

# Documento de Requisitos do Software

<UAIQueFOME>

Versão <1.0>

## Sumário

<b>Sumário</b>	<b>1</b>
<b>1. Nome do software:</b>	<b>2</b>
<b>2. Objetivos do software:</b>	<b>2</b>
<b>3. Clientes e Stakeholders:</b>	<b>2</b>
<b>4. Escopo e limitações:</b>	<b>2</b>
4.1 – Escopo:	2
4.2 – Limitações:	3
<b>5. Equipe e recursos necessários:</b>	<b>3</b>
5.1 – Equipe:	3
5.2 – Linguagem de programação e ferramentas:	3
<b>6. Requisitos funcionais:</b>	<b>3</b>
<b>7. Requisitos não funcionais:</b>	<b>4</b>
<b>8. Diagrama de casos de uso:</b>	<b>5</b>
<b>9. Instruções de execução do protótipo:</b>	<b>6</b>
<b>10. Diagrama de Classes</b>	<b>7</b>
11. Diagrama de Atividades	8
<b>12. Diagramas de sequência</b>	<b>9</b>
RF.3 Manter promoções:	9
RF.4 Manter dados do usuário:	9
RF.5 Pesquisar restaurante:	10
RF.8 Favoritar restaurante:	10
RF.11 Realizar venda:	11
RF.12 Selecionar restaurante:	11
RF.13 Finalizar compra	12
CRUD: Read e Update	14
CRUD: Delete	15

## 1. Nome do software:

UAiQueFOME.

O acesso ao software pode ser realizado através do Link do GitHub: <https://github.com/Daniel13Caldeira/UAiQueFome>

## 2. Objetivos do software:

O software tem como objetivo permitir que restaurantes anunciem seus cardápios, organizem suas vendas e visualizem seu histórico de vendas, e permitir que usuários pesquisem por restaurantes por nome ou categoria, facilitando que ele decida qual será sua refeição.

## 3. Clientes e Stakeholders:

Os possíveis interessados nesse sistema são restaurantes que trabalham com delivery que se beneficiariam em publicar seus produtos em uma plataforma digital, alcançando um número maior de clientes.

## 4. Escopo e limitações:

### 4.1 – Escopo:

O sistema permitirá que sejam cadastrados perfis de restaurantes, que poderão gerenciar seus cardápios, promoções, pedidos feitos por clientes no sistema e informações do estabelecimento.

Permitirá também que sejam cadastrados usuários que desejem buscar por restaurantes, filtrá-los por categoria, visualizar seus cardápios, realizar pedidos de um ou mais pratos de um restaurante, salvar seus restaurantes favoritos e visualizar seu histórico de compras.

## 4.2 – Limitações:

O sistema não será responsável por gerenciar pagamentos: estes devem ser realizados no momento da entrega. Não será possível adicionar ao carrinho produtos de mais de um restaurante. O sistema não possuirá funcionalidade de visualização de endereços em mapa ou de planejamento de rotas de entrega.

## 5. Equipe e recursos necessários:

### 5.1 – Equipe:

O sistema será desenvolvido pelos alunos Igor Miranda Fam, Márcio Felipe Daniel Gonçalves e Daniel Jorge Reis Caldeira.

### 5.2 – Linguagem de programação e ferramentas:

O sistema será desenvolvido em Java, utilizando o Maven para gerenciar o projeto e suas dependências.

## 6. Requisitos funcionais:

**[RF 01] – Manter Produtos:** O restaurante pode adicionar, remover, editar os dados e visualizar os produtos.

**[RF 02] – Manter Pedidos (restaurantes):** O restaurante recebe os pedidos dos clientes e os prepara para a entrega.

**[RF 03] – Manter Promoções:** o restaurante deve poder colocar e retirar os produtos de uma promoção.

**[RF 04] – Manter dados do usuário:** O usuário pode visualizar e alterar seus dados pessoais, não será possível alterar o CPF.

**[RF 05] – Pesquisar e filtrar restaurantes:** O cliente pode pesquisar restaurantes pelo nome deles e pode filtrá-los por categorias.

**[RF 06] – Gerenciar carrinho e realizar pedidos (clientes):** O cliente pode adicionar e remover produtos ao carrinho. E quando quiser pode finalizar o seu pedido.

**[RF 07] – Visualizar histórico de compras:** O cliente pode visualizar seus pedidos anteriores.

**[RF 08] – Salvar restaurantes favoritos:** O cliente pode adicionar restaurantes aos favoritos.

**[RF 09] – Manter dados do restaurante:** O restaurante pode visualizar e alterar seus dados, não será possível alterar o nome e o CNPJ.

**[RF 10] – Visualizar histórico de vendas:** O restaurante pode visualizar o seu histórico de vendas.

**[RF 11] – Realizar venda:** O restaurante finaliza um pedido que estava em aberto.

**[RF 12] – Selecionar restaurante:** O cliente escolhe um restaurante para visualizar seus produtos.

**[RF 13] – Finalizar compra:** O cliente envia o pedido para o restaurante.

**[RF 14] – Login:** Todos os usuários devem fazer login para utilizar o sistema.

## 7. Requisitos não funcionais:

**[RNF 01] – Usabilidade:** O sistema deve ser fácil de se utilizar.

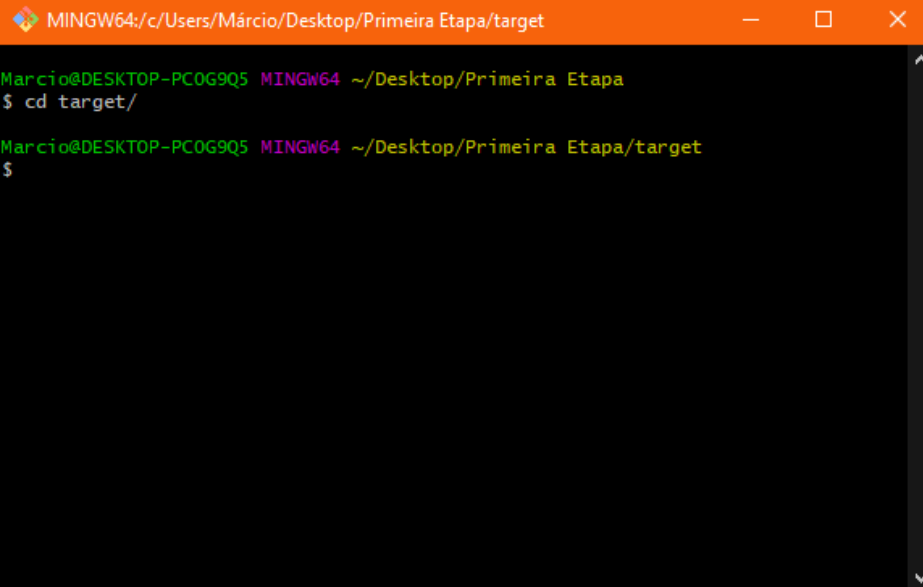
**[RNF 02] – Segurança:** O sistema não deixará que os usuários vejam dados confidenciais de outros usuários.

## 8. Diagrama de casos de uso:



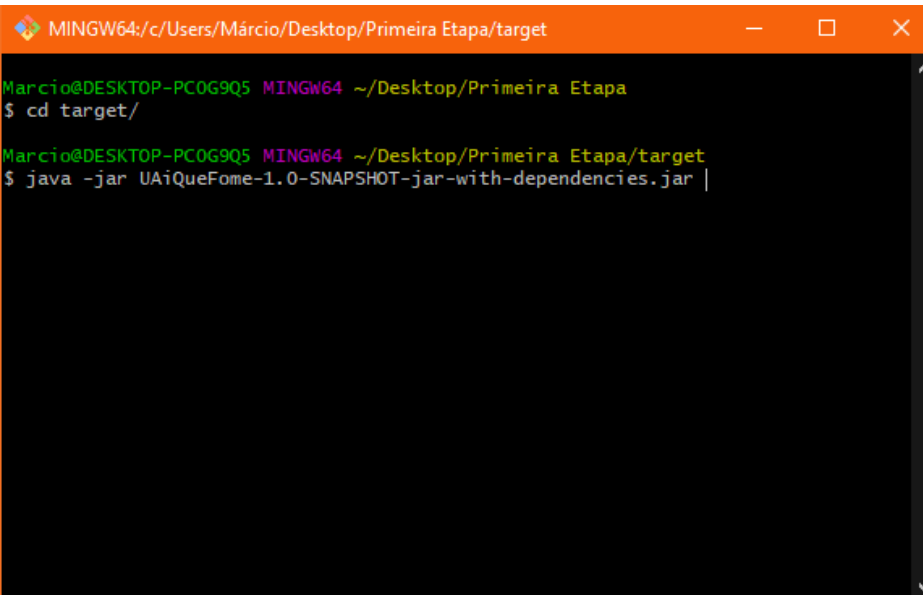
## 9. Instruções de execução do protótipo:

Para executar o programa com o maven faça:

A terminal window with an orange title bar containing the text 'MINGW64:/c/Users/Márcio/Desktop/Primeira Etapa/target'. The terminal shows the user 'Marcio@DESKTOP-PC0G9Q5' in a 'MINGW64' environment at the path '~/Desktop/Primeira Etapa'. The user enters '\$ cd target/' and the prompt changes to '~/Desktop/Primeira Etapa/target'.

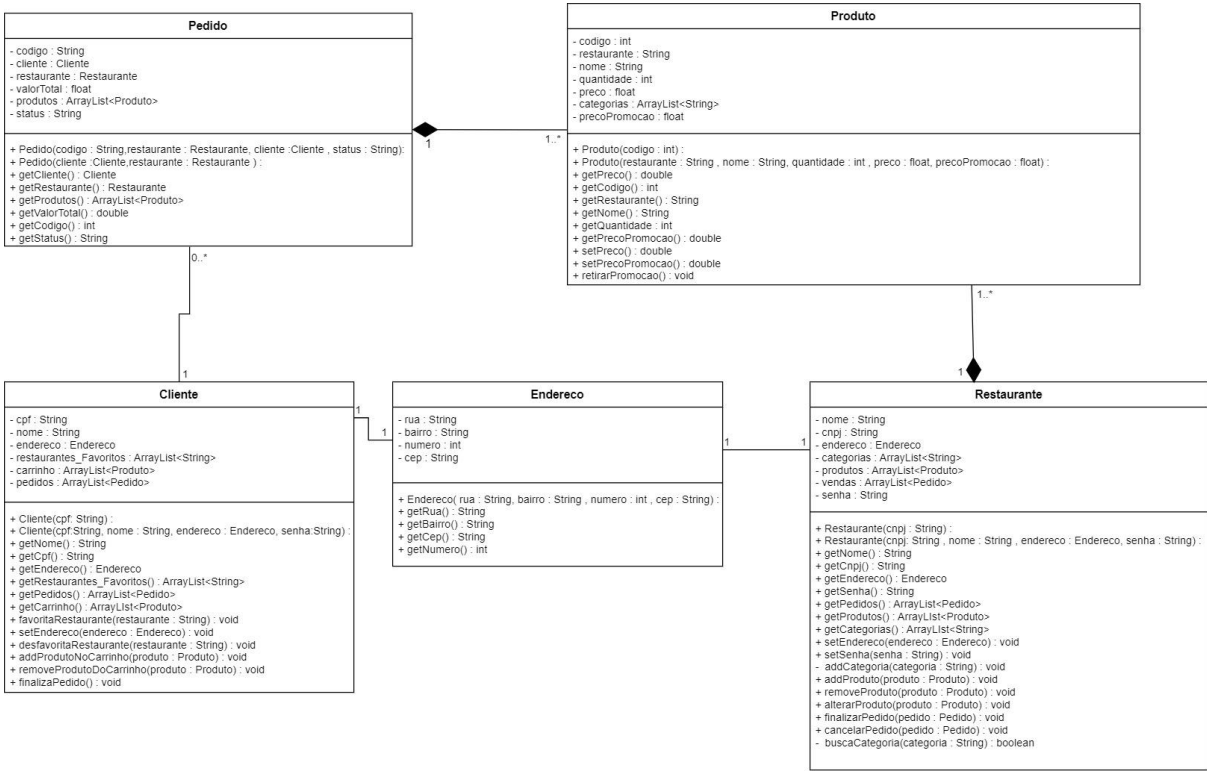
```
MINGW64:/c/Users/Márcio/Desktop/Primeira Etapa/target
Marcio@DESKTOP-PC0G9Q5 MINGW64 ~/Desktop/Primeira Etapa
$ cd target/
Marcio@DESKTOP-PC0G9Q5 MINGW64 ~/Desktop/Primeira Etapa/target
$
```

Após isso faça como na imagem abaixo para executar o programa.

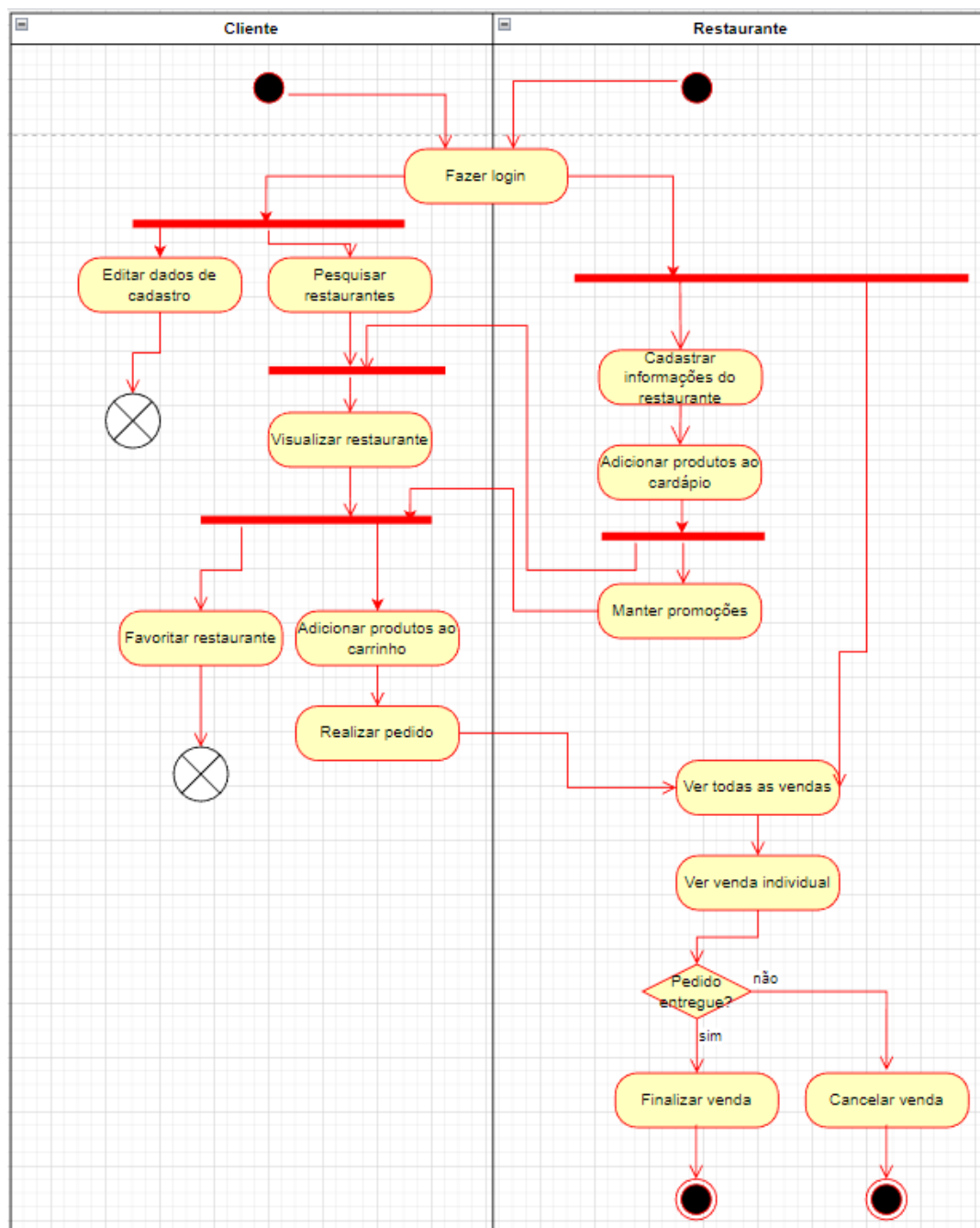
A terminal window with an orange title bar containing the text 'MINGW64:/c/Users/Márcio/Desktop/Primeira Etapa/target'. The terminal shows the user 'Marcio@DESKTOP-PC0G9Q5' in a 'MINGW64' environment at the path '~/Desktop/Primeira Etapa'. The user enters '\$ cd target/' and the prompt changes to '~/Desktop/Primeira Etapa/target'. Then, the user enters '\$ java -jar UAiQueFome-1.0-SNAPSHOT-jar-with-dependencies.jar' and the prompt returns to the root of the directory.

```
MINGW64:/c/Users/Márcio/Desktop/Primeira Etapa/target
Marcio@DESKTOP-PC0G9Q5 MINGW64 ~/Desktop/Primeira Etapa
$ cd target/
Marcio@DESKTOP-PC0G9Q5 MINGW64 ~/Desktop/Primeira Etapa/target
$ java -jar UAiQueFome-1.0-SNAPSHOT-jar-with-dependencies.jar |
$
```

# 10. Diagrama de Classes



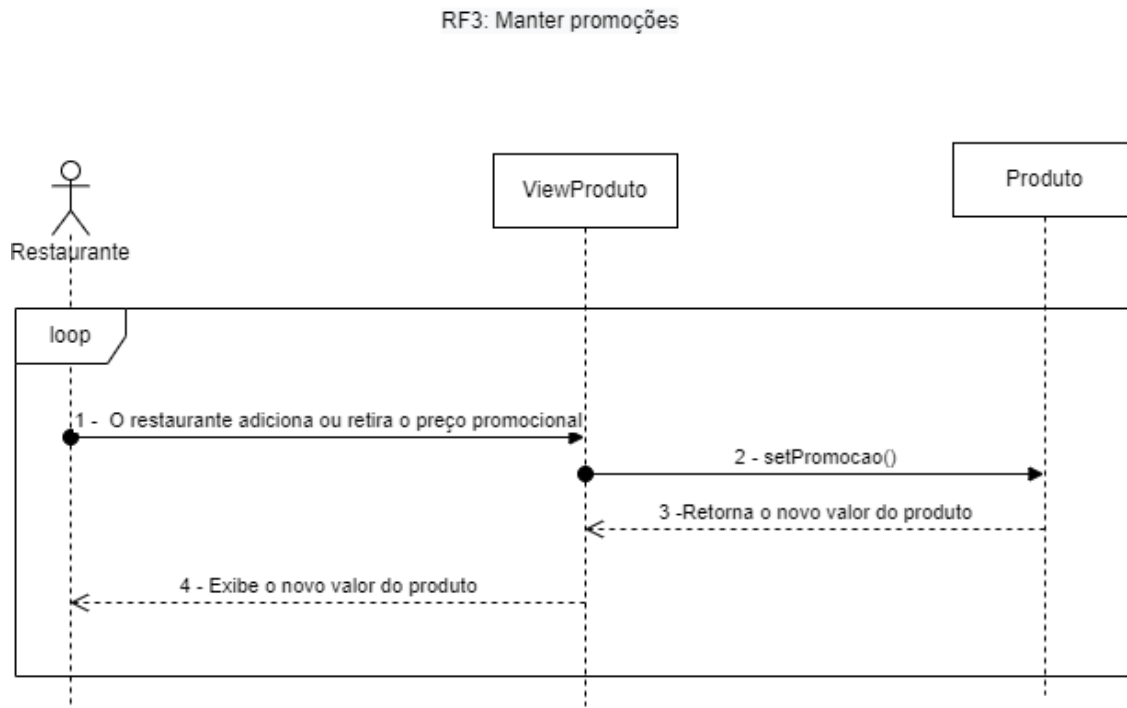
## 11. Diagrama de Atividades



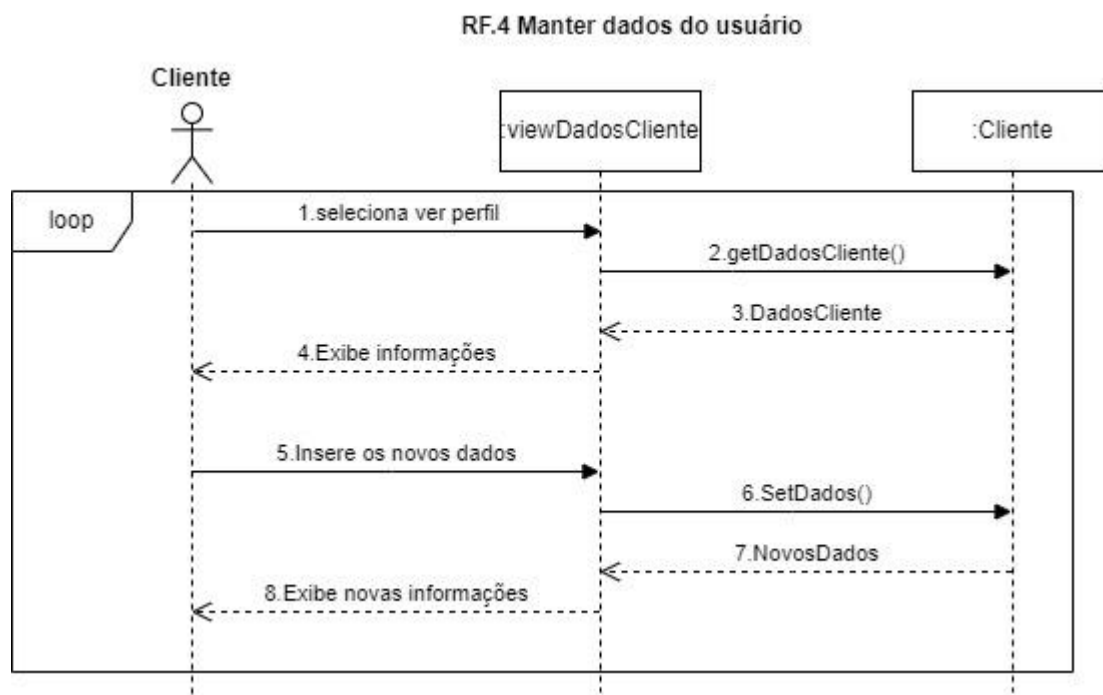


## 12. Diagramas de sequência

### RF.3 Manter promoções:

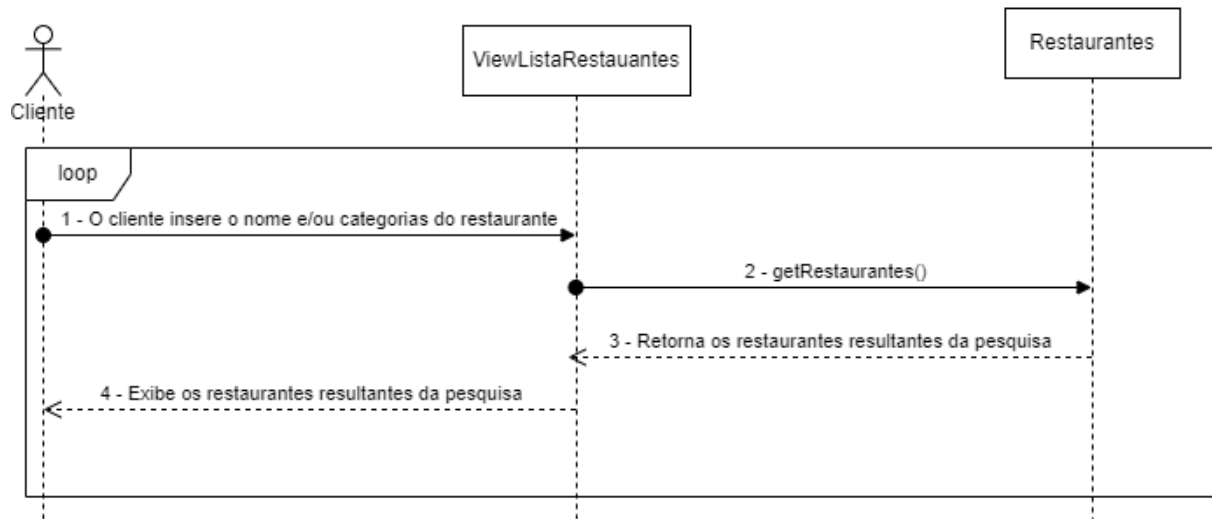


### RF.4 Manter dados do usuário:



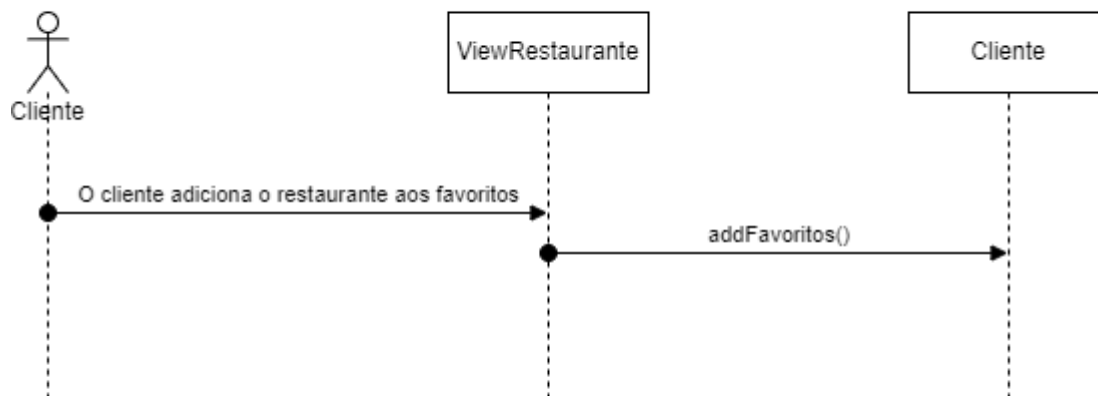
## RF.5 Pesquisar restaurante:

RF5: Pesquisar restaurante

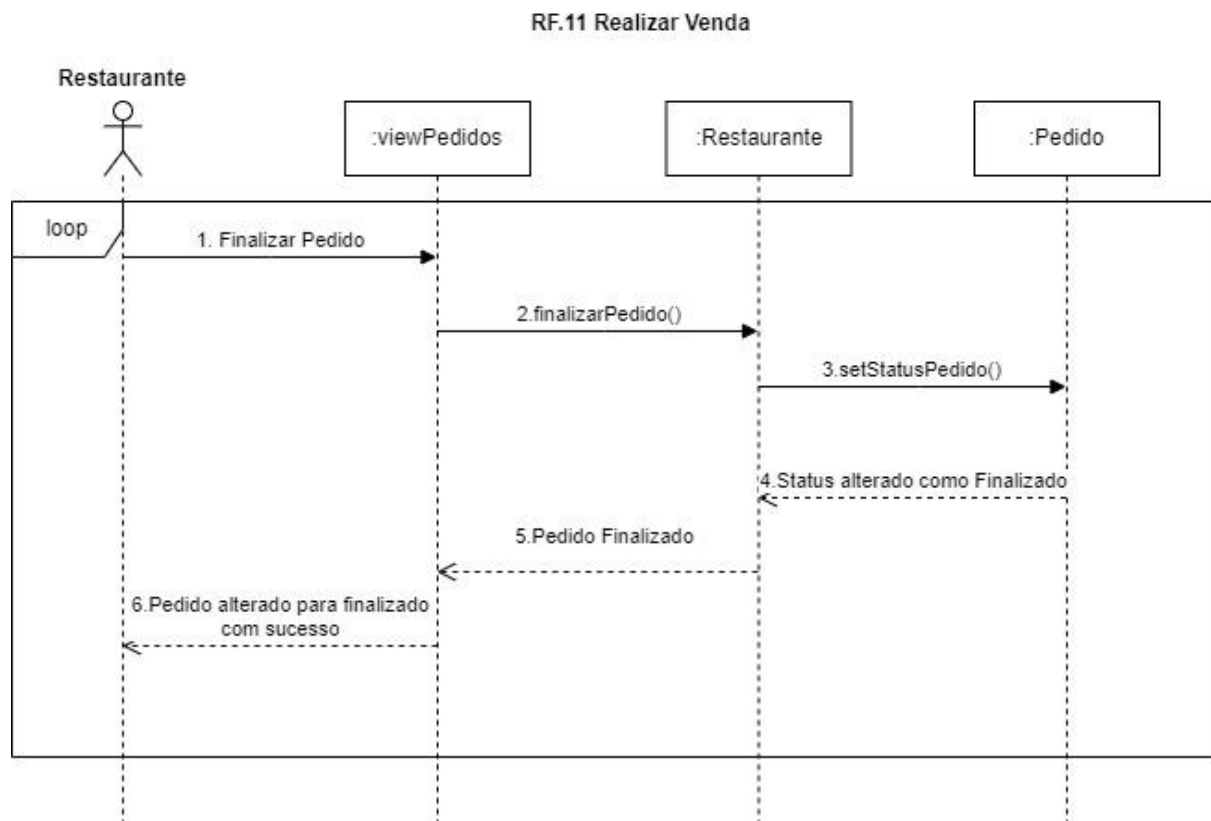


## RF 8 Favoritar restaurante:

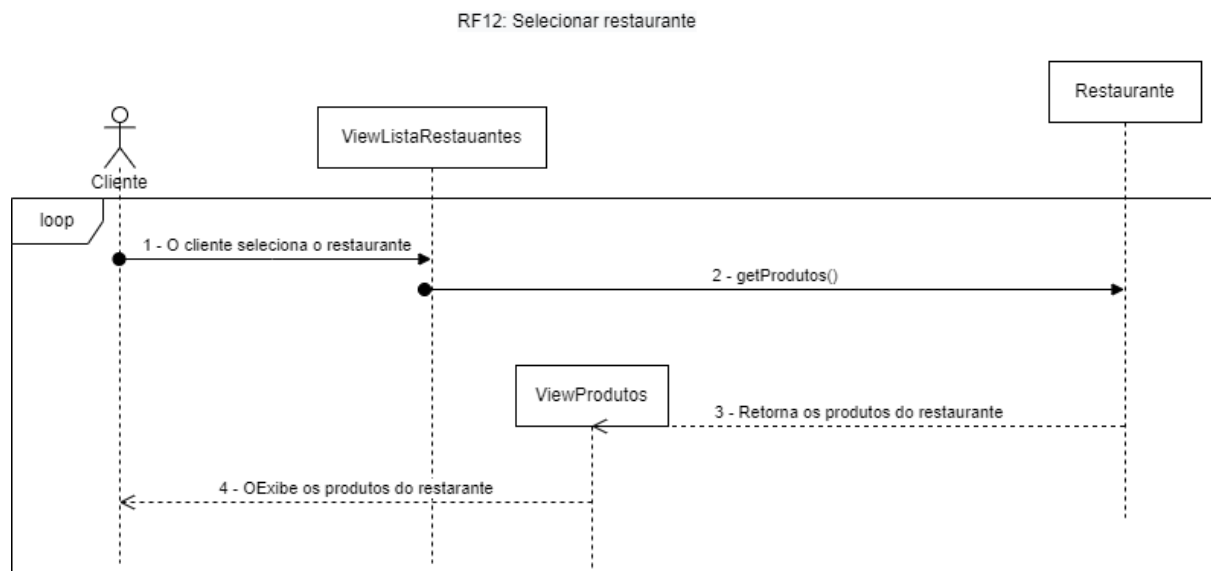
RF8: Favoritar restaurante



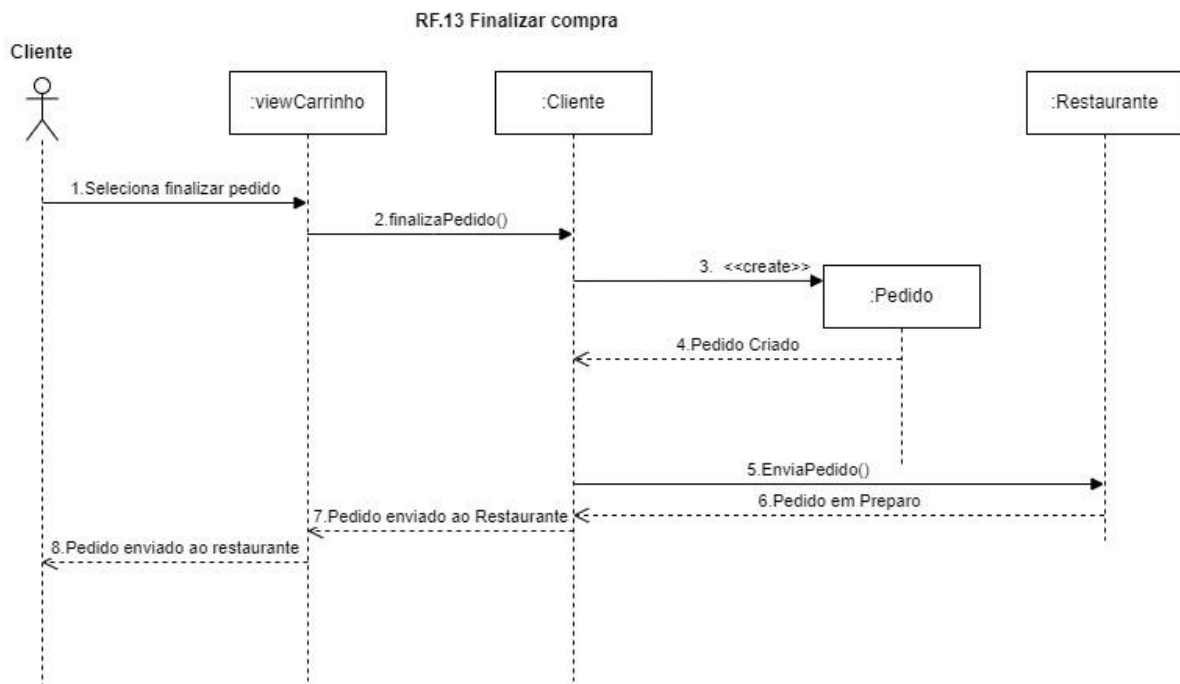
## RF.11 Realizar venda:



## RF.12 Selecionar restaurante:

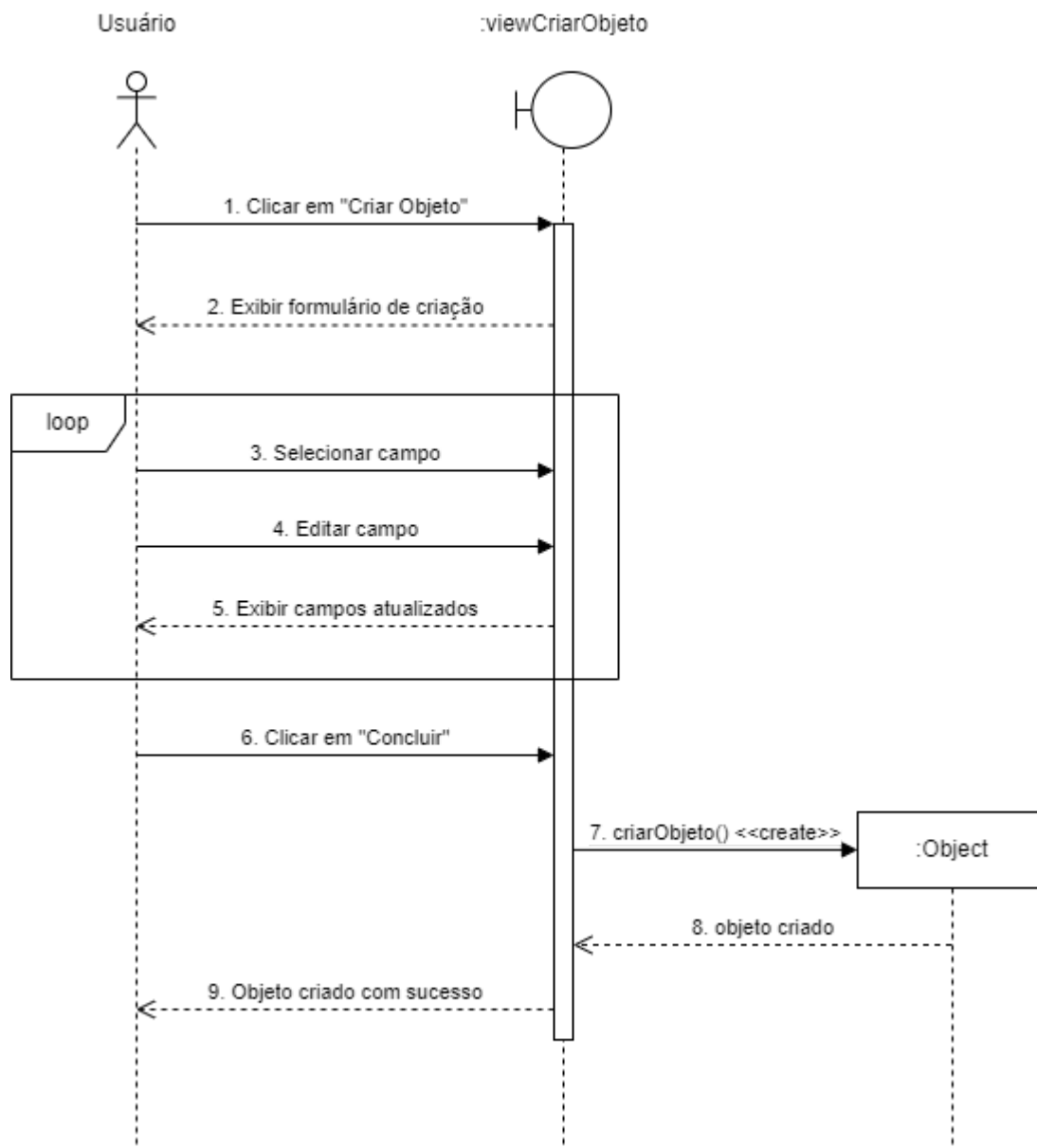


## RF.13 Finalizar compra

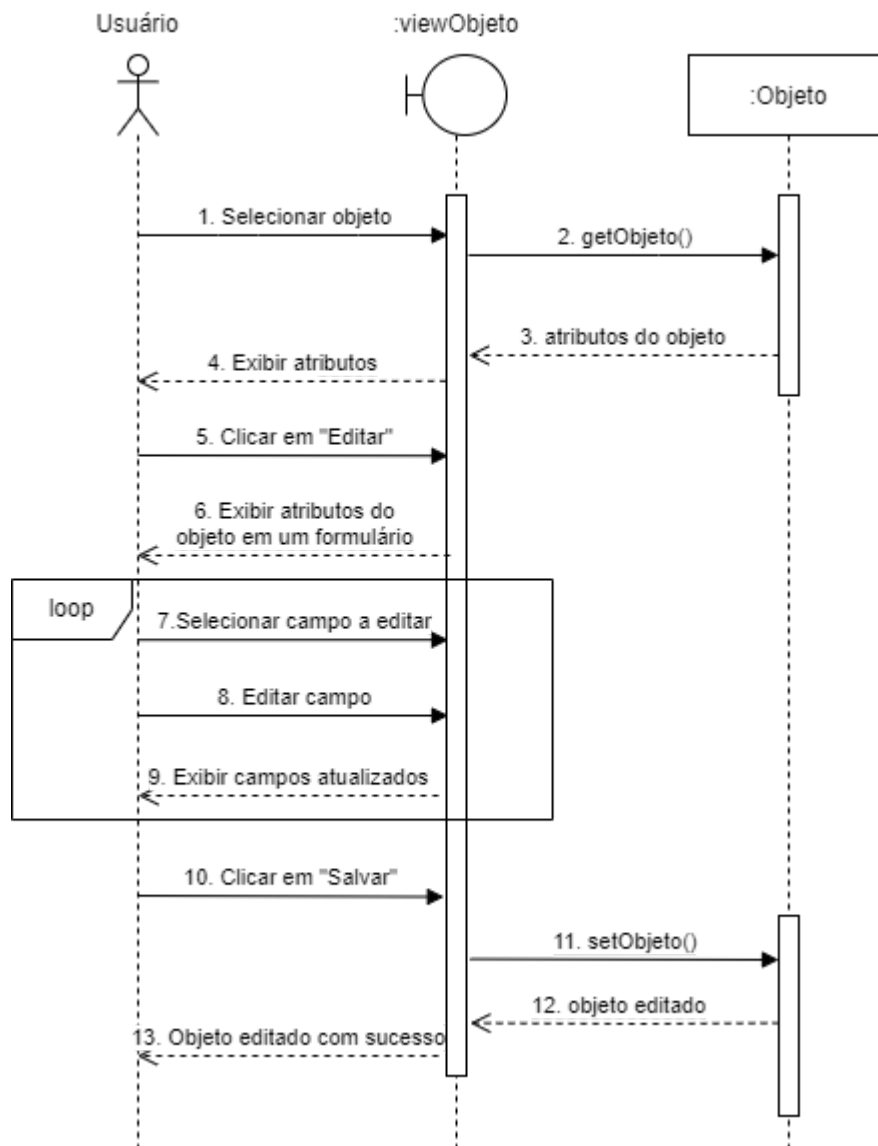


**\*CRUD: RF1, RF2, RF6, RF7, RF9 e RF10.**

**CRUD: Create**



## CRUD: Read e Update



## CRUD: Delete

