

# Trabalho Prático De Modelação e Design - Fase 3 (Design da Solução)

## 1. Identificação

**1.1 Meta a que diz respeito o documento:** Este documento refere-se à Fase 3 do trabalho.

**1.2 Turma prática:** P4/P1

**1.3 Nome, número e endereço de e-mail de cada elemento do grupo:**

- Duarte Xavier de Oliveira Santos – 2022149622 – [a2022149622@isec.pt](mailto:a2022149622@isec.pt)
- Gustavo Trigo Costa - 2023145800 – [a2023145800@isec.pt](mailto:a2023145800@isec.pt)
- Gonçalo Ferraz Bento – 2023137116 – [a2023137116@isec.pt](mailto:a2023137116@isec.pt)

<b>1. Identificação</b>	<b>1</b>
<b>2. Descrição Pormenorizada do Caso de Uso</b>	<b>2</b>
2.1 Nome: Agendar Refeição	2
2.2 Atores:	2
2.3 Objetivo:	3
2.4 Pré-Condições:	3
2.5 Pós-Condições:	3
2.6 Fluxo de eventos:	3
2.7 Cenário Principal:	3
2.8 Cenários Alternativos:	4
2.9 Casos de uso relacionados:	5
<b>3. Modelo do Domínio</b>	<b>7</b>
<b>4. Diagrama de Sequência do Sistema (DSS)</b>	<b>7</b>
<b>5. Glossário</b>	<b>9</b>
<b>6. Mensagens</b>	<b>10</b>
6.1 Diagrama de Classes (Baixo Nível)	10
6.2 Diagrama de Sequência (Baixo Nível)	12

## **2. Descrição Pormenorizada do Caso de Uso**

### **2.1 Nome: Agendar Refeição**

### **2.2 Atores:**

Membro da família - Quem vai agendar a refeição

Serviço de Receitas – Sugere receitas ao utilizador

Serviço de calendário - Permite ao utilizador exportar a agenda de refeições para um calendário externo.

Serviço de supermercado – Sugere os supermercados para o utilizador comprar ingredientes caso estes estejam em falta.

## **2.3 Objetivo:**

- Permitir ao Membro da Família planejar refeições futuras, associando receitas a horários específicos, garantindo a disponibilidade de ingredientes e a integração com serviços externos de calendário e supermercados.

## **2.4 Pré-Condições:**

- O utilizador deve estar autenticado no sistema.
- O serviço de receitas deve de estar disponível para sugerir ou selecionar receitas.
- O serviço de calendário deve estar disponível para registar o agendamento.
- O serviço de supermercado deve estar funcional para sugerir o melhor local para se comprar os ingredientes em falta.

## **2.5 Pós-Condições:**

- A refeição fica registada na Agenda de Refeições.
- Caso existam ingredientes em falta, são adicionados automaticamente à Lista de Compras feita pela aplicação.
- A agenda de refeições é exportada para o calendário externo caso o utilizador o deseje.
- Caso seja necessário descongelar algum ingrediente para a refeição, o utilizador recebe um lembrete da aplicação.

## **2.6 Fluxo de eventos:**

## **2.7 Cenário Principal:**

1. O Membro da Família seleciona a opção "Agendar Refeição".
2. O sistema apresenta as receitas disponíveis e/ou sugestões do Serviço de Receitas.
3. O Membro da Família seleciona uma receita e define a data e hora da refeição.
4. O sistema verifica a disponibilidade dos ingredientes no stock atual (despensa, frigorífico ou congelador).

5. Caso faltem ingredientes, o sistema irá adicioná-los à Lista de Compras automaticamente.
6. O sistema regista a refeição na Agenda de Refeições.
7. O sistema envia a atualização para o Serviço de Calendário, caso o utilizador tenha exportado a Agenda de Refeições.
8. O sistema verifica se algum ingrediente requer descongelação e, se for esse o caso, cria um lembrete de descongelação.
9. O Membro da Família confirma o agendamento e o processo é concluído.

## **2.8 Cenários Alternativos:**

### **1. a) [Alternativa 1] - Serviço de Receitas não disponível:**

1.a).1 O membro da família seleciona a opção de agendar refeição.

1.a).2 O sistema tenta obter uma receita através do Serviço de Receitas, mas este está indisponível.

1.a).3 O sistema irá avisar o cliente, permitindo que este selecione uma receita previamente guardada, podendo o utilizador adicionar essa receita ou cancelar o agendamento.

1.a).3.1 O utilizador escolhe adicionar uma receita guardada e continua para o ponto 1.a).4

1.a).3.2 O utilizador decide cancelar o agendamento, e então o caso de uso termina em falha.

1.a).4 Tudo continua igual após a seleção da receita.

### **2. [Alternativa 2] - Ingredientes em Falta e Serviço externo de Supermercados indisponível:**

2.a)1 O sistema irá verificar a disponibilidade dos ingredientes.

2.a).2 O sistema vê que está um ou mais ingredientes em falta e tenta sugerir um supermercado.

2.a).3 O serviço de sugestão de supermercados está inativo.

2.a).4 O sistema avisa o utilizador, permitindo que o utilizador adicione assim manualmente os ingredientes em falta à Lista de Compras da aplicação, ou então que este cancele a operação.

2.a).4.1 O utilizador adiciona manualmente o ingrediente em falta e continuando assim o agendamento da refeição, passando para o ponto 2.a).5

1.a).4.2 O utilizador decide cancelar o agendamento, e então o caso de uso termina em falha.

2.a).5 O fluxo segue normalmente para o agendamento da refeição.

## 2.9 Casos de uso relacionados:

Consultar receitas sugeridas pela aplicação, Gerir Lista de Compras, Exportar Agenda para Calendário Externo e Ver os Supermercados sugeridos pela aplicação.



### 3. Modelo do Domínio

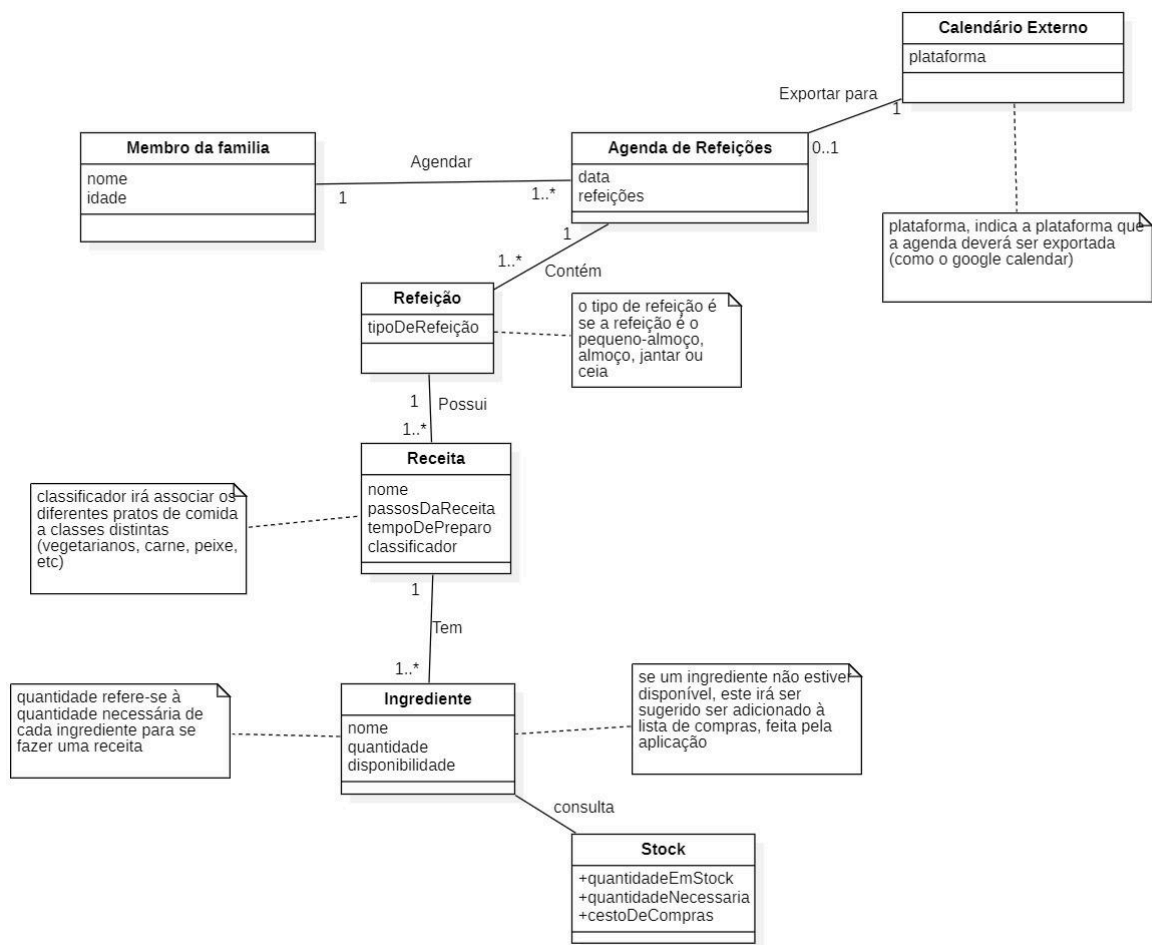
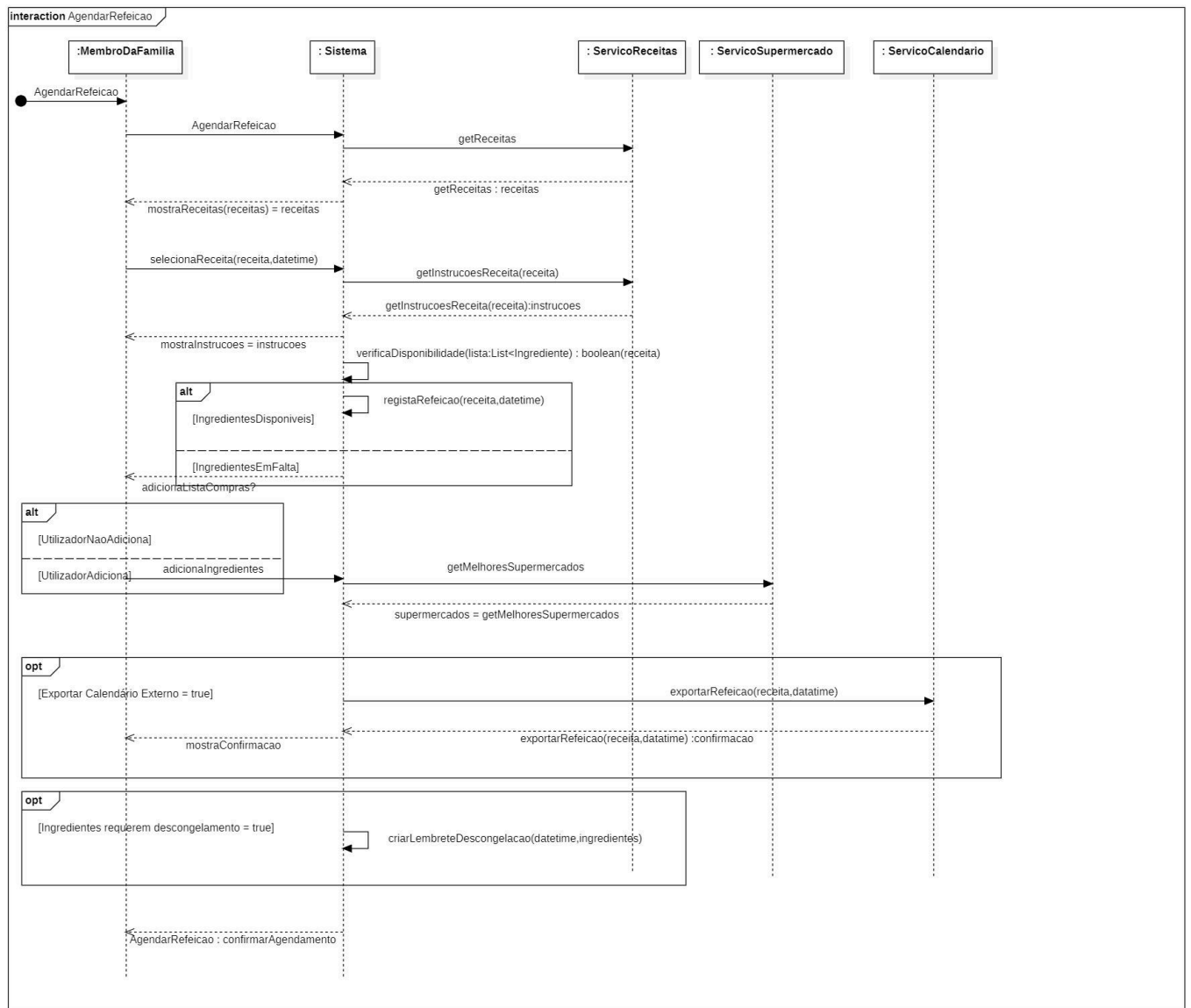


Figura 2 – Modelo de domínio

## 4. Diagrama de Sequência do Sistema (DSS)



## 5. Glossário

Termo	Definição
Membro da Família	<p><b>Definição:</b> Consiste no utilizador principal do sistema que agenda refeições, consulta receitas, gere listas e interage com os serviços disponíveis.</p> <p><b>Formato:</b> Objeto com nome e credenciais.</p> <p><b>Relação com outros termos:</b> Utilizador Autenticado, Agenda de Refeições.</p> <p><b>Regras de validação:</b> Deve possuir uma conta ativa no sistema.</p>
Serviço de Receitas	<p><b>Definição:</b> Módulo do sistema externo que sugere receitas ao utilizador com base em pré-requisitos.</p> <p><b>Formato:</b> Lista de objetos Receita.</p> <p><b>Relação com outros termos:</b> Receita Guardada, Lista de Compras, Agendar Refeição.</p> <p><b>Regras de validação:</b> Deve conter informação nutricional e ingredientes disponíveis e tem que estar disponível para que o agendamento possa ocorrer.</p>
Serviço de Calendário	<p><b>Definição:</b> Funcionalidade que permite exportar a agenda de refeições para um calendário externo.</p> <p><b>Formato:</b> Arquivo compatível com aplicações de calendário.</p> <p><b>Relação com outros termos:</b> Exportar Agenda, Agendar Refeição, Agenda de refeições.</p> <p><b>Regras de validação:</b> Apenas utilizadores autenticados podem exportar dados.</p>



Serviço de  
Supermercado

**Definição:** Funcionalidade que sugere supermercados para compra de ingredientes em falta.

**Formato:** Lista de strings (nomes dos supermercados).

**Relação com outros termos:** Lista de Compras, Ingrediente em Falta.

**Regras de validação:** Deve listar apenas supermercados ativos na região do utilizador.

Lembrete de  
Descongelação

**Definição:** Notificação enviada pelo sistema para alertar o utilizador sobre a necessidade de descongelar um ingrediente antes da preparação da refeição.

**Formato:** Notificação no aplicativo.

**Relação com outros termos:** Agendar Refeição, Stock.

**Regras de validação:** Deve ser emitido com antecedência suficiente para permitir a descongelação adequada.

Stock

**Definição:** Conjunto de ingredientes armazenados pelo utilizador, incluindo despensa, frigorífico e congelador.

**Formato:** Lista categorizada de strings.

**Relação com outros termos:** Ingrediente em Falta, Lista de Compras, Receita.

**Regras de validação:** Deve ser atualizado sempre que há consumo ou reposição de itens.

Utilizador Autenticado

**Definição:** Consiste no utilizador que realizou login no sistema e tem permissão para aceder as diferentes funções.

**Formato:** Sessão verificada por login e senha.

**Relação com outros termos:** Membro da Família, Exportar Agenda.

**Regras de validação:** O login deve ser válido e ativo.

## Agenda de Refeições

**Definição:** Registo das refeições agendadas, contendo a receita, a data e hora.

**Formato:** Lista estruturada de objetos Refeição.

**Relação com outros termos:** Membro da Família, Serviço de calendário, Receita.

**Regras de validação:** O login deve ser válido e ativo.

## Lista de Compras

**Definição:** Lista automática ou manual dos ingredientes que estão em falta e precisam de ser comprados.

**Formato:** Lista de Strings (ingredientes)

**Relação com outros termos:** Ingrediente em falta, Serviço de Supermercado, Agendar refeição.

**Regras de validação:** Atualiza automaticamente ao detetar ingrediente, mas pode também ser editada pelo utilizador.

## Ingrediente em falta

**Definição:** Ingrediente em falta que não está disponível para uma receita no stock atual do utilizador.

**Formato:** String

**Relação com outros termos:** Lista de Compras, Stock.

**Regras de validação:** Verificado através da verificaDisponibilidade(receita).

## Receita

**Definição:** Conjunto de instruções e ingredientes necessários para fazer uma refeição.

**Formato:** Objeto (nome, descrição, lista de ingredientes).

**Relação com outros termos:** Serviço de receitas, Ingredientes, Agenda de refeições.

**Regras de validação:** O utilizador deve de ter todos os ingredientes em stock.

**Definição:** Mensagem enviada pelo utilizador ao sistema para indicar a receita escolhida e a sua respetiva hora e data de confeção.

**Formato:** Método com argumentos(receita, datetime).

**Relação com outros termos:** Agendar Refeição, Receita, Agenda de Refeições

**Regras de validação:** A receita e o datetime têm que ser válidos (datetime tem que ser depois do datetime atual).

**Definição:** Mensagem interna do sistema que verifica se os ingredientes do sistema estão disponíveis ou não no stock do utilizador.

**Formato:** Método com argumento (receita).

**Relação com outros termos:** Ingredientes em falta, Lista de Compras.

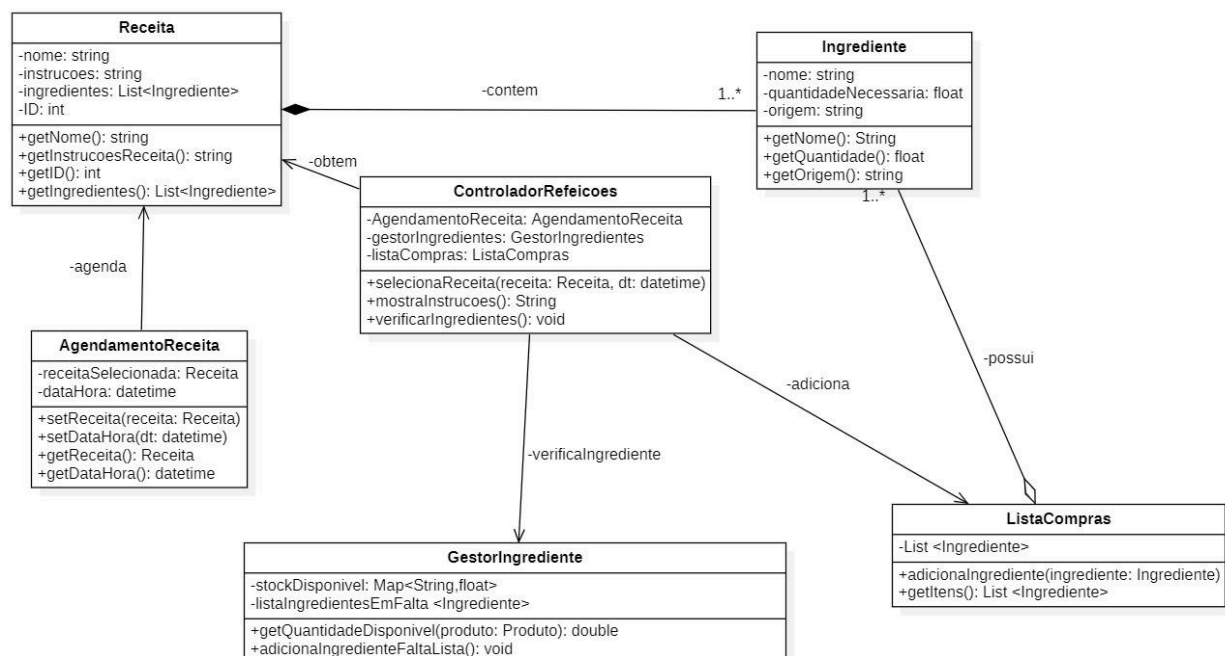
**Regras de validação:** Retorna true ou false e uma lista de ingredientes em falta, caso estes existam.

## 6. Mensagens

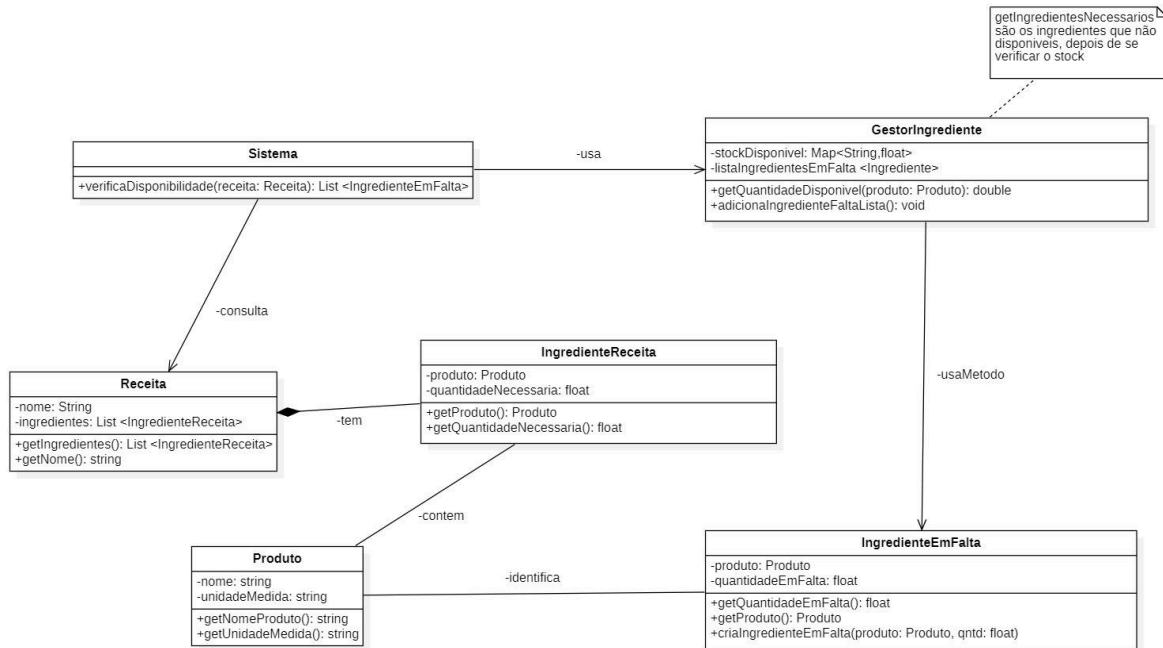
Escolhemos as mensagens `selecionaReceita(receita, datetime)` e `verificaDisponibilidade(receita)` por representarem momentos centrais e com lógica relevante no processo de agendamento de uma refeição. Estas permitem evidenciar a interação entre classes do sistema, a verificação de dados em memória e a aplicação de princípios de design orientado a objetos.

### 6.1 Diagrama de Classes (Baixo Nível)

Mensagem: `selecionaReceita(receita,datetime)`:

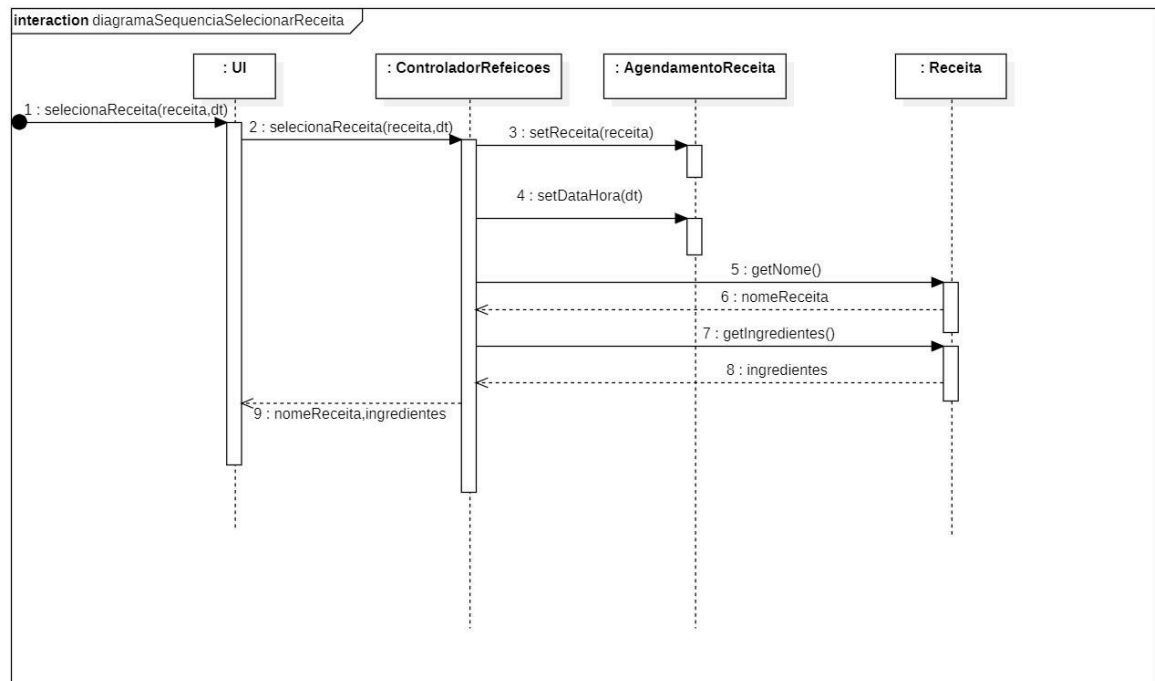


## Mensagem: verificaDisponibilidade(receita)



## 6.2 Diagrama de Sequência (Baixo Nível)

### Mensagem: selecionaReceita(receita,datetime):



## Mensagem: verificaDisponibilidade(receita)

