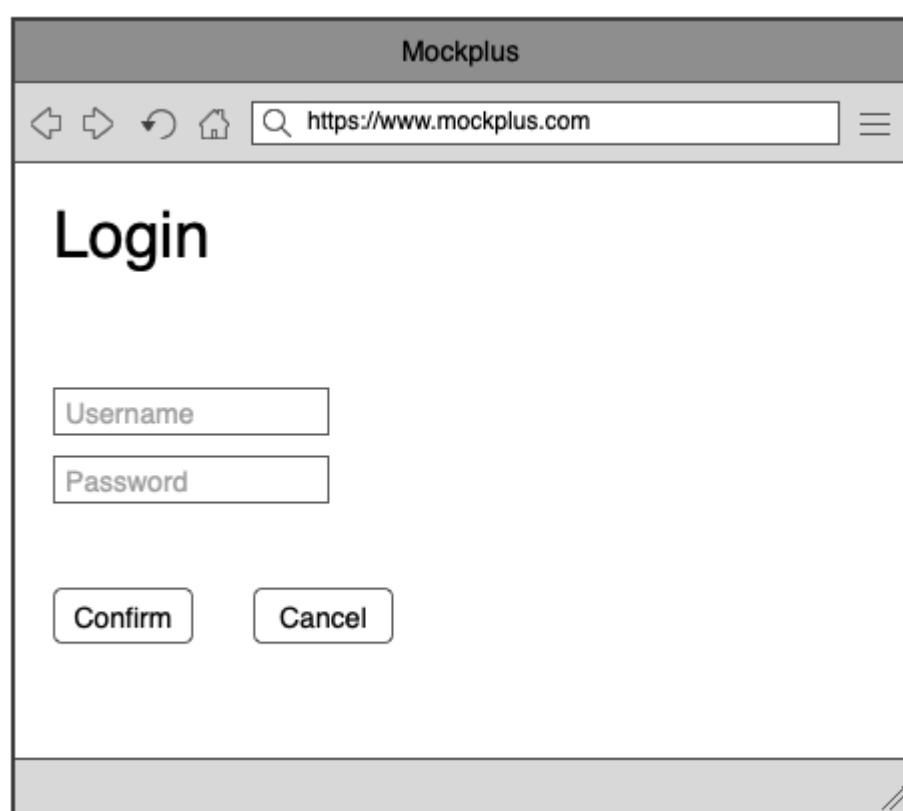
The form fields are labeled as follows:

- UserName
- Password
- Confirm password
- Job
- Address
- Description

At the bottom of the form are two buttons: "Confirm" and "Cancel".

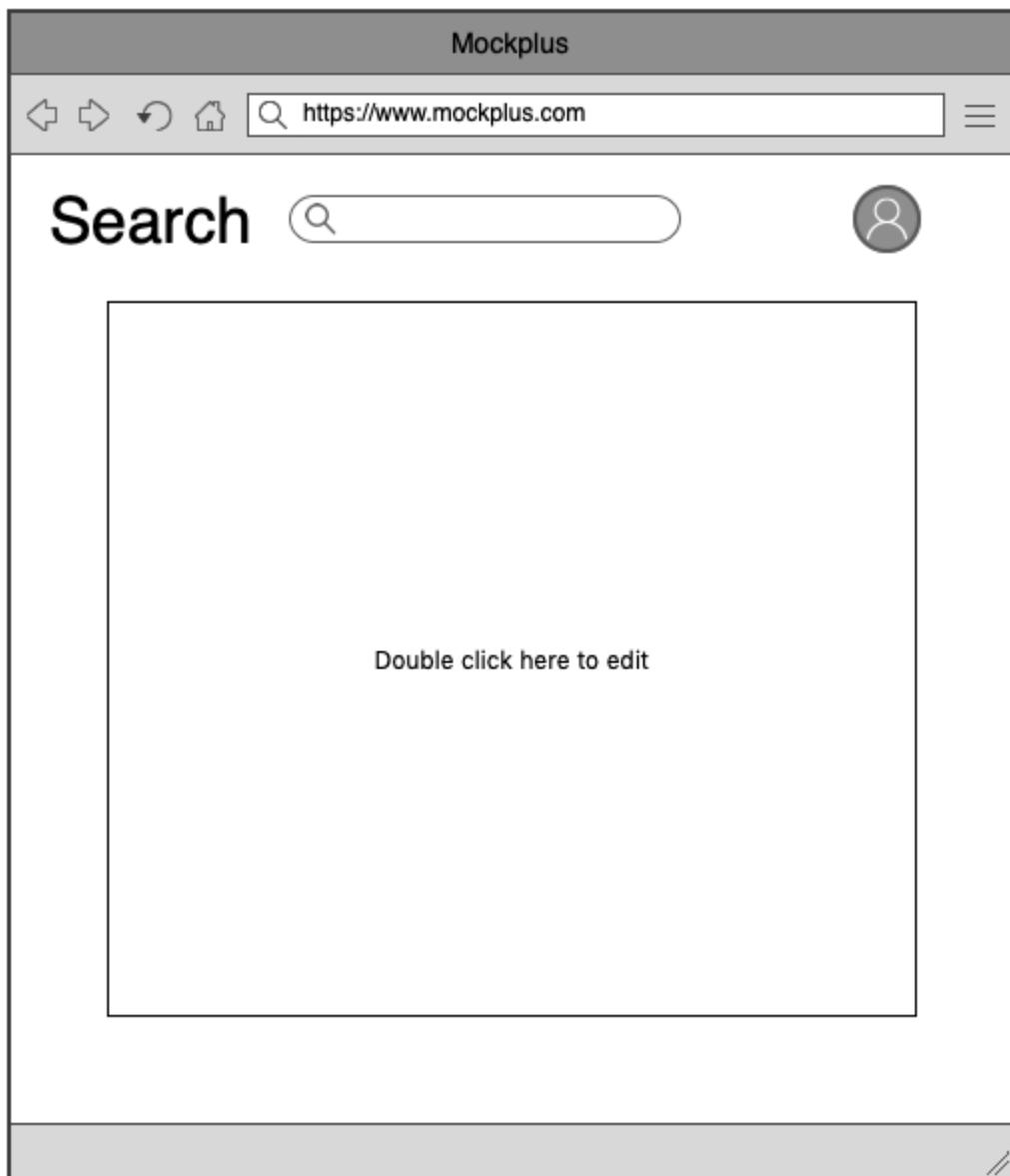
**Figura 105 - MockUp Register**

## Login



*Figura 106 - MockUp Login*

**Procurar**



**Figura 107 - MockUp Procurar**

## Consultar Receita

Mockplus

Home



Arroz de Pato

Description

Vivamus efficitur vestibulum elit id semper. Donec eleifend metus justo, ac luctus arcu finibus non. Phasellus dignissim diam ac ipsum egestas malesuada non mollis metus. Aenean luctus ex sit amet elementum tincidunt. Duis venenatis euismod tempor.

Tools

Vivamus efficitur vestibulum elit id semper. Donec eleifend metus justo, ac luctus arcu finibus non. Phasellus dignissim diam ac ipsum egestas malesuada non mollis metus. Aenean luctus ex sit amet elementum tincidunt. Duis venenatis euismod tempor.

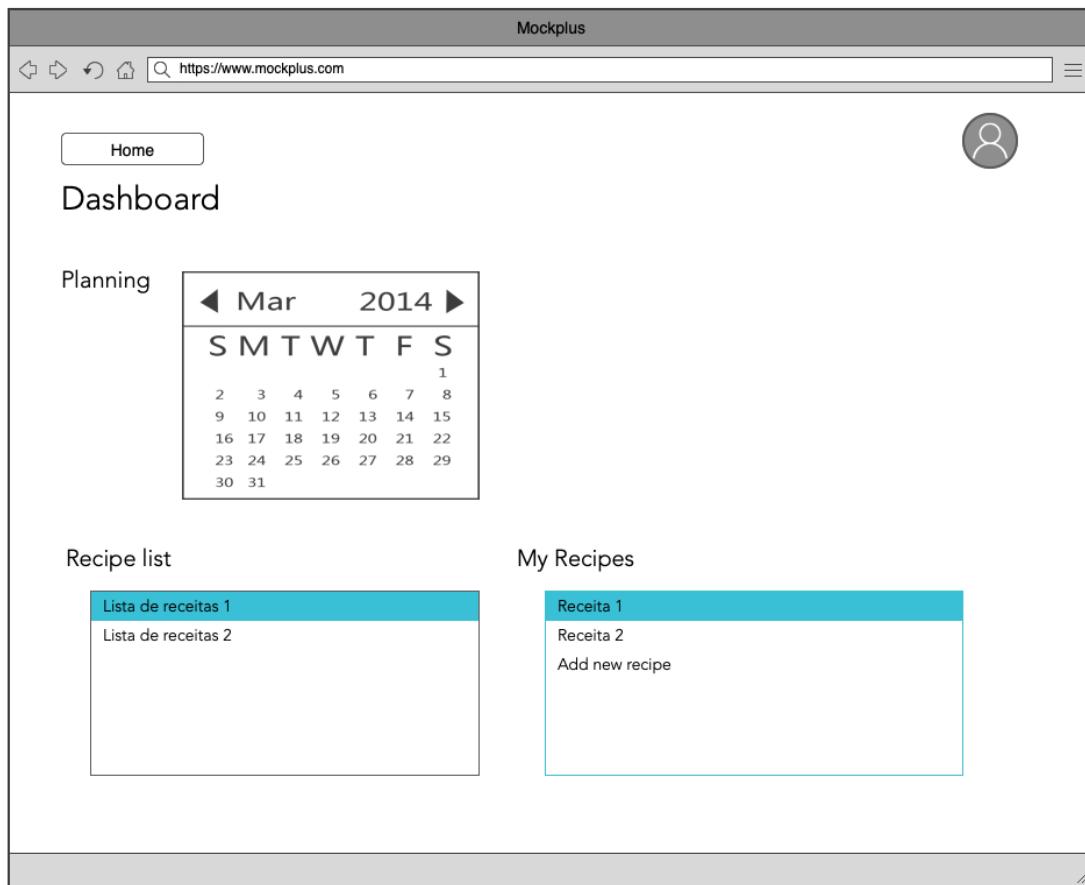
Ingredients

Vivamus efficitur vestibulum elit id semper. Donec eleifend metus justo, ac luctus arcu finibus non. Phasellus dignissim diam ac ipsum egestas malesuada non mollis metus. Aenean luctus ex sit amet elementum tincidunt. Duis venenatis euismod tempor.

Begin

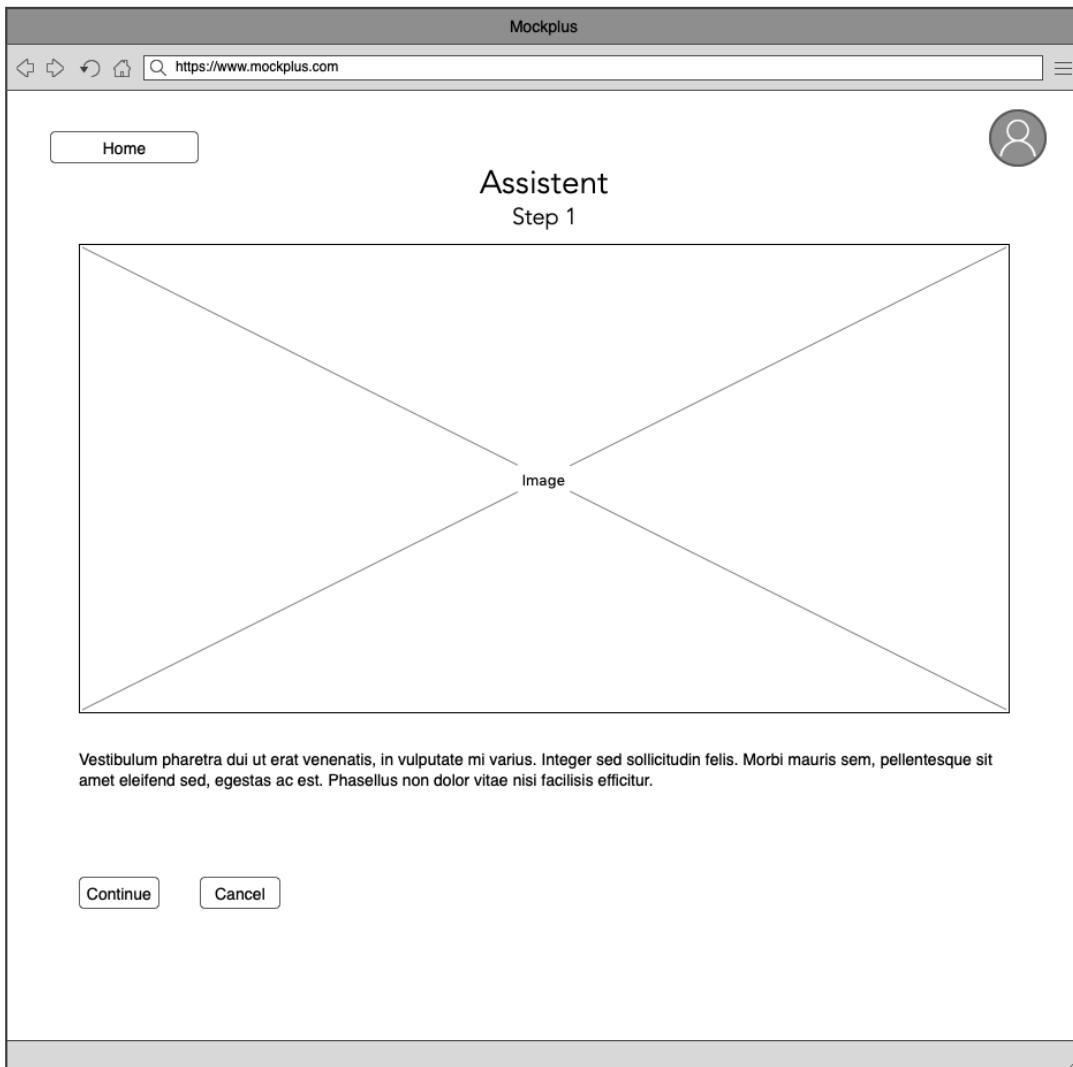
**Figura 108 - MockUp Consultar Receita**

## Dashboard



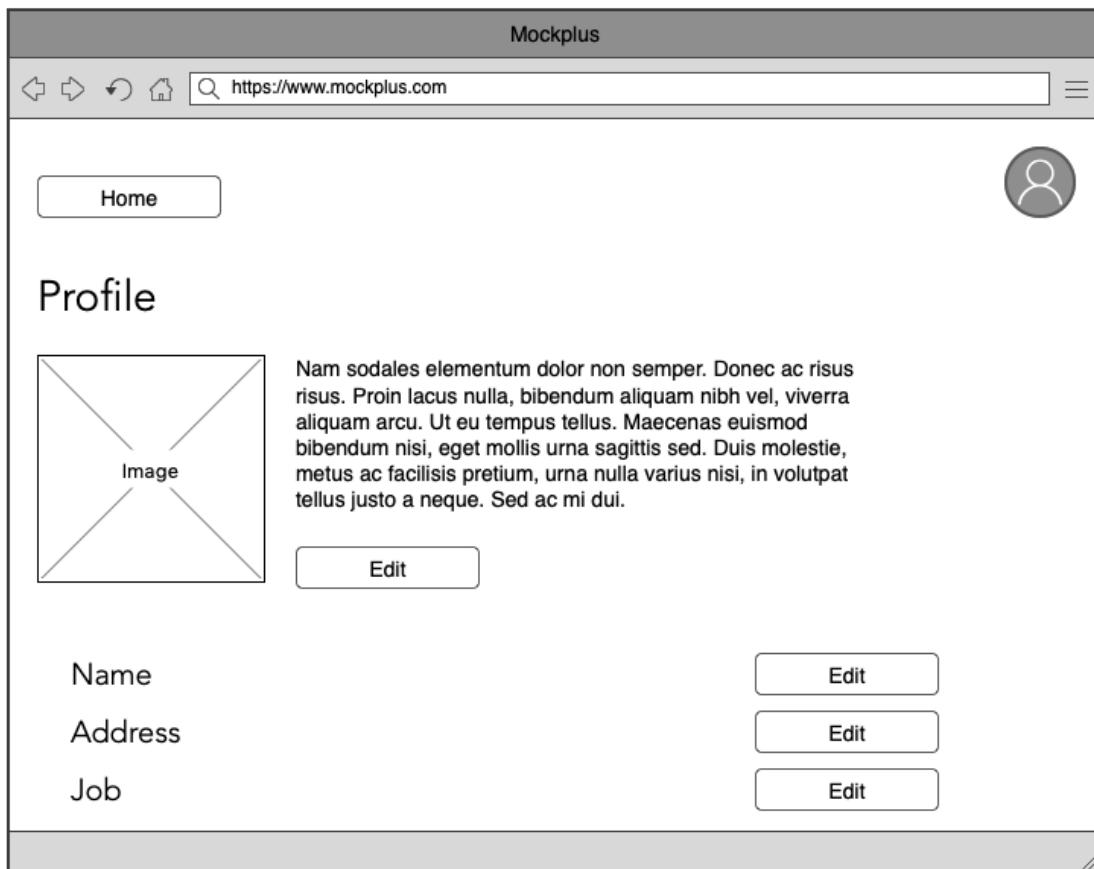
**Figura 109 - MockUp DashBoard**

## **Assistente CookieFy**



**Figura 110 - MockUp Assistente CookieFy**

## **Visualizar Perfil**



**Figura 111 - MockUp Visualizar Perfil**

## 12. Implementação

Tendo sido concluída a fase de modelação, tanto da arquitetura como também de base de dados, a equipa procedeu então ao desenvolvimento físico do website-aplicação proposto.

Para esta fase, tivemos de reunir e decidir quais as ferramentas que iríamos utilizar, de modo a que fossem eficazes e úteis face às necessidades correntes.

### 12.1. Análise de requisitos e tecnologias

Atendendo aos objetivos pretendidos nas fases anteriores do trabalho, era imperativo sermos capazes de utilizar a ferramenta, quer em PC como em *smartphone/tablet*.

Deste modo e após pesquisa e comparação com o desenvolvimento de *apps* específicas para cada um dos Sistemas Operativos (Windows e Android, fontes de utilização principal da tecnologia providenciada pelas ferramentas da Microsoft Azure) com o desenvolvimento de um sítio *web*, foi escolhida a opção de utilizar em ambos. Assim decidimos que deveria ser possível utilizar a aplicação em qualquer dispositivo com *browser*, de forma descomplicada e sem ter de fazer o *download* de qualquer *software*. A exigência de desenvolver uma aplicação para telemóvel será cumprida com um simples *wrapper* sobre a página, uma vez que esta terá propriedades adaptativas em relação ao tamanho do ecrã em uso, nomeadamente a ferramenta react.

#### 12.1.1. ASP.NET CORE

Uma vez que optamos por construir uma página *web*, surgiu-nos imediatamente a ideia de recorrer a ferramenta ASP.net.

As suas poderosas utilidades e documentação, simplicidade e rapidez de desenvolvimento, bem como a quantidade de conteúdo disponível *online* que a suporta, foram os principais componentes no que vou a esta tomada de decisão.

Ao contrário da habitual escolha de MVC, foi selecionado o ASP.NET Core. Devido à baixa complexidade das páginas a apresentar, que ou apresentam informação ou permitem introdução simples de informação, concluiu-se que não seria necessário implementar MVC, reduzindo o *overhead* aplicacional.

Assim, somos capazes de fornecer um ambiente que é mais organizado e compacto, uma vez que cada página funciona como *View* e *Model* por si só, mantendo o bom funcionamento e estabilidade da aplicação.

### 12.1.2. Visual Studio

Para acompanhar o ASP.net recorremos o VIsual Studio da Microsoft. Este é um poderoso IDE, com as mais diversas de extensões que permitem uma enorme diversidade nas opções e caminhos de desenvolvimento do projeto. De entre varias extensões, foram utilizadas algumas relacionadas à base de dados escolhida, SQL Server. Bem como outras com o intuito de ajudar ao funcionamento da web page que criamos para a aplicação.

### 12.1.3. SQL Server

Sendo esta uma solução de escolha na industria, capaz de oferecer quer rapidez quer segurança no seu funcionamento e apresentar uma metodologia relacional, para além também de uma completa integração com Visual Studio torna-se a melhor opção para o desenvolvimento do projeto.

### 12.1.4. Azure

Para fazer o *deployment* da página *web*, facilitado por uma extensão do VS, deveriam ter sido usados os serviços de *cloud* da Microsoft Azure. Contudo devido a um problema que encontramos relativamente às licenças universitárias que possuímos tal que optamos por um sistema de base de dados local para apresentar o projeto. Contudo está nos planos do grupo num tempo mais tardio migrar para os serviços de *cloud* da Microsoft Azure.

## 12.2. Desenho

Uma das características do ASP.net é permitir a replicação de uma página base, para outras instâncias. Assim sendo, era lógico desenhar esta página para conferir consistências visual e simplicidade ao sistema, tal como especificado.

## 12.3. Base de Dados

A base de dados requerida teria o seu *host* nos serviços da Microsoft Azure. Contudo devido a problemas relativos à subscrição ao serviço, bem como problemas de falta de privilégios optamos por migrar a base de dados para um host local a máquina. O seu povoamento foi realizado manualmente, através da plataforma, como forma de teste, através de Users que aceitaram entrar no programa beta. A criação e conexão à mesma é explorada nos tópicos seguintes.

### 12.3.1. Criação da Base de Dados

```
namespace CookieFy.Data
{
    public class CookieDbContext : DbContext
    {
        public CookieDbContext(DbContextOptions<CookieDbContext> options) : base(options) { }

        public DbSet<User> User { get; set; }
        public DbSet<Subscription> Subscription { get; set; }
        public DbSet<Recipe> Recipe { get; set; }
        public DbSet<RecipePlan> RecipesPlan { get; set; }
        public DbSet<Ingredient> Ingredient { get; set; }
        public DbSet<RecipeList> RecipeLists { get; set; }
        public DbSet<Step> Step { get; set; }
        public DbSet<Tag> Tag { get; set; }
        public DbSet<Utensil> Utensil { get; set; }
        public DbSet<UserSubscription> UserSubscription { get; set; }
        public DbSet<RecipeUtensils> RecipeUtensils { get; set; }
        public DbSet<RecipeTag> RecipeTag { get; set; }
        public DbSet<RecipePlan> RecipePlan { get; set; }
        public DbSet<Comment> Comment { get; set; }
        public DbSet<RecipeIngredient> RecipeIngredient { get; set; }
        public DbSet<RecipeList> RecipeList { get; set; }
    }
}
```

---

**Figura 112** - Definição da Base de Dados e das tabelas nela definidas

```

public class User
{

    // Props
    public int UserID { get; set; } // id to identify an user in db

    [Required] // model validation
    [StringLength(30)]
    public string Name { get; set; } // user name

    [Required]
    [StringLength(200)]
    public string Profession { get; set; } // user profession

    [Required]
    [StringLength(500)]
    public string Country { get; set; } // user address -> country

    [Required]
    [StringLength(500)]
    public string City { get; set; } // user address -> city

    [Required]
    public double Balance { get; set; } = 0; // user money

    [Required]
    [StringLength(500)]
    public string Email { get; set; } // user email

    public byte[] ImgPath { get; set; } = new byte[1024]; // user profile pic path

    [Required]
    [StringLength(500)]
    public string Description { get; set; } // user description

    [Required]
    public string Password { get; set; }

    //public string PasswordSalt { get; set; } // for encrypt

    // ----- Relationships -----
    // Comments
    public List<Comment> Comments { get; set; }
    // Recipes
    public List<Recipe> Recipes { get; set; }
    // Follows
    public List<User> Follows { get; set; }
    public User UserFollow { get; set; }
    // WeekPlan
    public List<RecipePlan> RecipesPlan { get; set; }
    // RecipeList
    public List<RecipeList> RecipeLists { get; set; }
    // Subscription
    public List<UserSubscription> Subscriptions { get; set; }
}

}

```

Figura 113 - Definição da tabela User

```

public class Recipe
{
    //Props
    public int RecipeID { get; set; } // to identify a recipe in db

    [Required] // model validation
    [StringLength(30)]
    public string Title { get; set; } // recipe title

    [Required]
    [StringLength(500)]
    public string Description { get; set; } // recipe description

    [Required]
    public int Rank { get; set; } = 0;

    [Required]
    public int Classifications { get; set; } = 0;

    [Required]
    public int Time { get; set; } // time to do the recipe

    [Required]
    public double Price { get; set; } = 0.0; // time to do the recipe

    //[[DatabaseGenerated(DatabaseGeneratedOption.Identity)]]
    public DateTime? Date { get; set; } // creation date

    public byte[] ImgRecipe { get; set; } = new byte[1024]; // user profile pic path

    // ----- Relationships -----
    // Recipes
    public int UserID { get; set; }
    public User User { get; set; }
    // Comments
    public List<Comment> Comments { get; set; }
    // WeekPlanRecipes
    public List<RecipePlan> RecipesPlan { get; set; }
    // RecipeUtensils
    public List<RecipeUtensils> RecipeUtensils { get; set; }
    // RecipeIngredients
    public List<RecipeIngredient> RecipeIngredients { get; set; }
    // RecipeTags
    public List<RecipeTag> RecipeTags { get; set; }
    // SubRecipes
    public List<Recipe> SubRecipes { get; set; }
    public Recipe SubRecipe { get; set; }
    // Steps
    public List<Step> Steps { get; set; }
}
}

```

**Figura 114** - Definição da tabela Recipe

```

// Recipes SubRecipes !! Auto-Relation
modelBuilder.Entity<Recipe>()
    .HasMany(r => r.SubRecipes)
    .WithOne(r => r.SubRecipe).IsRequired(false);
// -----

```

```

public class Ingredient
{
    //Prop
    public int IngredientID { get; set; }

    [Required]
    public string Name { get; set; }

    // ----- Relationships -----
    public List<RecipeIngredient> RecipeIngredients { get; set; }
}
}

```

**Figura 115** - Definição da tabela Ingredient

```

// Recipes Types !! One-to-Many
modelBuilder.Entity<RecipeType>()
    .HasMany(t => t.Recipes)
    .WithOne(r => r.RecipeType);
// -----

```

**Figura 116** - Definição de uma relação 1: N

```

// User's Comment !! Many-to-Many
modelBuilder.Entity<Comment>()
    //HasKey(c => new { c.UserID, c.RecipeID });
    //HasKey(c => new { c.CommentID });
modelBuilder.Entity<Comment>()
    .HasOne(c => c.User)
    .WithMany(u => u.Comments)
    .HasForeignKey(c => c.UserID);
modelBuilder.Entity<Comment>()
    .HasOne(c => c.Recipe)
    .WithMany(r => r.Comments)
    .HasForeignKey(c => c.RecipeID);

```

**Figura 117** - Definição de uma relação N : N

```

// Recipes SubRecipes !! Auto-Relation
modelBuilder.Entity<Recipe>()
    .HasMany(r => r.SubRecipes)
    .WithOne(r => r.SubRecipe).IsRequired(false);
// -----

```

**Figura 118** - Definição de uma auto-relação

## 13. HomePage



**Figura 119 - HomePage**

## 14. Funcionalidades

### 14.1. Login e Registo

A autenticação no sistema é realizada com username e password que cada utilizador define no seu registo como método de autentificação. Quando os dados são inseridos, o servidor confirma os dados de acesso (usando um comparador de hash, já que a password é encriptada no momento do registo do utilizador) e se se verificarem cria um cookie que é armazenado pelo browser do utilizador até este terminar sessão ou esta expirar. Desta forma, garantimos que a sessão é salvaguardada em todas as páginas do servidor.

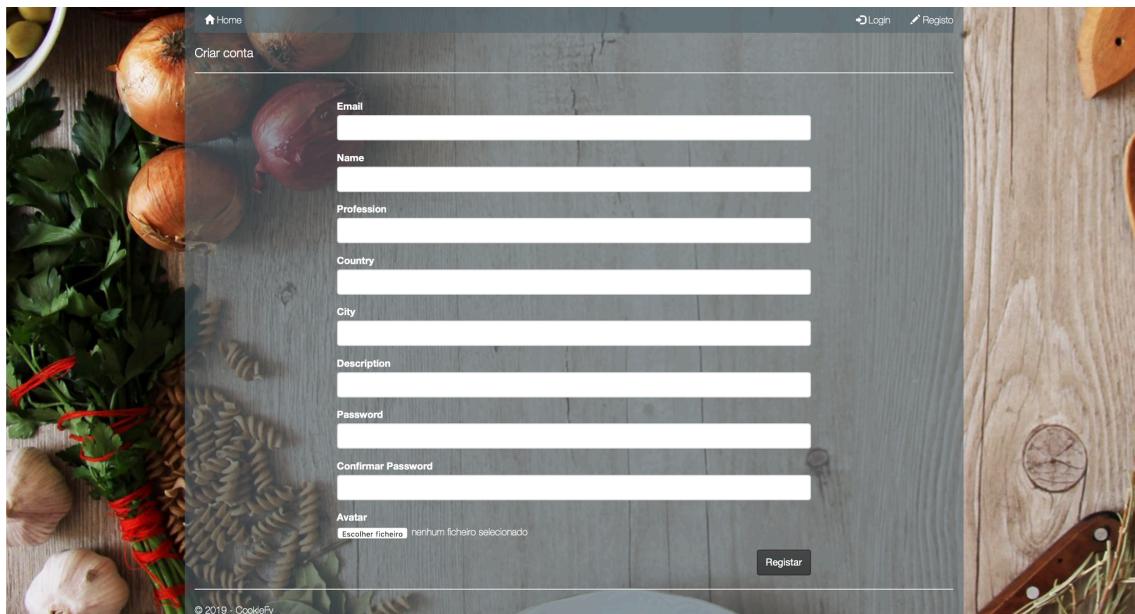
Providenciamos também a possibilidade de o browser guardar a sessão nos trinta dias posteriores ao início de sessão desde que o utilizador marque a caixa lembrar-me ao efetuar login. Desta forma, manter-se-á autenticado mesmo depois de fechar o browser.



**Figura 120** - Página de início de sessão

O registo no sistema é feito através da inserção de dados num formulário. Sendo que a inserção de uma imagem de perfil não é obrigatória e neste caso o sistema tratará de associar ao utilizador um avatar por defeito, o qual o utilizador pode mudar posteriormente.

A password é encriptada no momento do registo, por questões de segurança.



**Figura 121** - Registo de cliente

## 14.2. Área de Administrador

área específica ao administrador é constituída por botões que redirecionam este para páginas respetivas onde pode gerir o projeto. Ou seja, a partir desta área poderá gerir receitas, subscrições e utilizadores, através da consulta da lista destas entidades. O administrador pode também criar subscrições a inserir na aplicação.



**Figura 122 - Área de administrador (Opções)**

Na página da lista de utilizadores, o administrador pode consultar e gerir a informação de todos os utilizadores, bem como removê-los.

UserID	Nome	Email	Profissão	País	Cidade	Crédito	Descrição	Remover
2	Alfred Pennyworth	Alfred@gmail.com	Mordomo	America	Gotham	0	Excelente Cozinheiro	[remover]
7	Andy Samberg	andy@gmail.com	Actor	USA	Boston	0	Just me	[remover]
3	Gordon Ramsay	gordon@gmail.com	Chef	America	New York	0	Hell's Kitchen greatest Chef	[remover]
5	Jhon Doe	jhon@sapo.pt	Investigador	Portugal - Ilhas	Faial	0	Just Someone	[remover]
6	John Smith	smith@hotmail.com	Poicia	England	London	0	A op with a love for the art of cooking	[remover]
8	Leroy Jenkins	leroy	Gamer	Canada	Upside Down	0	LEEEEEEEROYYYYY JUUUUUUUENKINS!!!!	[remover]
4	Mark Zuckerberg	mark@gmail.com	Facebook Owner	America	Silicon Valley	0	Make a facebook account	[remover]

Figura 123 - Lista de utilizadores (Administrador)

Na página da lista de receitas, o administrador pode consultar os dados de todas as receitas na aplicação incluindo a classificação destas, bem como remover ou condicionar o uso destas.

Título	Descrição	Rank	Classificação	Tempo	Price	Type	Data de Registo	Detalhes
Lasanha	Hot Sauce lasagna for all ages	0	0	0	0	Free	03/06/2019	[detalhes]
okdovo	wodqw	0	0	0	0	Free	03/06/2019	[detalhes]
Almondegas saborosas	Bela comida	0	0	0	0	Free	03/06/2019	[detalhes]
Lasagna	Awesome Lasagna	0	0	0	0	Free	04/06/2019	[detalhes]
Massa à Bolonhesa	Great Pasta!!!	0	0	0	0	Free	04/06/2019	[detalhes]
Pizza Carbonara	Pizza	0	0	0	0	Free	04/06/2019	[detalhes]
Saladinha verde	saudável	0	0	0	0	Free	04/06/2019	[detalhes]
Sopa de tomate	Tomate bom	0	0	0	0	Free	04/06/2019	[detalhes]

Figura 124 - Lista de receitas (Administrador)

The screenshot shows the 'CookieFy' administrator interface for managing subscriptions. The top navigation bar includes links for 'Home', 'Área de Admin' (Administrator Area), and 'Logout'. A cartoon chef character is prominently displayed in the center. The main content area is titled 'Lista de Subscrições' (List of Subscriptions) and features a table with the following data:

SubscriptionID	Name	Price	Remover
2	Premium	29.99	[Remove]
3	Subscrição 1 mês	14.99	[Remove]
4	Subscrição 3 meses	49	[Remove]
5	Subscrição 1 ano	60	[Remove]
6	Subscrição Free-Trial : 1 semana	0	[Remove]
7	Subscrição Vitalícia	250	[Remove]

Below the table, there is a note 'A mostrar a página 1 de 1' (Showing page 1 of 1) and navigation buttons for 'Anterior' (Previous), 'Próximo' (Next), and a red '1' indicating the current page.

**Figura 125 - Registar de Subscrições (Administrador)**

### 14.3. Área de Utilizador

A área reservada ao utilizador tem funcionalidades diferentes do administrador uma vez que é este quem vai usufruir das atividades da aplicação. Aqui, cada utilizador poderá ver os dados da sua conta e alterar o seu avatar assim como algumas das suas informações. Pode também criar uma nova receita, começar a confeccionar uma receita e procurar outros utilizadores ou receitas.

The screenshot shows the 'CookieFy' user interface for profile options. The top navigation bar includes links for 'Home', 'Área do Utilizador' (User Area), and 'Logout'. A cartoon chef character is in the top right. The main content area is titled 'Área do Utilizador' (User Area) and displays a user profile for 'Gordon Ramsay' with the following details:

- Email: gordon@gmail.com
- Profissão: Chef
- País: América
- Cidade: New York
- Crédito: 0
- Descrição: Hell's Kitchen greatest. Chef

Below the profile, there are several buttons:

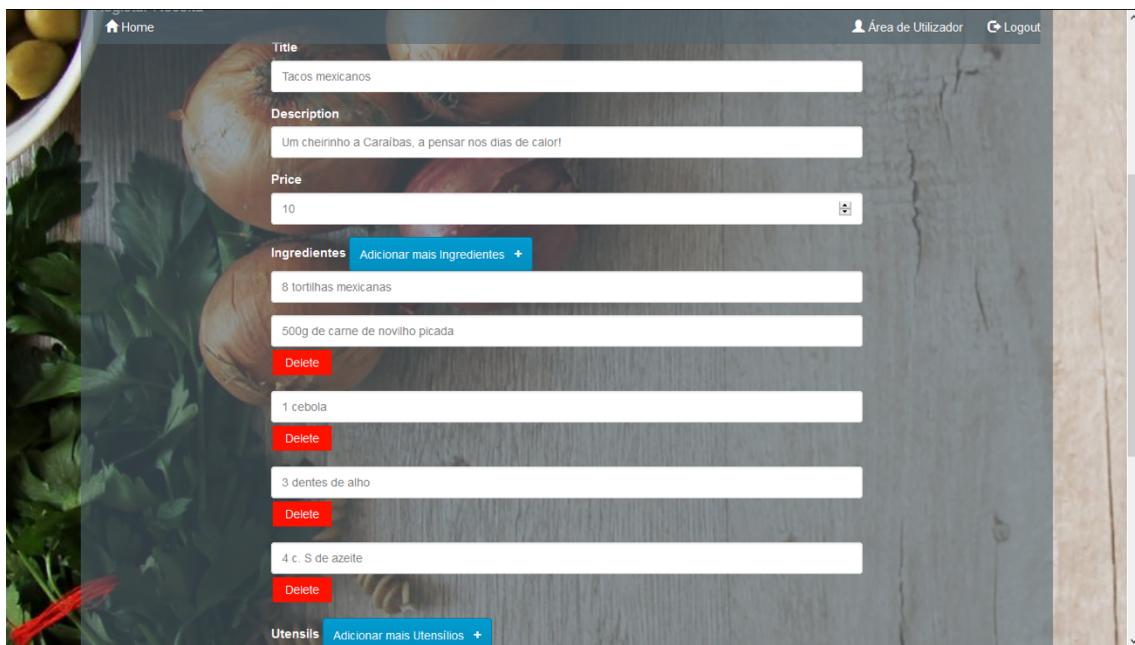
- Recetas (Recipes)
- Lista de Receitas (List of Recipes)
- Piano Semanal (Weekly Planner)
- Editar Perfil (Edit Profile)
- Registrar Receita (Register Recipe)
- Carregar Créditos (Load Credits)
- Adicionar Subscrição (Add Subscription)

At the bottom left, it says '© 2018 - CookieFy'.

**Figura 126 – Área de Utilizador (Opções)**



**Figura 127 - Editar perfil (Utilizador)**



**Figura 128 - Criar Receita (Utilizador)**



**Figura 129 - Carregar Créditos (Utilizador)**

## 14.4. Área de Confeção de Receita

### 14.4.1. Menu de Receita

Nesta página estão todas as informações mais pertinentes de uma dada receita que pode ajudar um utilizador a escolher confeccionar a receita.

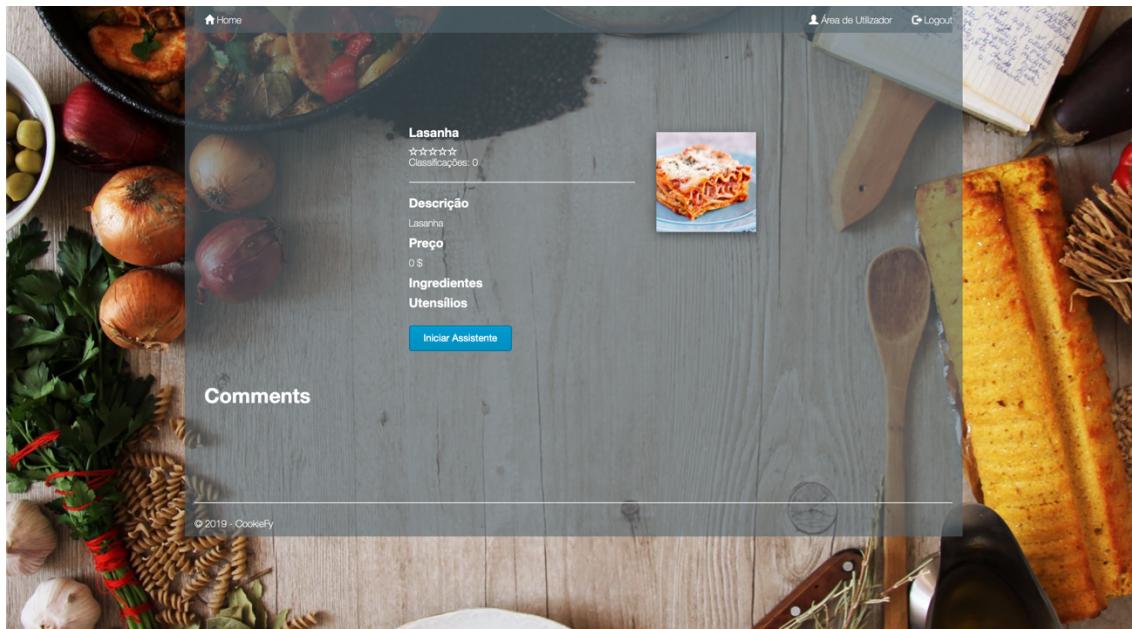
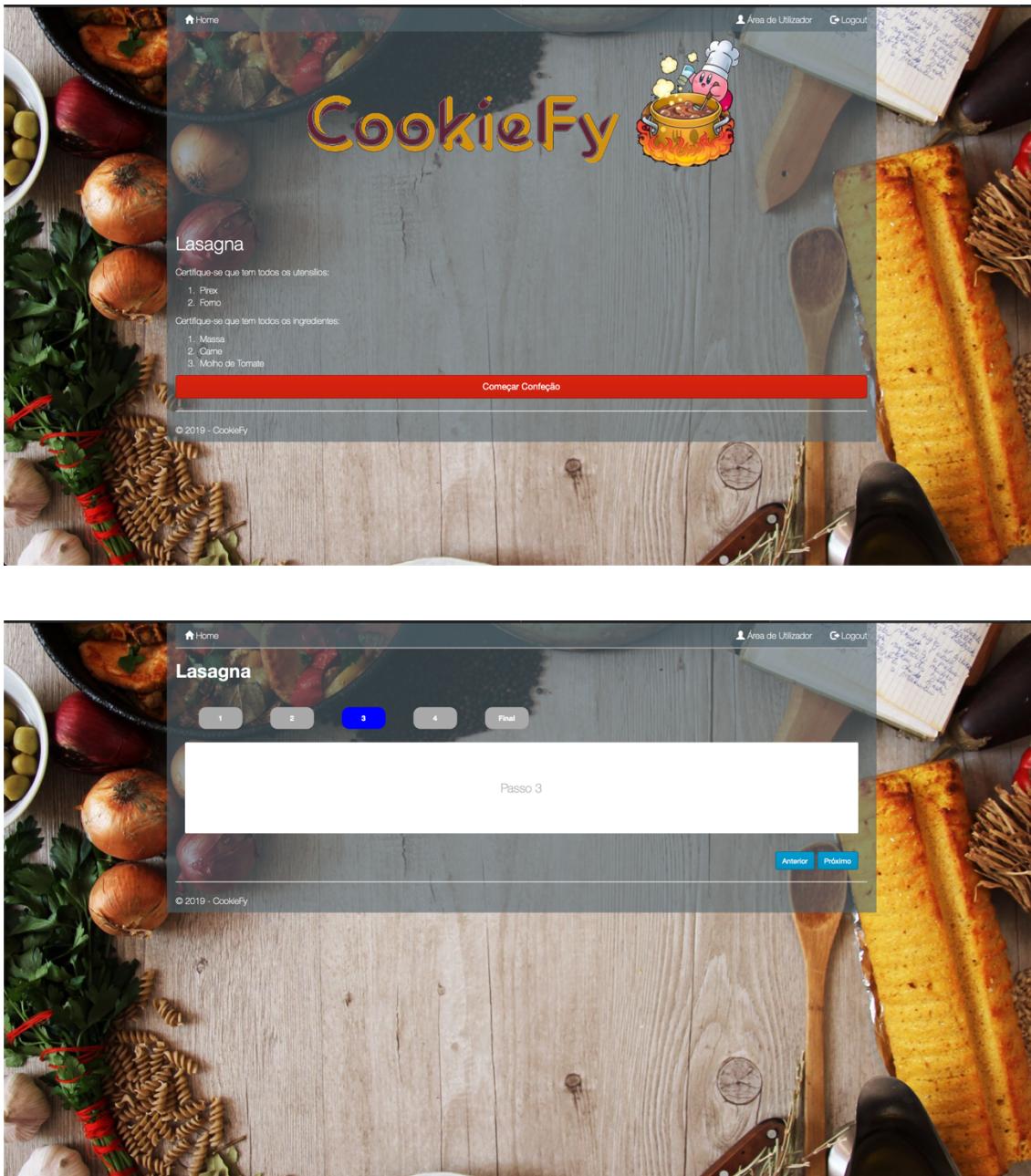


Figura 130 - Dados de uma Receita

#### 14.4.2. Assistente de Cozinha

Ao iniciar a confeção de receita e ativado o assistente de cozinha que vai acompanhar o utilizador ao longo dos passos apresentando os dados e informação de cada passo. O utilizador pode controlar o andamento do assistente através de botões.



**Figura 131** - Assistente de cozinha

O assistente foi inicialmente idealizado como capaz de comunicar através de reconhecimento de voz com o utilizador, contudo não conseguimos implementar esta capacidade devido a uma dificuldade em encontrar um método para inserir o Cortana e trabalhar dentro da aplicação.

## 15. Conclusões e Trabalho Futuro

O projeto desenvolvido debruça-se sobre a ideia de criar uma WebApp com assistente de voz capaz de acompanhar o utilizador na confeção de uma receita de comida, deixando ao critério do grupo a escolha do tema alimentar e todo o trabalho envolvido na criação e fundamentação do mesmo.

Será ainda importante referir que todo o processo de idealização e todo o esforço envolvido na adequação da ideia a possíveis contextos prático aos quais se iria utilizar a aplicação.

Na primeira fase, foi necessário desenvolver um modelo geral para o pedido do projeto, e apresentar uma conceção inicial, bem como esclarecer alguns pontos de sucesso e viabilidade, e analisar os recursos que consideramos necessários ao serviço.

Já na segunda fase, especificamos os requisitos de utilizador e de sistema, elaboramos os diagramas de especificação de projeto e mãe todas a atividades que o projeto implementaria, construímos uma base de dados que achamos ser adequada na altura e, por fim, desenhamos os *mockups* para a aplicação.

Um dos passos mais importantes foi a descrição dos requisitos, pois são estes onde nos iríamos basear para o desenvolvimento do sistema.

Na terceira e última fase, o grupo procedeu à implementação das funcionalidades propostas, incluindo um servidor web em ASP.NET e uma base de dados SQL Server que foi revista após diversas dificuldades na implementação de alguns métodos desenvolvidos. O processo de desenvolvimento envolveu todo o serviço de gestão utilizadores, receitas e outros recursos necessários ao bom funcionamento do projeto.

Contudo sabemos que ainda não atingimos tudo o que almejamos com esta aplicação ainda existe uma diversidade de funcionalidades que esperamos vir a implementar através de atualizações futuras tal que o objetivo inicial de uma aplicação para todos os gostos capaz de acompanhar qualquer pessoa independentemente do gosto ou intenção culinária. Existem ainda algumas funcionalidades chave que nos faltam, mas sabemos que com algum trabalho e tempo mais seriam possíveis de implementar.

Ao concluirmos o projeto sentimos que ganhamos um novo nível de experiência, uma vez que enredamos um trabalho de nível mais elevado que todos os anteriores que tínhamos desenvolvido no Curso, em que aplicamos o conhecimento adquirido ao longo dos anos em várias unidades Curriculares. Adicionalmente sentimos como é trabalhar num contexto mais profissional em que iniciamos o desenvolvimento do projeto seguindo todos os passos essenciais nos métodos de desenvolvimento de projetos, sendo assim uma experiência muito positiva que nos prepara para um futuro pós-académico.