

## Introdução

Este documento tem como propósito servir como referência principal para a documentação do sistema ERP (Enterprise Resource Planning) a ser desenvolvido. Ele oferece uma visão abrangente e detalhada do sistema, alinhando o desenvolvimento às expectativas do cliente e garantindo que os requisitos sejam compreendidos de forma clara e objetiva.

O sistema consiste em um ERP destinado ao planejamento e gestão dos recursos de uma empresa, administrado pelos sócios. Suas principais funcionalidades incluem:

- Registro de produtos produzidos pela empresa;
- Controle de estoque;
- Cadastro de clientes e pedidos;
- Controle de entradas e saídas financeiras.

O público-alvo deste documento inclui futuros desenvolvedores que integrarão a equipe, analistas de negócio, membros da equipe de desenvolvimento e clientes finais. Ele visa facilitar a compreensão dos processos envolvidos, além de orientar e padronizar o desenvolvimento do sistema.

## Visão Geral do Sistema

O sistema ERP será projetado para otimizar a gestão dos recursos empresariais, com foco especial no controle de estoque e no gerenciamento de pedidos. Este se propõe como uma solução integrada que permitira a administração centralizada nos processos essenciais da empresa. Com sua implementação, esperamos trazer uma melhor eficiência no registro e controle de produtos, além de maior organização e visibilidade sobre o fluxo de pedidos e estoque.

Este sistema será importante para reduzir erros manuais e comunicativos, melhorar a precisão das informações e fornecer insights valiosos sobre a movimentação de produtos e finanças. Com a possibilidade de ferramentas para criação de relatórios detalhados, também ajudará na tomada de decisões estratégicas pelos sócios e gestores.

O sistema conta com as seguintes funcionalidades principais:

- Cadastro e controle de clientes:
  - Registro de clientes, com informações como cpf, nome, sobrenome, telefone, cep, endereço, e-mail.
- Cadastro e controle de produtos:
  - Registro detalhado de produtos, com informações como código, nome, descrição, preço, quantidade e categoria;
- Gerenciamento de estoque:

- Controle de entrada e saída de produtos, atualizando automaticamente os níveis de estoque.
- Emissão e controle de pedidos:
  - Criação e acompanhamento de pedidos de clientes, incluindo status e histórico de transações.
- Relatórios de inventário e movimentação de estoque\*:
  - Geração de relatórios personalizados para análise de inventário, movimentações possibilitando acompanhar tendências.
- Notificação de baixo estoque\*:
  - Alertas automáticos para produtos com níveis de estoque abaixo do mínimo configurado.

As principais operações do sistema incluem:

- CRUD para clientes
  - Cadastrar novo cliente
  - Consultar clientes e seus pedidos
  - Atualizar dados do cliente
  - Exclusão de cliente não poderá ser feita diretamente
- CRUD para produtos
  - Criar registros de produtos
  - Consultar informações sobre produtos existentes
  - Atualizar detalhes como, preço, forma de pagamento, quantidade e descrição
  - Excluir produtos obsoletos ou descontinuados
- CRUD para pedidos
  - Criar pedidos de clientes
  - Consultar o histórico e status de pedidos
  - Atualizar pedidos com informações adicionais ou alterações
  - Excluir pedidos cancelados
- Controle de estoque
  - Registro automático de entradas e saídas durante operações de vendas ou reposição
  - Atualização em tempo real dos níveis de estoque
- Relatórios personalizados\*
  - Geração de relatórios e exportação em formatos como PDF e Excel

# Requisitos do Sistema

## Requisitos Funcionais

### Clientes:

- **Cadastro:** Deve incluir um CPF único, nome completo, telefone com DDD, endereço contendo Rua, Bairro, CEP, cidade, estado e Email, forma de pagamento.
- **Consultas:** Deve retornar os dados detalhados do cliente.
- **Atualizações:** Somente serão realizadas partindo do CPF do cliente.
- **Exclusão:** Solicitada através do CPF do cliente, e com a obrigatoriedade de uma justificativa. Não poderá ser efetuada caso exista pedidos em andamento para este CPF.

### Produtos:

- **Cadastro:** Deve incluir nome, código, preço, uma forma de pagamento, quantidade e uma ou mais categorias.
- **Consultas:** Devem retornar todas as informações solicitadas de acordo com o filtro, incluindo estoque, saídas e entradas.
- **Atualizações:** Serão realizadas com base no código do produto. A data de validade, uma vez cadastrada, não poderá ser alterada diretamente. Estoques negativos deverão ser exibidos como "falta em estoque".
- **Exclusão:** Solicitada através do código do produto, com a obrigatoriedade de uma justificativa para o cancelamento ou descontinuação da produção.

### Pedidos:

- **Cadastro:** Deve incluir cliente, registro automático da data, ao menos um ou mais produtos, valor total e nível de urgência.
- **Consultas:** Devem listar todos os pedidos ou exibir detalhes específicos, como data, estado atual, produtos associados, cliente, preço total e data de finalização (se finalizado).
- **Atualizações:** Realizadas com base no estado do pedido, como "cadastrado", "em produção", "em separacao", "cancelado" ou "finalizado". Detalhes adicionais e comentários poderão ser adicionados durante o andamento. O nível de urgência poderá ser ajustado manualmente mediante justificativa ou automaticamente conforme o tempo de demora.
- **Exclusão:** Permitida em casos de cancelamento do pedido.

### Estoque:

- **Reposição:** Atualizações automáticas baseadas no ID do produto, tanto para reposição quanto para remoção após vendas.
- **Validade:** Produtos fora da validade gerarão alertas e não serão contabilizados no estoque ativo, mas permanecerão visíveis em uma lista específica.

- **Atualizações em tempo real:** Permitirão acompanhamento simultâneo para evitar conflitos de pedidos ou vendas sem estoque, além de auxiliar na produção conforme a demanda.

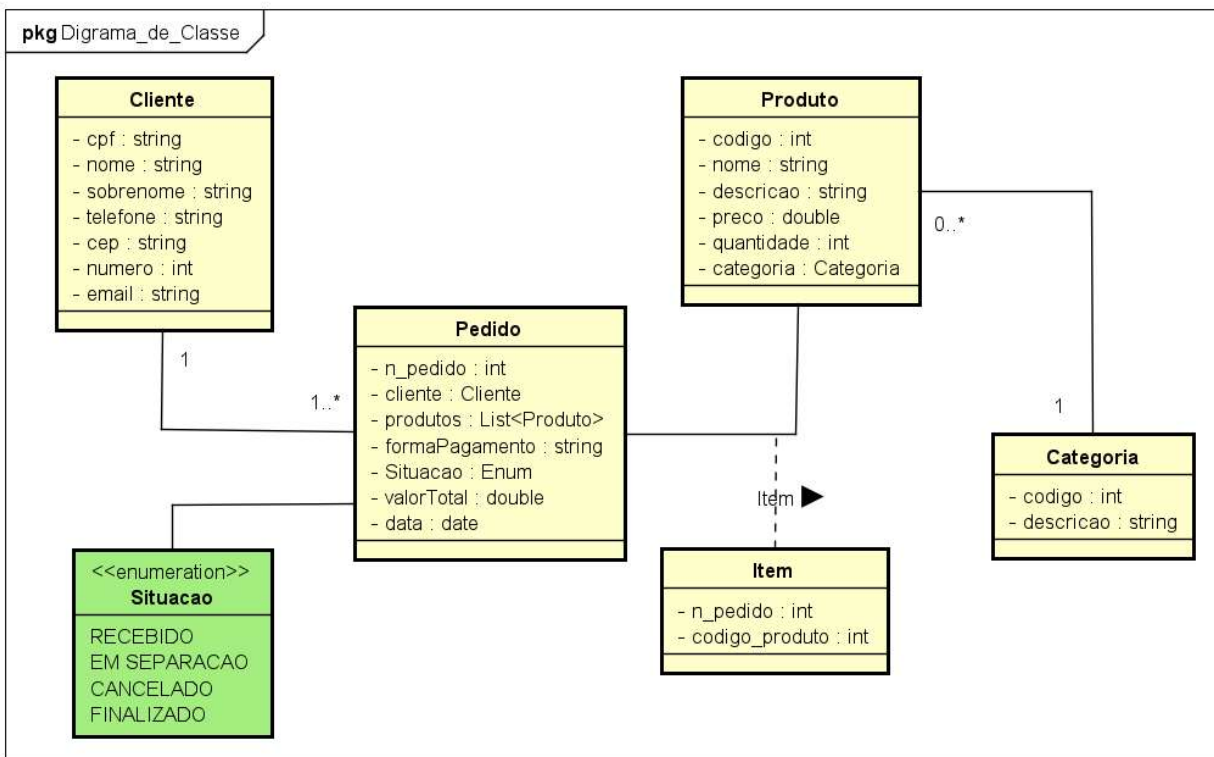
## Relatórios:

- **Geração:** Relatórios analíticos baseados em filtros como pedidos por período, clientes, entradas e saídas de produtos.
- **Exportação:** Disponível em formatos PDF e Excel.

## Requisitos Não Funcionais

- **Desempenho:** O sistema deve suportar um alto volume de movimentações sem degradação de desempenho. (O limite será definido posteriormente após testes).
- **Usabilidade:** Interface clara e intuitiva, acessível para usuários com diferentes níveis de conhecimento técnico.
- **Segurança:** Implementação de criptografia para dados sensíveis, garantindo a integridade e confidencialidade das informações armazenadas.

## Diagrama de Classe



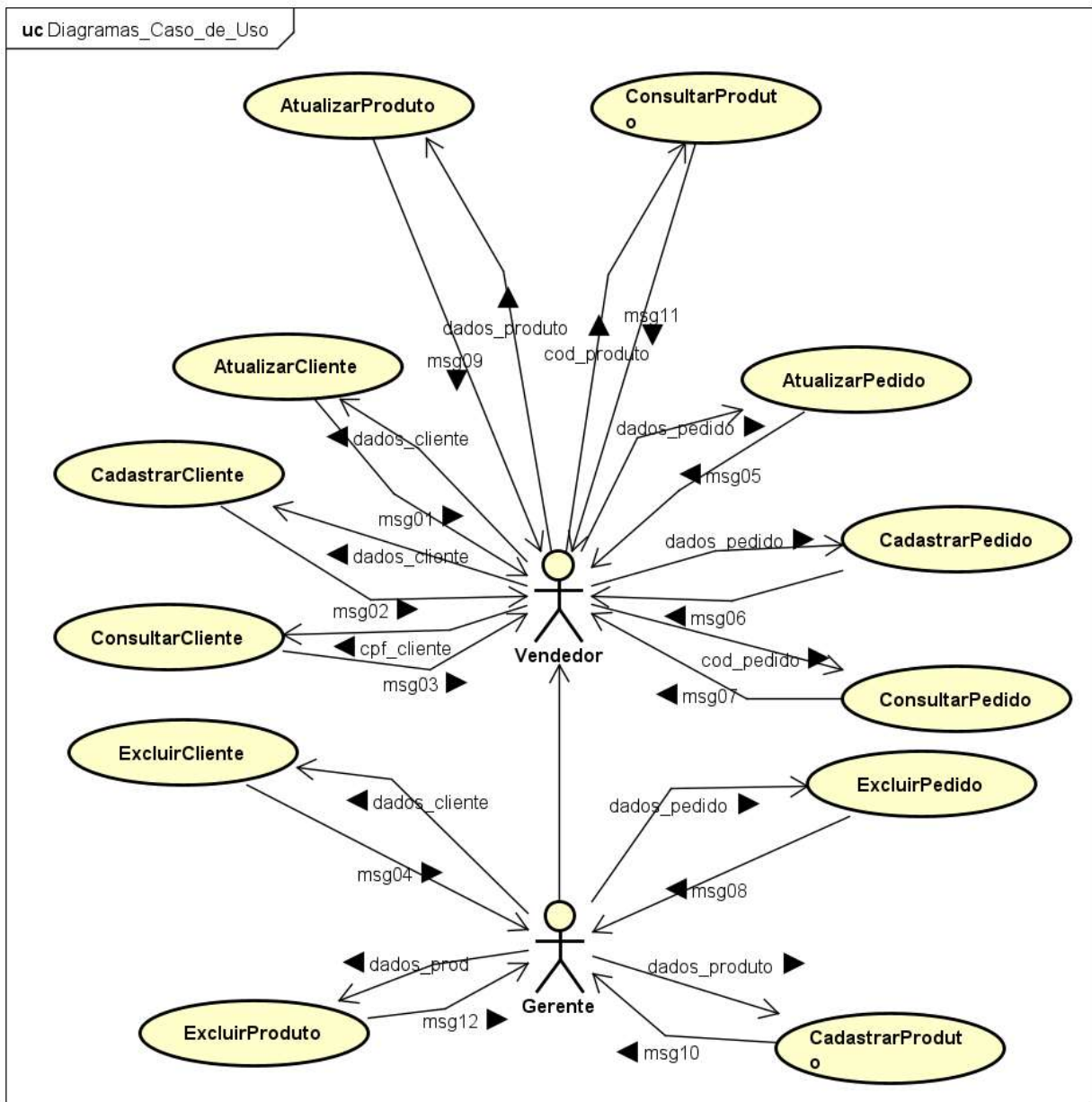
## Descrição

Um PEDIDO deve conter apenas um cliente, e um ou vários PRODUTO. Um CLIENTE pode ter um ou vários pedidos. Um PRODUTO deve conter uma CATEGORIA e pode pertencer a nenhum ou vários PEDIDO.

### 1. Lista de Eventos

Nr	Descrição	Tipo Fluxo	Estímulo	Ação	Resposta	Usuário
1	Vendedor Atualiza cliente	F	Dados_cliente	Atualizar_dados_Cliente	Msg01	Vendedor
2	Vendedor Cadastra Cliente	F	Dados_cliente	Cadastrar_cliente	Msg02	Vendedor
3	Vendedor consulta Cliente	F	Cpf_cliente	Consultar_cliente	Msg03	Vendedor
4	Gerente exclui Cliente	F	Dados_cliente	Excluir_cliente	Msg04	Gerente
5	Vendedor atualiza Pedido	F	Dados_pedido	Atualizar_pedido	Msg05	Vendedor
6	Vendedor cadastra Pedido	F	Dados_pedido	Cadastrar_pedido	Msg06	Vendedor
7	Vendedor consulta Pedido	F	N_pedido	Consultar_pedido	Msg07	Vendedor
8	Gerente exclui Pedido	F	N_pedido	Excluir_pedido	Msg08	Gerente
9	Vendedor atualiza Produto	F	Dados_produto	Atualizar_produto	Msg09	Vendedor
10	Gerente cadastra Produto	F	Dados_produto	Cadastrar_produto	Msg10	Gerente
11	Vendedor consulta Produto	F	Código_produto	Consultar_produto	Mgs11	Vendedor
12	Gerente exclui Produto	F	Codigo_produto	Excluir_produto	Msg12	Gerente

## Diagrama de Caso de Uso (Use Case)



### Miniespecificação Evento 01 –

```
RECEBER dados_cliente
LOCALIZAR dados_cliente EM CLIENTES
    SE dados_cliente NÃO EXISTIR FACA
        EMITIR msg01 (“Cliente não encontrado – Verifique os dados”)
    SENAO
        ATUALIZAR dados_cliente EM CLIENTES
        EMITIR msg01 (“Dados do Cliente Atualizados com sucesso!”)
FIMSE
```

### Miniespecificação Evento 02 –

```
RECEBER dados_cliente
LOCALIZAR dados_cliente EM CLIENTES
    SE dados_cliente NÃO EXISTIR FACA
        GUARDAR dados_cliente EM CLIENTES
        EMITIR msg02 (“Cliente cadastrado com sucesso”)
    SENAO
        EMITIR msg02 (“Cliente já esta cadastrado no sistema”)
FIMSE
```

### Miniespecificação Evento 03 –

```
RECEBER cpf_cliente
LOCALIZAR cpf_cliente EM CLIENTES
    SE cpf_cliente NÃO ENCONTRADO EM CLIENTES FACA
        EMITIR msg03 (“Cliente não encontrado – Verifique os dados”)
    SENAO
        EMITIR msg03 (“Cliente encontrado: “ + dados_cliente )
FIMSE
```

### Miniespecificação Evento 04 –

```
RECEBER dados_cliente
LOCALIZAR dados_cliente em CLIENTES
    SE dados_cliente NÃO EXISTIR EM CLIENTES FACA
        EMITIR msg04 (“Cliente não encontrado – Verifique os dados”)
```

```
        SENAO
            VERIFICAR dados_cliente EM PEDIDOS
                SE EXISTIR
                    EMITIR msg04 ("Cliente com pedido em andamento - Verifique
antes de remover")
                SENAO
                    REMOVER dados_cliente DE CLIENTES
                    EMITIR msg04 ("Cliente removido com sucesso")
        FIMSE
```

#### Miniespecificação Evento 05 –

```
RECEBER dados_pedido
LOCALIZAR dados_pedido EM PEDIDOS
    SE dados_pedido NÃO EXISTIR FAÇA
        EMITIR msg05 ("Pedido não encontrado – Verifique os dados")
    SENAO
        SUBSTITUIR dados_pedido EM PEDIDOS
        EMITIR msg05 ("Pedido atualizado com sucesso!")
    FIMSE
```

#### Miniespecificação Evento 06 –

```
RECEBER dados_pedido
LOCALIZAR dados_pedido EM PEDIDOS
    SE dados_pedido NÃO EXISTIR FAÇA
        GUARDAR dados_pedido EM PEDIDOS
        EMITIR msg06 ("Pedido registrado com sucesso")
    SENAO
        EMITIR msg06 ("Já existe este mesmo pedido registrado no sistema")
    FIMSE
```



### Miniespecificação Evento 07 –

RECEBER cod\_pedido

LOCALIZAR cod\_pedido EM PEDIDOS

SE cod\_pedido NÃO EXISTIR EM PEDIDOS FAÇA

EMITIR msg07 (“Pedido não encontrado – Verifique os dados”)

SENAO

EMITIR msg07 (“Pedido encontrado: “ + dados\_pedido)

FIMSE

### Miniespecificação Evento 08 –

RECEBER dados\_pedido

LOCALIZAR dados\_pedido EM PEDIDOS

SE dados\_pedido NÃO EXISTIR EM PEDIDOS FAÇA

EMITIR msg08 (“Pedido não encontrado – Verifique os dados”)

SENAO

REMOVER dados\_pedido DE PEDIDOS

EMITIR msg09 (“Pedido removido com sucesso”)

FIMSE

### Miniespecificação Evento 09 -

RECEBER dados\_produto

LOCALIZAR dados\_produto EM PRODUTOS

SE dados\_produto NÃO EXISTIR FAÇA

EMITIR msg09 (“Produto não encontrado – Verifique os dados” )

SENAO

ATUALIZAR dados\_produto EM PRODUTOS

EMITIR msg09 ( “Produto atualizado com sucesso”)

FIMSE

#### Miniespecificação Evento 10 –

RECEBER dados\_produto

LOCALIZAR dados\_produto EM PRODUTOS

SE dados\_produto NÃO EXISTIR FACA

GUARDAR dados\_produto EM PRODUTOS

EMITIR msg10 ( “Produto cadastrado com sucesso”)

SENAO

EMITIR msg10 ( “Produto já existente no banco de dados”)

FIMSE

#### Miniespecificação Evento 11 –

RECEBER cod\_produto

LOCALIZAR cod\_produto EM PRODUTOS

SE cod\_produto NÃO EXISTIR FACA

EMITIR msg11 ( “Produto não encontrado – Verifique os dados”)

SENAO

EMITIR msg11( “Produto encontrado: “+ dados\_produto)

FIMSE

#### Miniespecificação Evento 12 –

RECEBER dados\_produto

LOCALIZAR dados\_produto EM PRODUTOS

SE dados\_produto NÃO EXISTIR FACA

EMITIR msg12 ( “Produto não encontrado – Verifique os dados”)

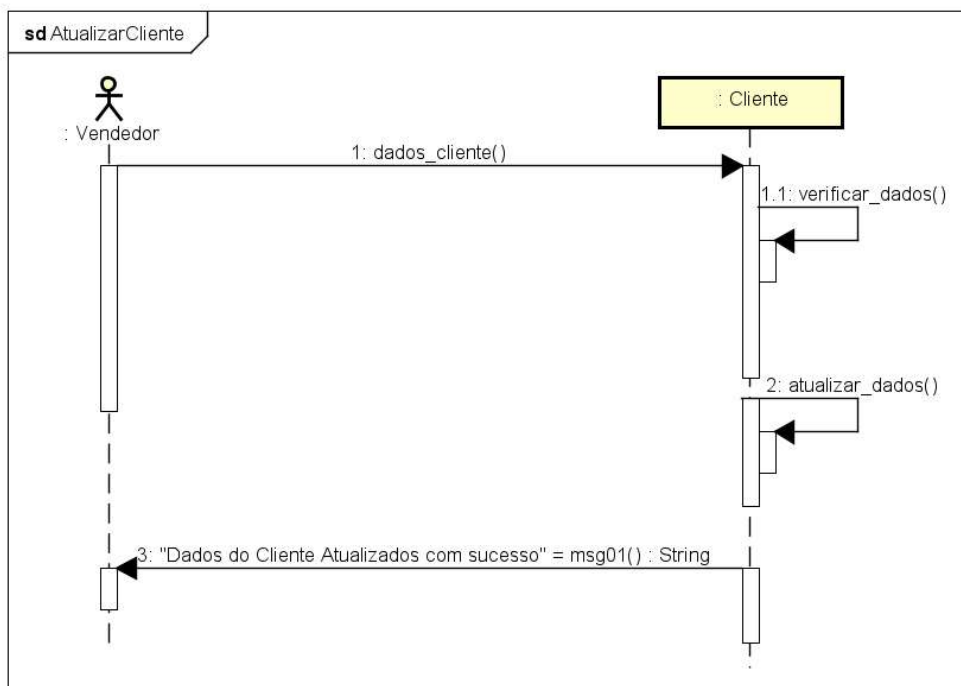
SENAO

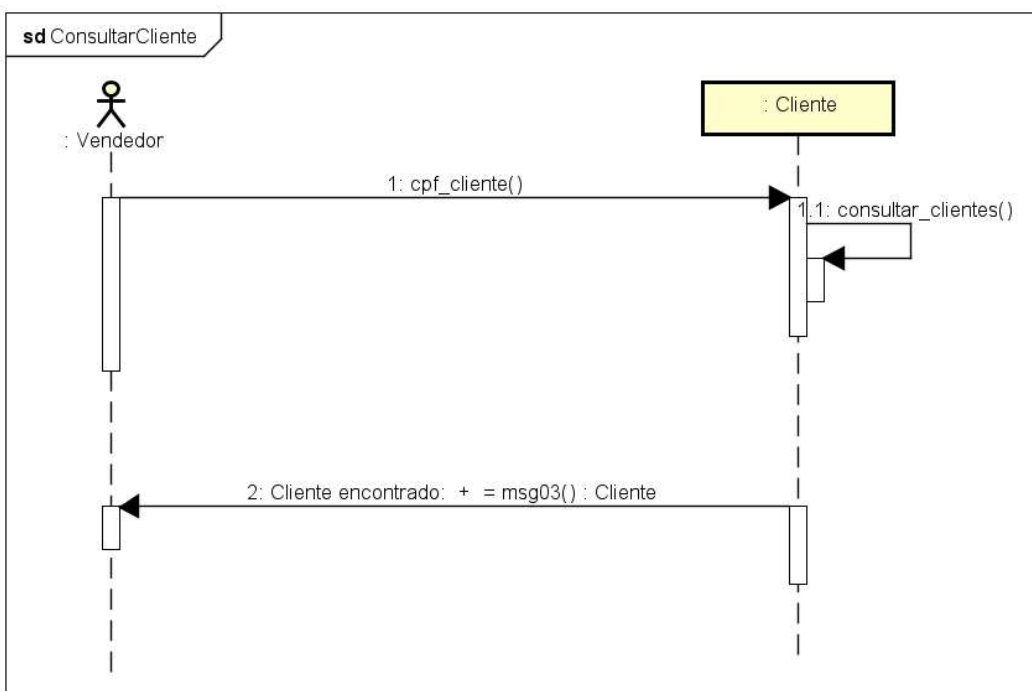
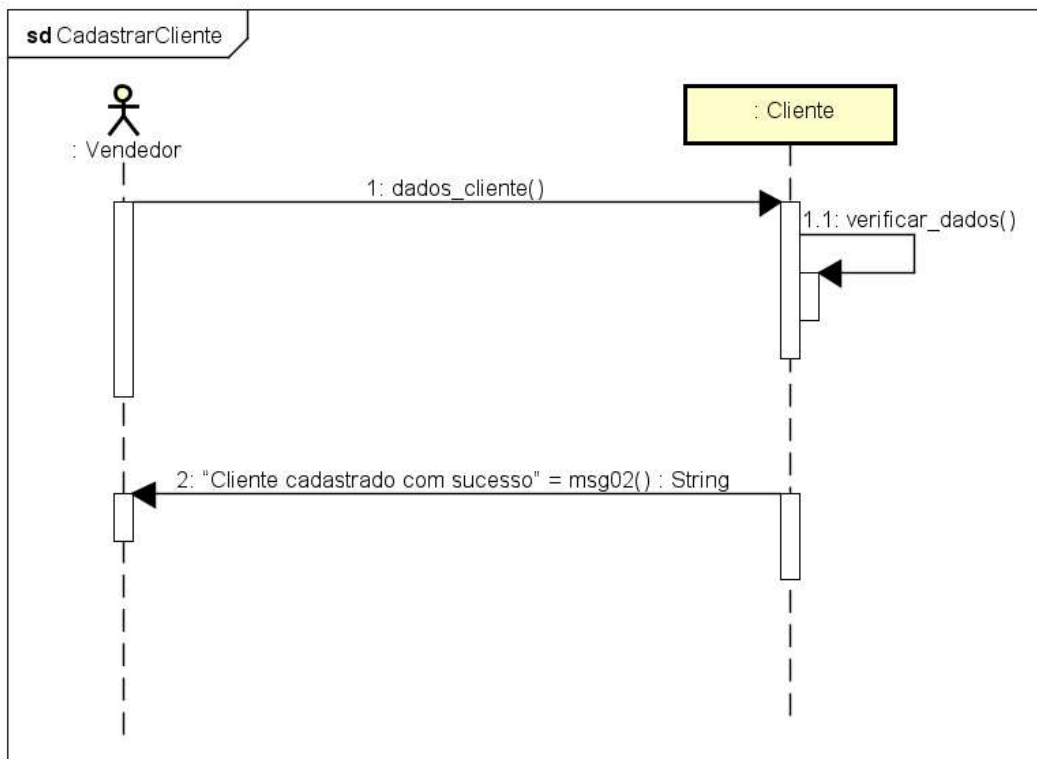
REMOVER dados\_produto DE PRODUTOS

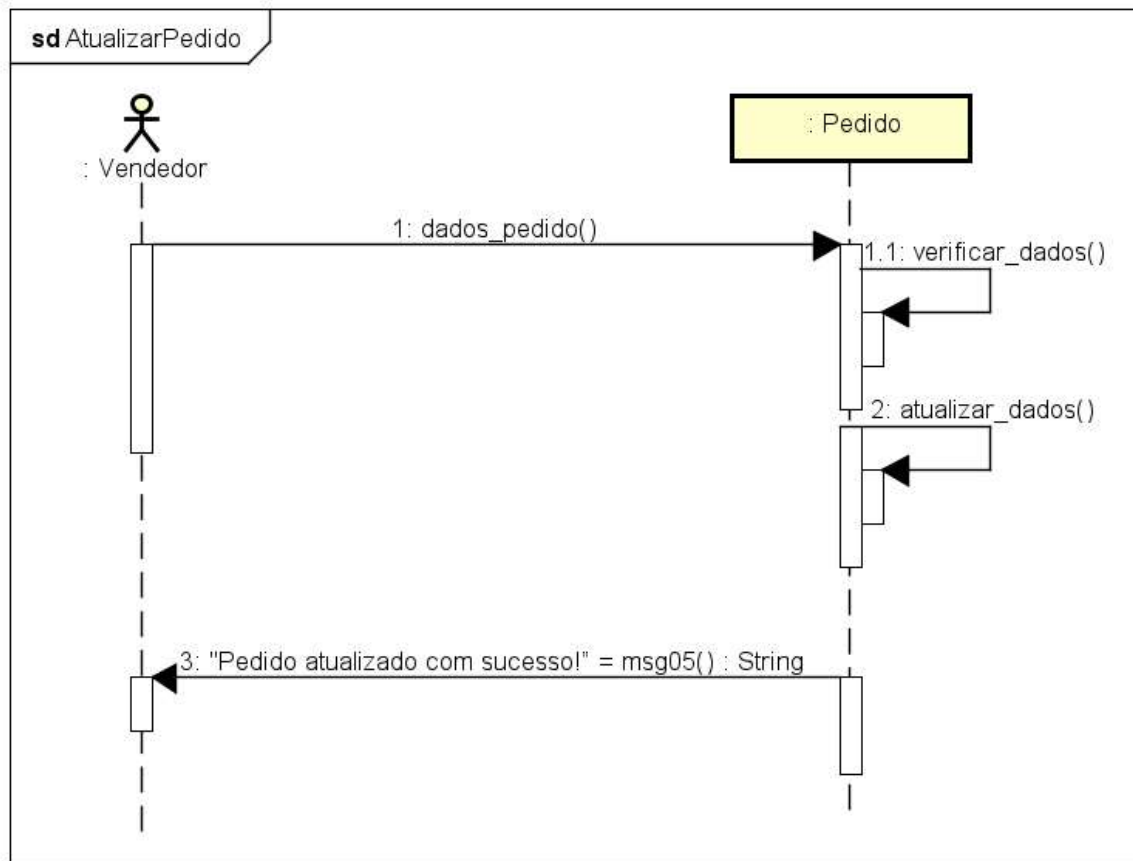
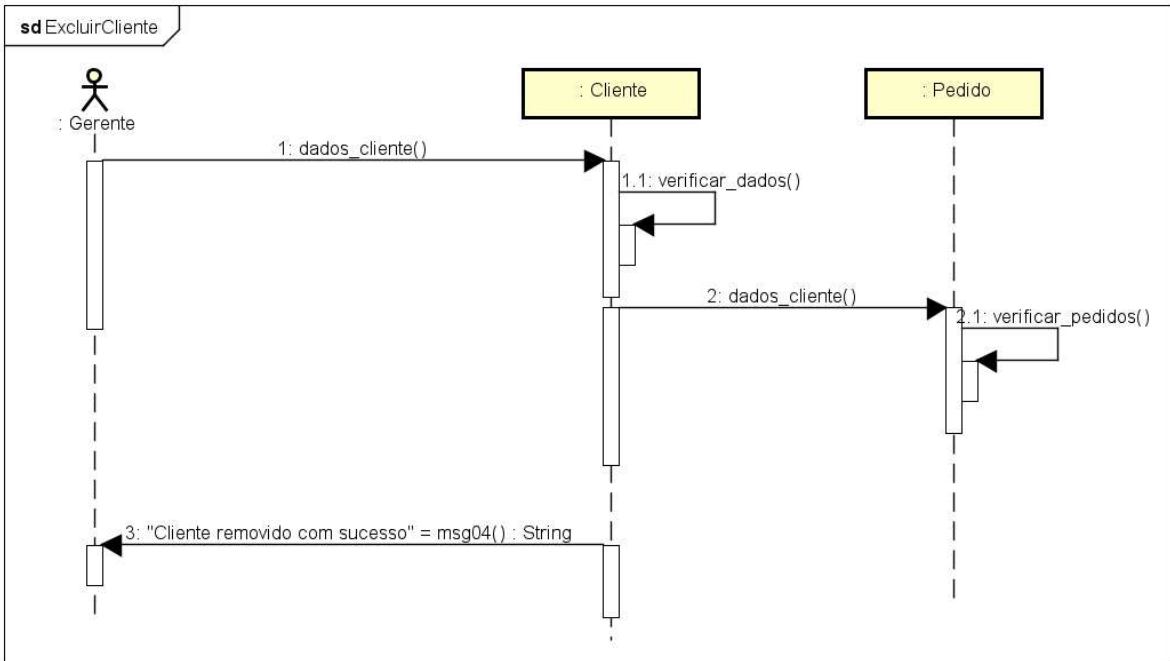
EMITIR msg12 (“Produto removido com sucesso”)

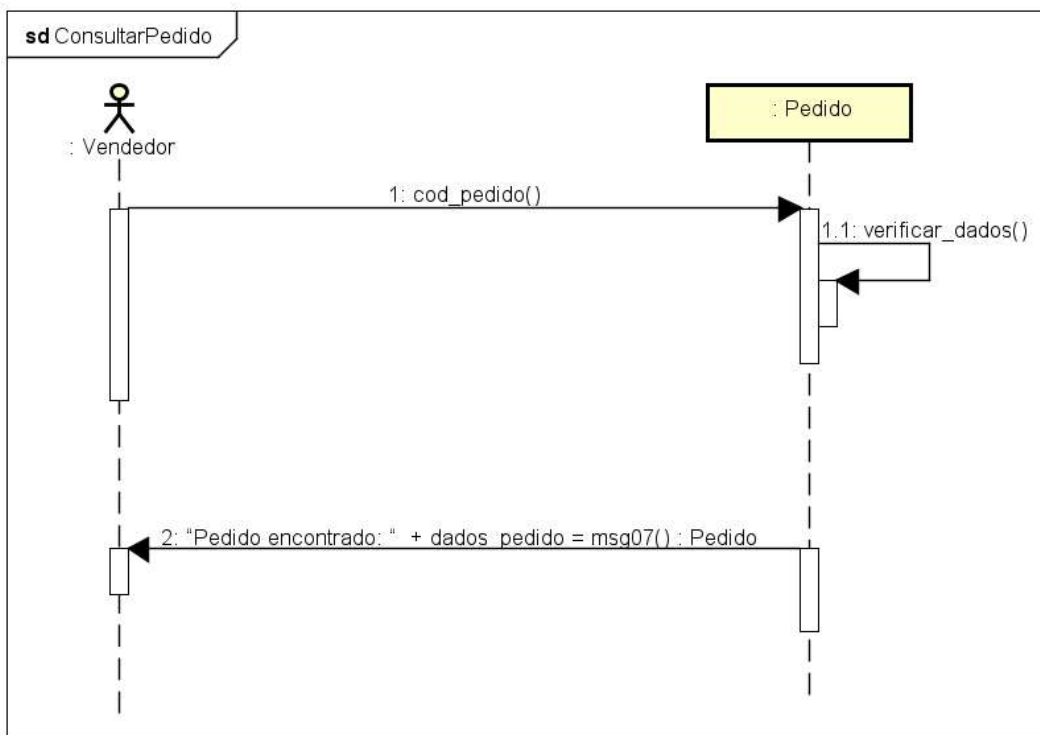
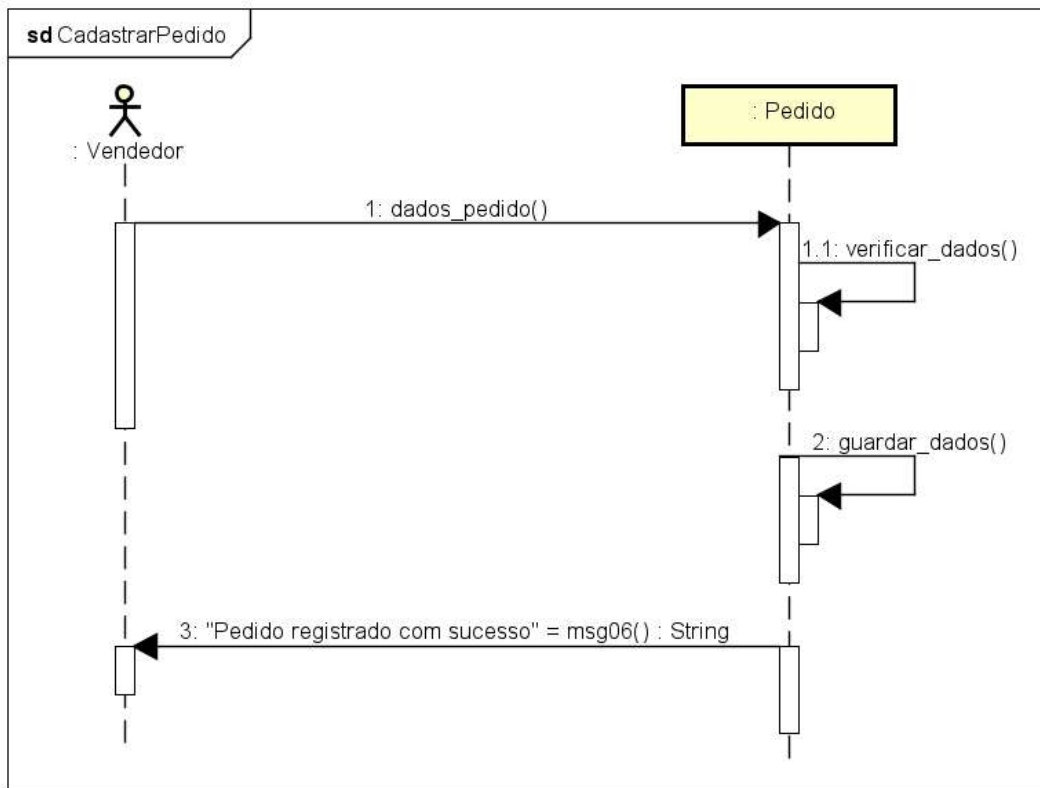
FIMSE

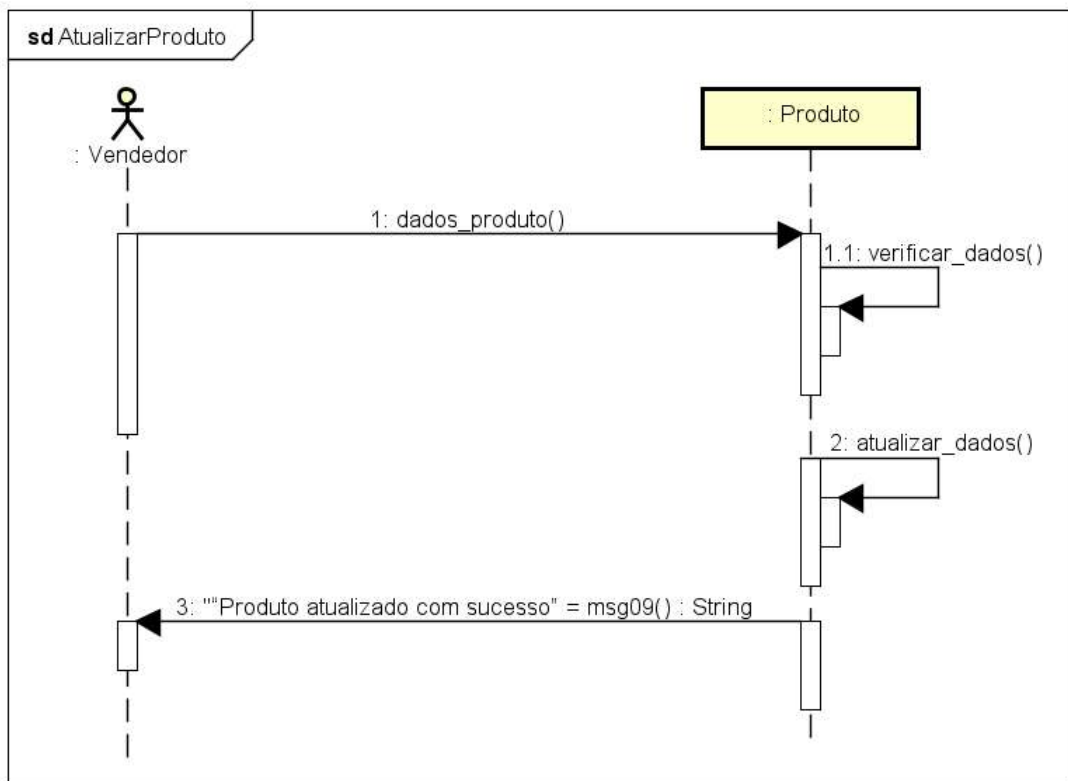
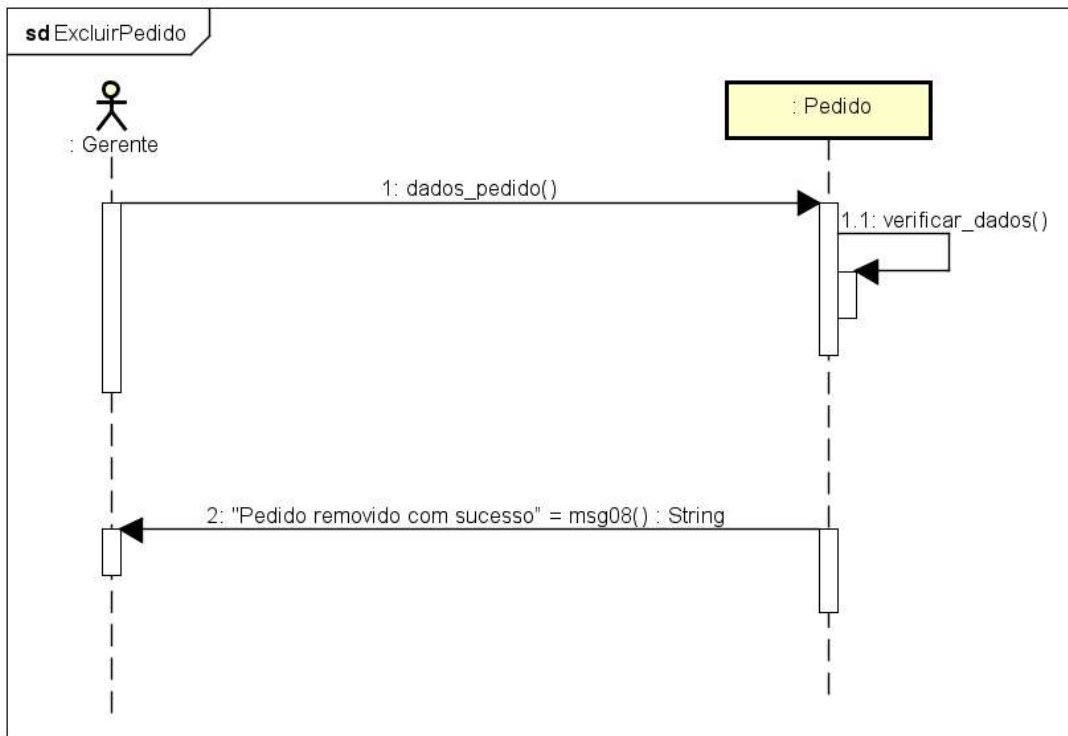
## Diagramas de sequência por Evento

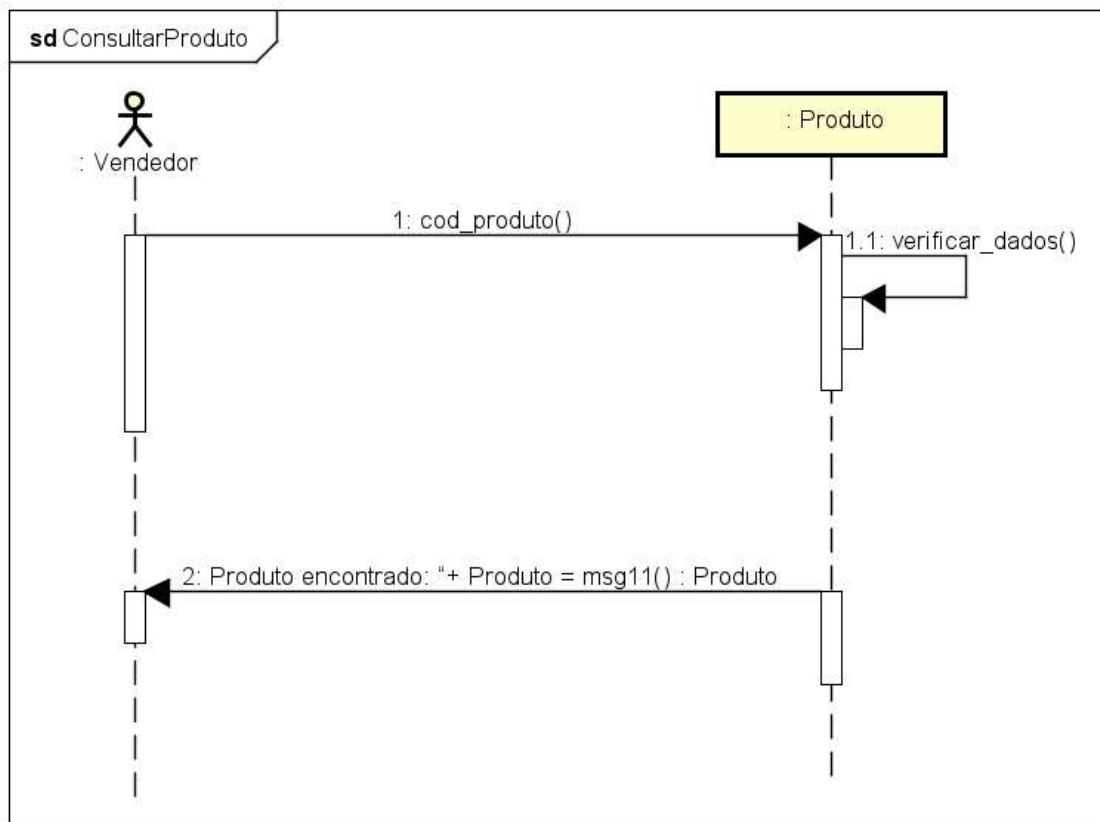
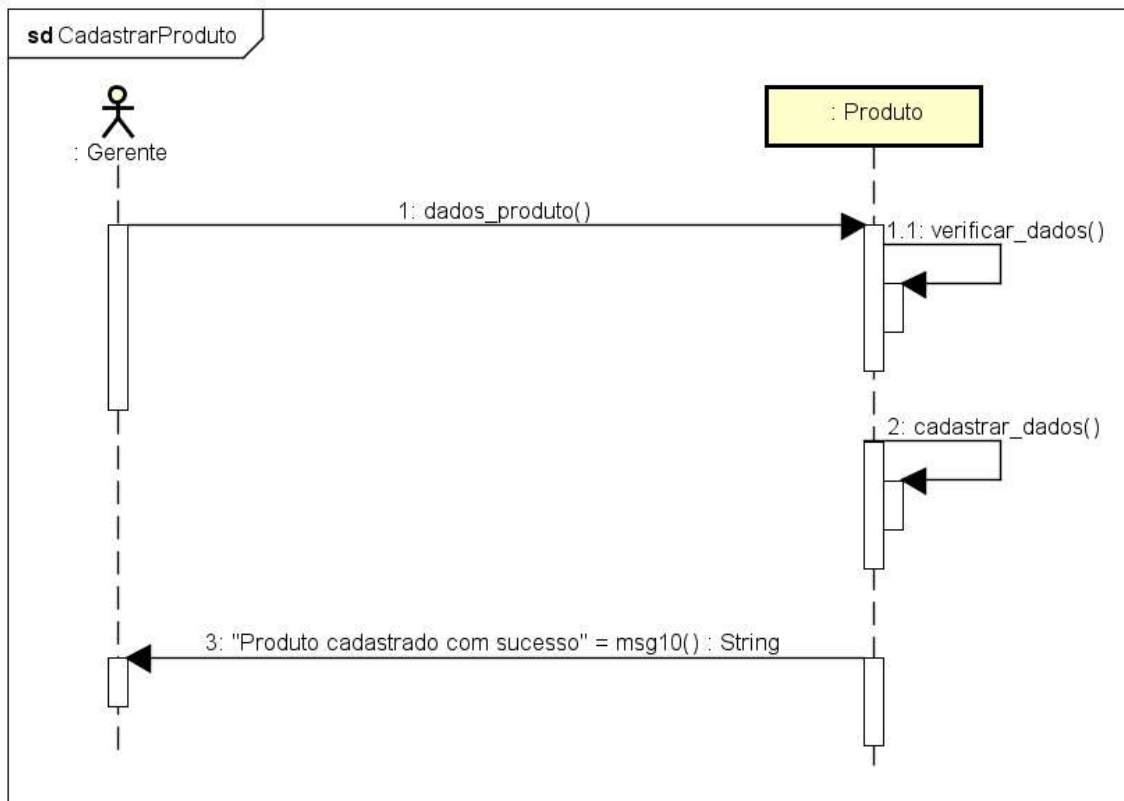




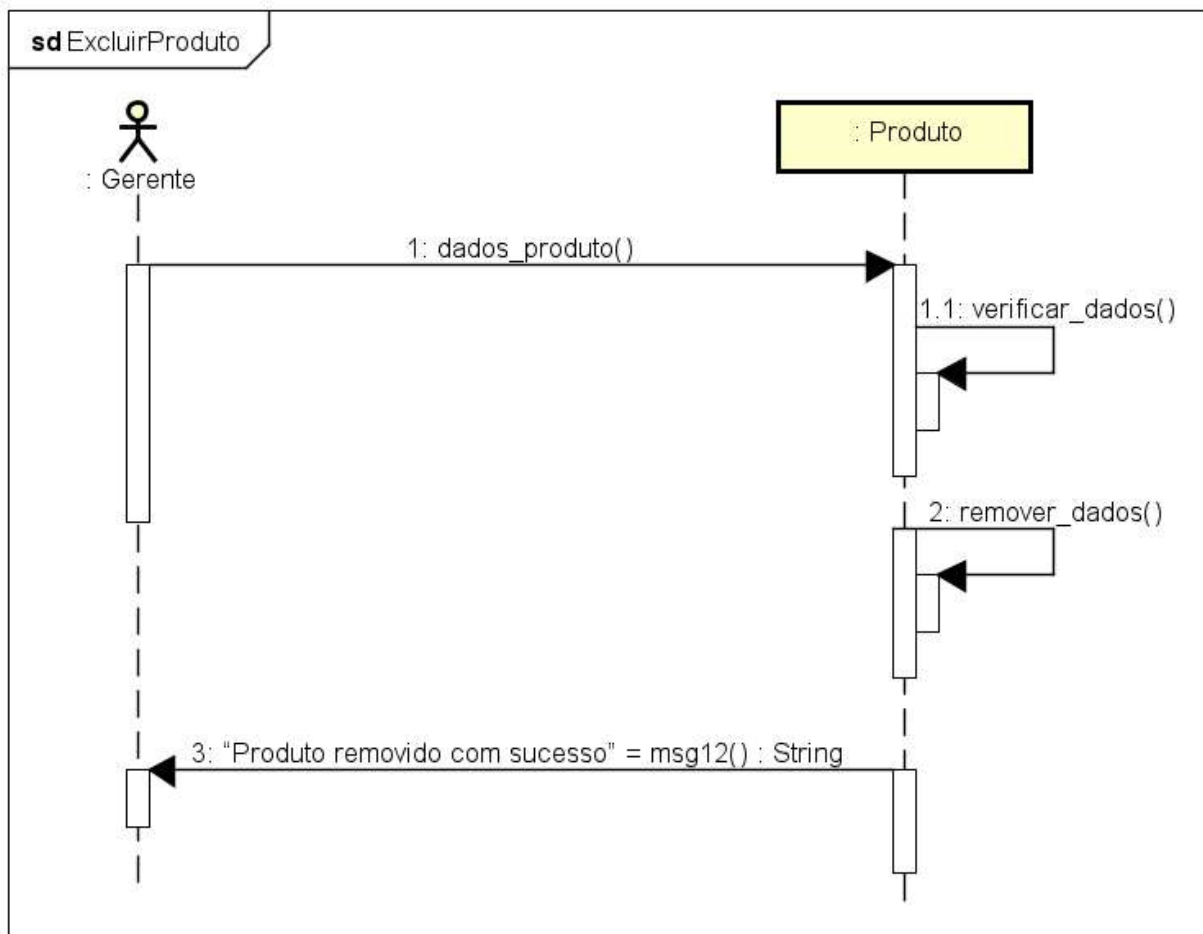












## Tecnologias Utilizadas

### Frontend:

O frontend da aplicação será desenvolvido utilizando Angular, um framework baseado em TypeScript que implementa o conceito de "Single Page Application" (SPA). Nesse modelo, apenas uma única página HTML é carregada inicialmente, e o conteúdo é atualizado dinamicamente conforme as interações do usuário. Essa abordagem elimina a necessidade de recarregar páginas inteiras, resultando em maior desempenho, melhor experiência do usuário, navegação otimizada e um desenvolvimento modular e reutilizável.

**Backend:** O backend será implementado em Java, adotando o padrão arquitetural Model-View-Controller (MVC), que organiza a aplicação em três camadas principais:

- **Model:** Responsável pela lógica de negócios e pela interação com o banco de dados.

- View: Encarregada de apresentar os dados ao usuário por meio da interface.
- Controller: Atua como intermediário, gerenciando a comunicação entre o usuário, a interface e a lógica de negócios.

Essa estrutura permite que o backend se comunique diretamente com o banco de dados para executar operações CRUD (Create, Read, Update, Delete), garantindo uma aplicação escalável e de fácil manutenção.

#### Banco de Dados:

O banco de dados escolhido será o MySQL, um sistema de gerenciamento de banco de dados relacional (RDBMS) reconhecido por sua robustez, suporte a transações, escalabilidade e compatibilidade com a tecnologia Java.

#### Infraestrutura:

A aplicação será hospedada na AWS (Amazon Web Services), selecionada por sua confiabilidade, escalabilidade e ampla gama de serviços de suporte. A AWS também oferece opções gratuitas, como o AWS Free Tier, que possibilitam realizar testes e implementar protótipos sem custo inicial, garantindo alta disponibilidade e segurança na produção.

**\*\*\*\*\*A PARTIR DAQUI ADICIONAREI IMAGENS ELABORADAS NO FIGMA PARA TER UMA PRIMEIRA IMPRESSAO DA APLICAÇÃO E PASSAR PARA O “CLIENTE” \*\*\*\*\***