

UNIVERSIDADE FEDERAL DE ALAGOAS

CIÊNCIA DA COMPUTAÇÃO

JOÃO VICTOR CAVALCANTE LIMA

JOSÉ FABIANO SILVA DE ANDRADE

PEDRO MARINHO DOS ANJOS FEITOSA

VITOR MAGNO

**Ambiente de Gerenciamento para MEI do Ramo Alimentício**

MACEIÓ/AL

2022

JOÃO VICTOR CAVALCANTE LIMA

JOSÉ FABIANO SILVA DE ANDRADE

PEDRO MARINHO DOS ANJOS FEITOSA

VITOR MAGNO

**Ambiente de Gerenciamento para MEI do Ramo Alimentício**

Projeto de P3, feito por alunos do curso

Ciência da Computação da UFAL.

Orientador: Ranilson Oscar Araújo Paiva

MACEIÓ/AL

2022

**Descrição do Projeto**

**O quê?**

Ferramenta de gerenciamento de encomendas, produtos, estoque, vendas e clientes B2B.

**Porquê?**

Para facilitar o gerenciamento de encomendas, como quais produtos, em qual data e para qual cliente deve ser entregue, ajudar em um melhor gerenciamento de itens em estoque, evitando contratempos, gerenciar a movimentação de capital envolvido, para manter um melhor tracking de custos, receita e lucro.

**Onde?**

Web

**Quando?**

08/09/2022 a 12/12/2022

**Por quem?**

Equipe de desenvolvimento:

* Pedro Marinho: Fullstack
* Vítor Magno: Fullstack
* João Lima: Front-end
* José Fabiano: Fullstack

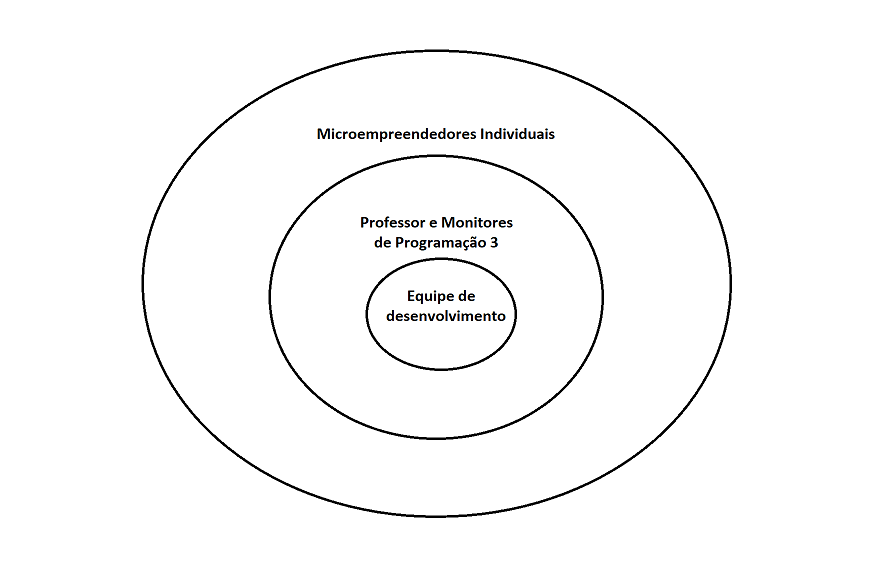
**Como?**

Utilizando Laravel no Back-end, Bootstrap ou/e Vue.js no Front-end. Banco de dados (MySQL)

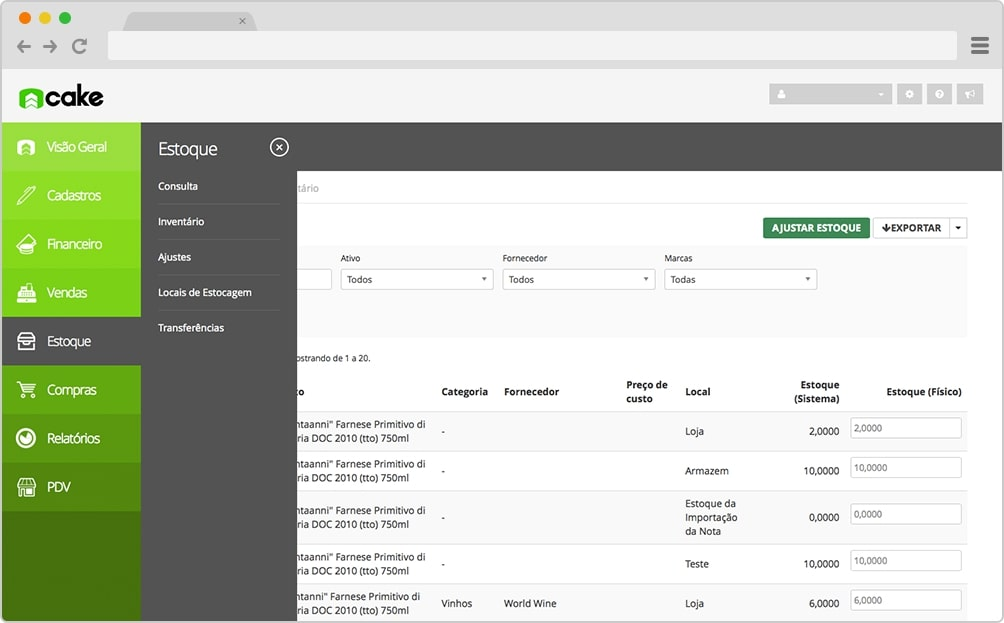
**Quanto?**

Orçamento inicial R$0,00

**Mapa de Stakeholders:**

****

**Sistema relacionado:**



**Requisitos Funcionais**

1. **Introdução**

Este documento especifica os requisitos do Sistema de Gerenciamento para MEI do Ramo Alimentício, fornecendo aos desenvolvedores as informações necessárias para o projeto e implementação, assim como para a realização dos testes e homologação do sistema.

* 1. **Visão geral do documento**

Além desta seção introdutória, as seções seguintes estão organizadas como descrito abaixo.

1. **Seção 2 – Descrição geral do sistema**: apresenta uma visão geral do sistema, caracterizando qual é o seu escopo e descrevendo seus usuários.
2. **Seção 3 – Requisitos funcionais (casos de uso)**: especifica todos os casos de uso do sistema, descrevendo os fluxos de eventos, prioridades, atores, entradas e saídas de cada caso de uso a ser implementado.
3. **Seção 4 – Requisitos não-funcionais**: especifica todos os requisitos não funcionais do sistema, divididos em requisitos de usabilidade, confiabilidade, desempenho, segurança, distribuição, adequação a padrões e requisitos de hardware e software.
4. **Seção 5 – Referências:** apresenta referências para outros documentos utilizados para a confecção deste documento.
   1. **Convenções, termos e abreviações**

A correta interpretação deste documento exige o conhecimento de algumas convenções e termos específicos, que são descritos a seguir.

* + 1. **Identificação dos requisitos**

Por convenção, a referência a requisitos é feita através do nome da subseção onde eles estão descritos, seguidos do identificador do requisito, de acordo com a especificação a seguir:

[*nome da subseção. identificador do requisito*]

Por exemplo, o requisito funcional [Recuperação de dados.RF016] deve estar descrito em uma subseção chamada “Recuperação de dados”, em um bloco identificado pelo número [RF016]. Já o requisito não-funcional [Confiabilidade.NF008] deve estar descrito na seção de requisitos não-funcionais de Confiabilidade, em um bloco identificado por [NF008].

Os requisitos devem ser identificados com um identificador único. A numeração inicia com o identificador [RF001] ou [NF001] e prossegue sendo incrementada à medida que forem surgindo novos requisitos.

* + 1. **Prioridades dos requisitos**

Para estabelecer a prioridade dos requisitos, nas seções 4 e 5, foram adotadas as denominações “essencial”, “importante” e “desejável”.

1. **Essencial** é o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.
2. **Importante** é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim.
3. **Desejável** é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.
4. **Descrição geral do sistema**
   1. **Abrangência e sistemas relacionados**

Esse sistema foi pensado para facilitar o gerenciamento e crescimento profissional de pedidos para MEIs do ramo alimentício. Para isso ele irá conter, cadastro de clientes, pedidos, materiais e produtos, além disso irá fornecer informações importantes para o negócio como; clientes mais recorrentes, pedidos mais comuns, materiais em falta ou estoque baixo e bairros mais atendidos.

1. **Requisitos funcionais**

* 1. **Cadastro de Clientes**

**[RF001] Criar componente**

**Descrição do caso de uso:** Este caso de uso permite que o usuário crie e armazene um novo componente no sistema.

| **Prioridade**: | ■ | Essencial | ◻ | Importante | ◻ | Desejável |
| --- | --- | --- | --- | --- | --- | --- |

**Entradas e pré-condições**: Nome, Cpf, Endereço e Contato

**Saídas e pós-condição**: um componente é cadastrado no sistema

**[RF002] Excluir componente**

**Descrição do caso de uso:** Este caso de uso permite que o usuário exclua um componente do cadastro de componentes do sistema. Um componente pode ser excluído de qualquer instanciação de metodologia (árvore).

| **Prioridade**: | ■ | Essencial | ◻ | Importante | ◻ | Desejável |
| --- | --- | --- | --- | --- | --- | --- |

**Entradas e pré-condições**: recebe como entrada o componente que se deseja excluir

**Saídas e pós-condição**: o usuário consegue excluir o componente que deseja

**[RF003] Alterar componente**

**Descrição do caso de uso:** Este caso de uso permite que o usuário altere os dados de um componente.

| **Prioridade**: | ■ | Essencial | ◻ | Importante | ◻ | Desejável |
| --- | --- | --- | --- | --- | --- | --- |

**Entradas e pré-condições**: recebe como entrada o componente que se deseja alterar.

**Saídas e pós-condição**: um componente é alterado no sistema.

* 1. **Cadastro de Pedidos**

**[RF001] Criar componente**

**Descrição do caso de uso:** Este caso de uso permite que o usuário crie e armazene um novo componente no sistema.

| **Prioridade**: | ■ | Essencial | ◻ | Importante | ◻ | Desejável |
| --- | --- | --- | --- | --- | --- | --- |

**Entradas e pré-condições**: Produtos, quantidade e cliente

**Saídas e pós-condição**: um componente é cadastrado no sistema

**[RF002] Excluir componente**

**Descrição do caso de uso:** Este caso de uso permite que o usuário exclua um componente do cadastro de componentes do sistema. Um componente pode ser excluído de qualquer instanciação de metodologia (árvore).

| **Prioridade**: | ■ | Essencial | ◻ | Importante | ◻ | Desejável |
| --- | --- | --- | --- | --- | --- | --- |

**Entradas e pré-condições**: recebe como entrada o componente que se deseja excluir

**Saídas e pós-condição**: o usuário consegue excluir o componente que deseja

**[RF003] Alterar componente**

**Descrição do caso de uso:** Este caso de uso permite que o usuário altere os dados de um componente.

| **Prioridade**: | ■ | Essencial | ◻ | Importante | ◻ | Desejável |
| --- | --- | --- | --- | --- | --- | --- |

**Entradas e pré-condições**: recebe como entrada o componente que se deseja alterar.

**Saídas e pós-condição**: um componente é alterado no sistema.

* 1. **Cadastro de Produtos**

**[RF001] Criar componente**

**Descrição do caso de uso:** Este caso de uso permite que o usuário crie e armazene um novo componente no sistema.

| **Prioridade**: | ■ | Essencial | ◻ | Importante | ◻ | Desejável |
| --- | --- | --- | --- | --- | --- | --- |

**Entradas e pré-condições**: Materiais, medidas e valor de venda

**Saídas e pós-condição**: um componente é cadastrado no sistema

**[RF002] Excluir componente**

**Descrição do caso de uso:** Este caso de uso permite que o usuário exclua um componente do cadastro de componentes do sistema. Um componente pode ser excluído de qualquer instanciação de metodologia (árvore).

| **Prioridade**: | ■ | Essencial | ◻ | Importante | ◻ | Desejável |
| --- | --- | --- | --- | --- | --- | --- |

**Entradas e pré-condições**: recebe como entrada o componente que se deseja excluir

**Saídas e pós-condição**: o usuário consegue excluir o componente que deseja

**[RF003] Alterar componente**

**Descrição do caso de uso:** Este caso de uso permite que o usuário altere os dados de um componente.

| **Prioridade**: | ■ | Essencial | ◻ | Importante | ◻ | Desejável |
| --- | --- | --- | --- | --- | --- | --- |

**Entradas e pré-condições**: recebe como entrada o componente que se deseja alterar.

**Saídas e pós-condição**: um componente é alterado no sistema.

* 1. **Cadastro de Materiais**

**[RF001] Criar componente**

**Descrição do caso de uso:** Este caso de uso permite que o usuário crie e armazene um novo componente no sistema.

| **Prioridade**: | ■ | Essencial | ◻ | Importante | ◻ | Desejável |
| --- | --- | --- | --- | --- | --- | --- |

**Entradas e pré-condições**: Nome do material, Quantidade, Valor.

**Saídas e pós-condição**: um componente é cadastrado no sistema

**[RF002] Excluir componente**

**Descrição do caso de uso:** Este caso de uso permite que o usuário exclua um componente do cadastro de componentes do sistema. Um componente pode ser excluído de qualquer instanciação de metodologia (árvore).

| **Prioridade**: | ■ | Essencial | ◻ | Importante | ◻ | Desejável |
| --- | --- | --- | --- | --- | --- | --- |

**Entradas e pré-condições**: recebe como entrada o componente que se deseja excluir

**Saídas e pós-condição**: o usuário consegue excluir o componente que deseja

**[RF003] Alterar componente**

**Descrição do caso de uso:** Este caso de uso permite que o usuário altere os dados de um componente.

| **Prioridade**: | ■ | Essencial | ◻ | Importante | ◻ | Desejável |
| --- | --- | --- | --- | --- | --- | --- |

**Entradas e pré-condições**: recebe como entrada o componente que se deseja alterar.

**Saídas e pós-condição**: um componente é alterado no sistema.

* 1. **Estatísticas de Vendas**

**[RF001] Visualizar componente**

**Descrição do caso de uso:** Este caso de uso permite que o usuário visualize itens mais vendidos, clientes que mais compram, materiais acabando no estoque e quantidade de clientes por bairros.

| **Prioridade**: | ◻ | Essencial | ◻ | Importante | ■ | Desejável |
| --- | --- | --- | --- | --- | --- | --- |

**Entradas e pré-condições**: não tem.

**Saídas e pós-condição**: um componente é visualizado no sistema.

1. **Requisitos não-funcionais**

**[NF001]Usabilidade**

A interface será bem simples e intuitiva com a intenção de não ser uma barreira para usuários inexperientes.

| **Prioridade**: | ■ | Essencial | ◻ | Importante | ◻ | Desejável |
| --- | --- | --- | --- | --- | --- | --- |

**[NF002] Desempenho**

Embora não seja um requisito essencial ao sistema, deve ser considerada por corresponder a um fator de qualidade de software.

| **Prioridade**: | ◻ | Essencial | ■ | Importante | ◻ | Desejável |
| --- | --- | --- | --- | --- | --- | --- |

**Descrição de Models e Controllers**

* **Model**

Para cada ***table*** no ***database***, iremos criar um ***model***, para facilitar a manipulação de dados no mesmo.

Em cada **model**, iremos listar as colunas do ***database*** que iremos usar dentro do ***protected $fillable***. Então não é necessário dizer isso em cada ***model*** deste documento.

* 1. **User**

Como só iremos ter um usuário (pelo menos nesse primeiro momento) no nosso sistema (que será o admin), não é necessário colocar um relacionamento de tabelas. Então nosso ***model User*** será apenas o ***$fillable*** e o ***$hidden*** (para o password).

* 1. **Cliente**

Além do ***$fillable***, será usado um relacionamento ***hasOne*** para ***Endereço***, das colunas ***id\_endereço*** em ***Cliente*** para ***id*** de ***Endereço***, e ***hasMany*** para ***Encomenda***, das colunas ***id\_cliente*** em ***Encomenda*** e ***id*** em ***Cliente***. Pois cada cliente tem um endereço e um cliente pode ter várias encomendas.

* 1. **Encomenda**

Além do ***$fillable***, será usado um relacionamento ***hasOne*** para ***Cliente***, das colunas ***id*** em ***Cliente*** para ***id\_cliente*** de ***Encomenda***, e ***hasMany*** para ***EncomendaProduto***, das colunas ***id\_encomenda*** em ***EncomendaProduto*** e ***id*** em ***Encomenda***. Pois cada encomenda pertence a apenas um cliente e uma encomenda contém um ou mais produtos.

* 1. **Endereço**

Além do ***$fillable***, será usado um relacionamento ***hasOne*** para ***Cliente***, das colunas ***id\_endereço*** em ***Cliente*** para ***id*** de ***Endereço***. Pois um endereço pertence á um cliente.

* 1. **EncomendaProduto**

Esse ***model*** é o meio de campo entre Encomenda e Produto, para usar um relacionamento de many-to-many. Então, além do ***$fillable***, será usado um relacionamento ***hasOne*** para ***Encomenda***, das colunas ***id*** em ***Encomenda*** para ***id\_encomenda*** de ***EncomendaProduto***, e ***hasOne*** para ***Produto***, das colunas ***id\_produto*** em ***EncomendaProduto*** e ***id*** em ***Produto***. Com isso, teremos um many-to-many entre ***Encomenda*** e ***Produto***

* 1. **Produto**

Além do ***$fillable***, será usado um relacionamento ***hasMany*** para ***Venda***, das colunas ***id*** em ***Produto*** para ***id\_produto*** de ***Venda***, um ***hasMany*** para ***EncomendaProduto***, das colunas ***id\_produto*** em ***EncomendaProduto*** e ***id*** em ***Produto*** e um outro ***hasMany*** para ***MaterialProduto***, das colunas ***id\_produto*** em ***MaterialProduto*** e ***id*** em ***Produto***. Pois cada produto contém um ou mais materiais, os produtos também estão associados à uma ou mais vendas e também à uma ou mais encomendas.

* 1. **MaterialProduto**

Esse ***model*** é o meio de campo entre Material e Produto, para usar um relacionamento de many-to-many.Então, além do ***$fillable***, será usado um relacionamento ***hasOne*** para ***Produto***, das colunas ***id*** em ***Produto*** para ***id\_produto*** de ***MaterialProduto***, e ***hasOne*** para ***Material***, das colunas ***id\_material*** em ***MaterialProduto*** e ***id*** em ***Material***. Com isso, teremos um many-to-many entre ***Material*** e ***Produto***

* 1. **Material**

Além do ***$fillable***, será usado um relacionamento ***hasMany*** para ***MaterialProduto***, das colunas ***id*** em ***Material*** para ***id\_material*** de ***MaterialProduto***. Pois um produto é feito por um ou mais material.

* 1. **Venda**

Além do ***$fillable***, será usado um relacionamento ***hasOne*** para ***Produto***, das colunas ***id*** em ***Produto*** para ***id\_produto*** de ***Produto***. Pois uma venda está associado à um produto.

* **Controllers**
  1. **ClienteController**

Neste controller, serão feitos métodos básicos que formam um CRUD:

* **Create** - Necessário para a **criação** das informações (ex.: nome, telefone, email e etc) relacionadas a cada cliente criado no sistema e salvo banco de dados.
* **Show** - Necessário para **resgatar** do banco de dados as informações relacionadas a cada cliente cadastrado no sistema, e mostrar na tela para o usuário.
* **Update** - Necessário para **editar** as informações (se necessário) do banco de dados de um certo cliente cadastrado no sistema.
* **Delete**  - Necessário para **excluir** as informações (se necessário) de cada cliente cadastrado no sistema.
  1. **EncomendaController**

Neste controller, serão feitos métodos básicos que formam um CRUD:

* **Create** - Necessário para a **criação** das informações (ex.: nome do cliente, data do pedido, data de entrega, estado da encomenda e etc) relacionadas a cada encomenda criada no sistema e salva no banco de dados.
* **Show** - Necessário para **resgatar** do banco de dados as informações relacionadas a cada encomenda cadastrada no sistema, e mostrar na tela para o usuário.
* **Update** - Necessário para **editar** as informações (se necessário) do banco de dados de uma certa encomenda cadastrada no sistema.
* **Delete**  - Necessário para **excluir** as informações (se necessário) de encomendas cadastradas no sistema.
  1. **EstoqueController**

Neste controller, serão feitos métodos básicos que formam um CRUD:

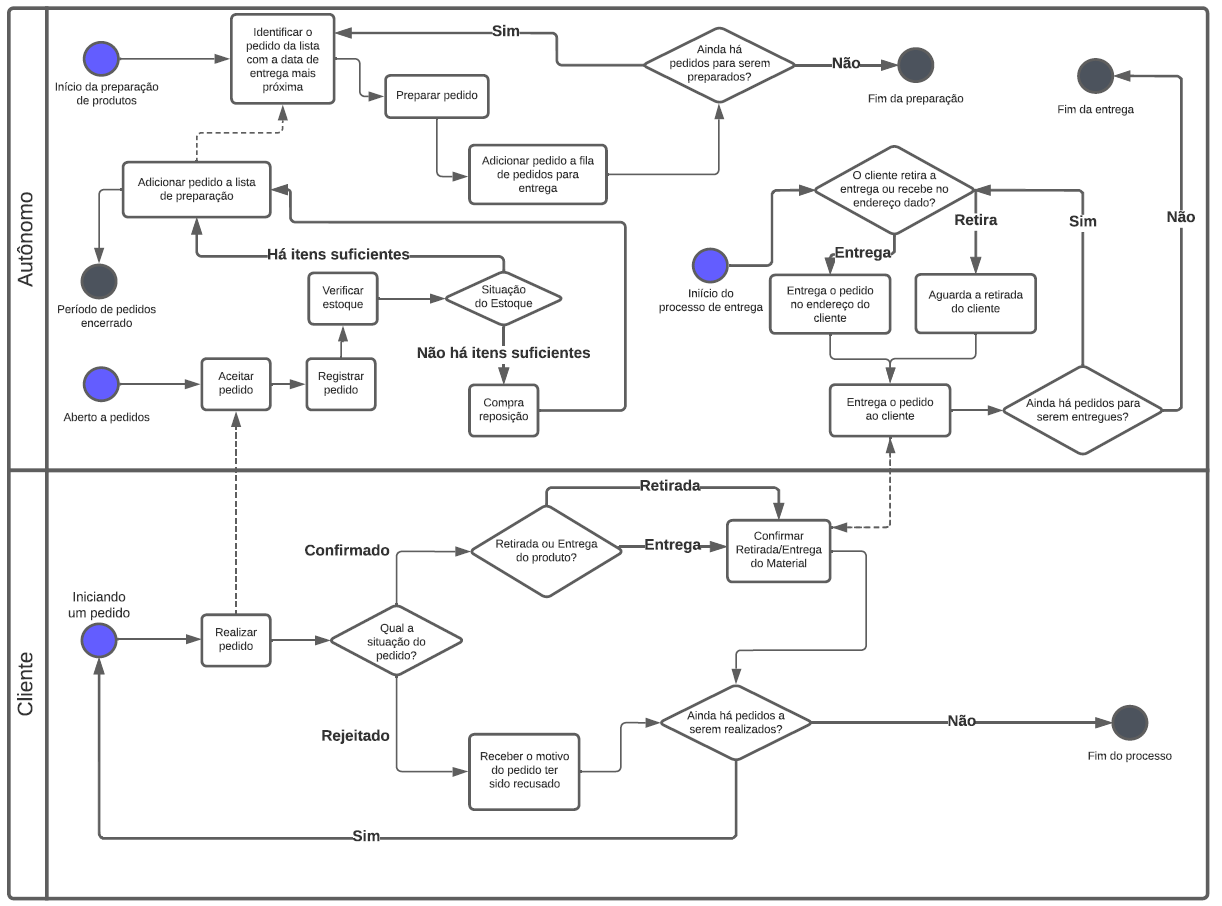
* **Create** - Necessário para a **criação** das informações (ex.: nome do item, quantidade, tipo e etc) relacionadas a cada item (guardado em estoque) criado no sistema e salvo no banco de dados.
* **Show** - Necessário para **resgatar** do banco de dados as informações relacionadas a cada item (guardado em estoque) cadastrado no sistema, e mostrar na tela para o usuário.
* **Update** - Necessário para **editar** as informações do banco de dados de um certo item (guardado em estoque) cadastrado no sistema.
* **Delete**  - Necessário para **excluir** as informações (se necessário) de um item qualquer (guardado em estoque) cadastrado no sistema.
  1. **ProdutoController**

Neste controller, serão feitos métodos básicos que formam um CRUD:

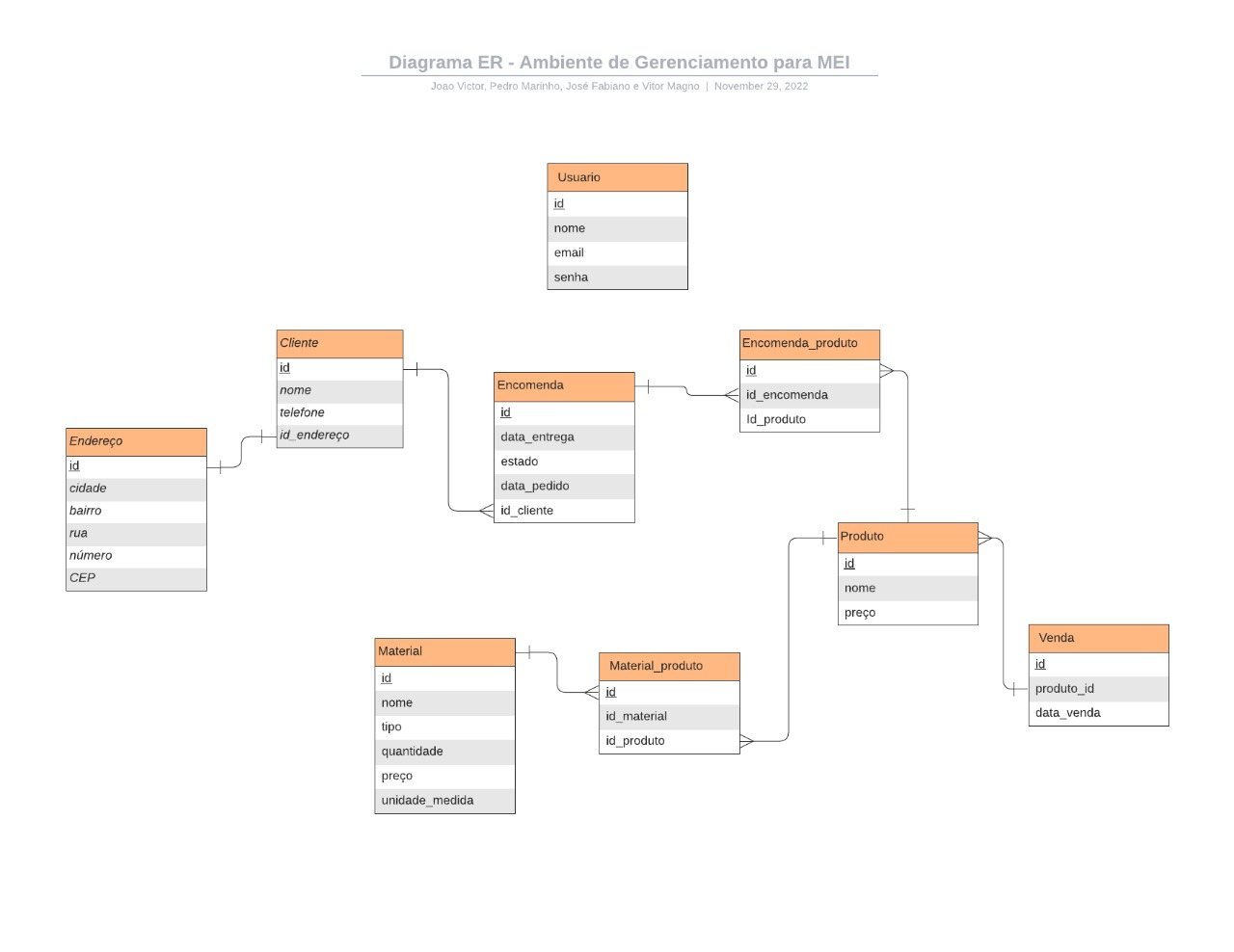
* **Create** - Necessário para a **criação** das informações (ex.: nome do produto, valor e etc) relacionadas a cada produto cadastrado no sistema e salvo no banco de dados.
* **Show** - Necessário para **resgatar** do banco de dados as informações relacionadas a cada produto cadastrado no sistema, e mostrar na tela para o usuário.
* **Update** - Necessário para **editar** as informações do banco de dados de um certo produto cadastrado no sistema.
* **Delete**  - Necessário para **excluir** as informações (se necessário) de um produto qualquer cadastrado no sistema.
  1. **VendaController**

Neste controller, ainda estamos elaborando uma estratégia para conseguirmos demonstrar de alguma forma estatísticas de vendas, lucros e outras informações adquiridas durante um certo período de tempo através de informações encontradas no banco de dados.

**Diagrama BPMN**

****

**Diagrama ER**

****

**Link:**[**https://lucid.app/lucidchart/5baf047a-e022-43db-8f52-8ef0f4234470/edit?invitationId=inv\_0aa72dc4-e1f0-4140-b32b-74c19801d79b#**](https://lucid.app/lucidchart/5baf047a-e022-43db-8f52-8ef0f4234470/edit?invitationId=inv_0aa72dc4-e1f0-4140-b32b-74c19801d79b#)