

**Geisiele de Oliveira - BP3053563**

**Thiago Oliveira - BP 3053636**

**Vinicius Arantes - BP3053709**

**Vinícius Magalhães - BP3054365**

**Loja – VENEZART**

**Bragança Paulista**

**2025**

## Descrição Completa do Projeto Venezart

O **Venezart** é uma aplicação web desenvolvida com **Flask** no backend e **React** no frontend, projetada para funcionar como uma loja especializada em **utensílios artísticos**, com foco em materiais voltados para **pintura em tela**.

O objetivo do sistema é oferecer uma experiência clara, rápida e intuitiva para os clientes, ao mesmo tempo em que fornece ao administrador ferramentas eficientes para controle do catálogo e dos pedidos realizados.

Os usuários do Venezart podem navegar pelos produtos, visualizar informações detalhadas, adicionar itens ao carrinho, ajustar quantidades, remover produtos e finalizar a compra. Após a conclusão do pedido, o sistema gera automaticamente um **comprovante em PDF**, permitindo ao usuário baixar e armazenar seu registro da compra.

Cada usuário também possui acesso ao próprio histórico de compras por meio de uma área reservada.

O sistema conta ainda com um **perfil administrador**, capaz de cadastrar novos produtos, editar informações existentes, excluir itens do catálogo e visualizar **todos os pedidos feitos na plataforma**, garantindo um gerenciamento eficiente e centralizado da loja.

## Modelos do Sistema (ORM – SQLAlchemy)

Abaixo está uma explicação detalhada das classes que compõem a estrutura de dados da Venezart.

### 1. Classe User

A classe **User** representa os usuários cadastrados na plataforma — clientes e administradores.

Por herdar **UserMixin**, ela integra automaticamente com o **Flask-Login**, possibilitando autenticação, gerenciamento de sessão e verificação de permissões.

#### Atributos principais:

- **id**: identificador único.
- **username**: nome de usuário utilizado no login (único).
- **password**: senha do usuário.
- **name**: nome completo.

- **email:** endereço de e-mail utilizado para cadastro e recuperação (único).
- **cart\_items:** relação com os itens adicionados ao carrinho.

### Métodos e propriedades:

- `is_authenticated`, `is_active`, `is_anonymous`: propriedades necessárias ao Flask-Login.
- `get_id()`: devolve o ID em formato string.
- `__repr__`: retorna uma representação textual útil para debug.

Essa classe é a base do sistema de autenticação e relacionamentos com carrinho e pedidos.

## 2. Classe Product

A classe **Product** representa os produtos disponíveis no catálogo da Venezart.

### Atributos:

- **id:** identificador do produto.
- **name:** nome do material artístico.
- **description:** detalhes do produto (descrição textual).
- **price:** preço unitário.
- **cart\_items:** itens de carrinho vinculados ao produto.

Essa entidade representa o catálogo administrado pelo administrador e acessado pelos clientes durante suas compras.

## 3. Classe CartItem

A classe **CartItem** representa os itens que o usuário adiciona ao carrinho, servindo como ligação entre as classes **User** e **Product**.

### Atributos:

- **id:** identificação do item no carrinho.
- **user\_id:** referência ao usuário.
- **product\_id:** referência ao produto selecionado.
- **quantity:** quantidade daquele produto no carrinho.

Essa tabela possibilita manipular o carrinho de compras, permitindo adicionar produtos, alterar quantidades ou remover itens.

#### 4. Classe Pedido

A classe **Pedido** representa uma compra confirmada.

##### Atributos:

- **id**: identificador do pedido.
- **user\_id**: referência ao usuário comprador.
- **data**: data e hora da realização.
- **status**: estado atual do pedido (ex.: *pendente*, *finalizado*).
- **itens**: lista de itens presentes no pedido.

Cada usuário visualiza apenas seus próprios pedidos; o administrador visualiza todos.

#### 5. Classe PedidoItem

A classe **PedidoItem** detalha cada produto que compõe um pedido finalizado.

##### Atributos:

- **id**: identificação do item.
- **pedido\_id**: referência ao pedido ao qual pertence.
- **product\_name**: nome do produto comprado (armazenado para histórico).
- **quantity**: quantidade adquirida.
- **price**: valor do produto no momento da compra.

Esses dados são essenciais para compor o comprovante em PDF e o histórico de compra.

## Diagrama de Classes por Domínio

Para compreender melhor a estrutura lógica e o funcionamento interno do sistema **Venezart**, foi elaborado um **Diagrama de Classes por Domínio**. Esse diagrama representa os principais componentes da aplicação e como eles se relacionam dentro do contexto da loja virtual.

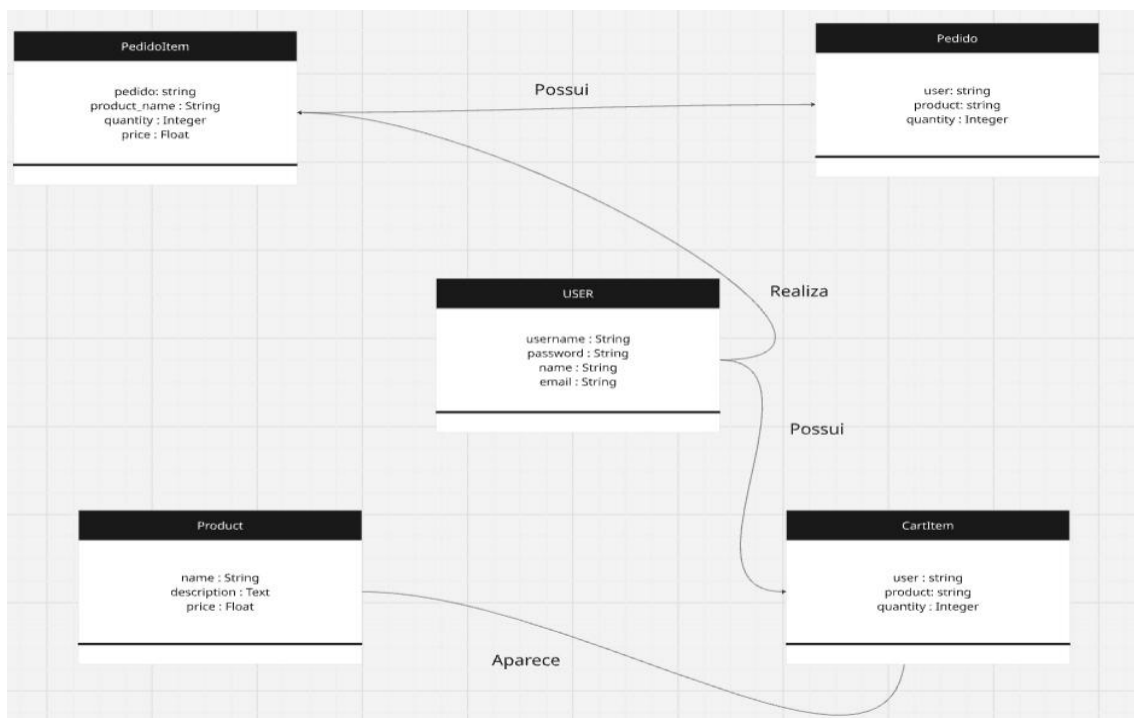
O domínio da aplicação envolve entidades fundamentais como **Usuário**, **Produto**, **Carrinho**, **Pedido** e **Itens de Pedido**. Cada classe reflete diretamente responsabilidades específicas dentro do sistema e está associada às ações que podem ser realizadas durante o fluxo da compra.

Através do diagrama, é possível visualizar de forma organizada:

- A estrutura interna de cada classe
- Seus atributos principais
- Seus métodos (quando aplicável)
- Os relacionamentos entre as classes
- Como cada entidade contribui para o funcionamento geral da plataforma

Esse diagrama auxilia a demonstrar a lógica de negócio do Venezart, representando o comportamento do sistema de forma abstrata, independente de implementação técnica.

A seguir, apresenta-se o **Diagrama de Classes por Domínio**:



## Modelo MER – Modelo Entidade Relacionamento

Após a análise das classes e do domínio da aplicação, foi desenvolvido também o **Modelo Entidade-Relacionamento (MER)**, responsável por representar de forma clara a estrutura do banco de dados utilizado pelo Venezart.

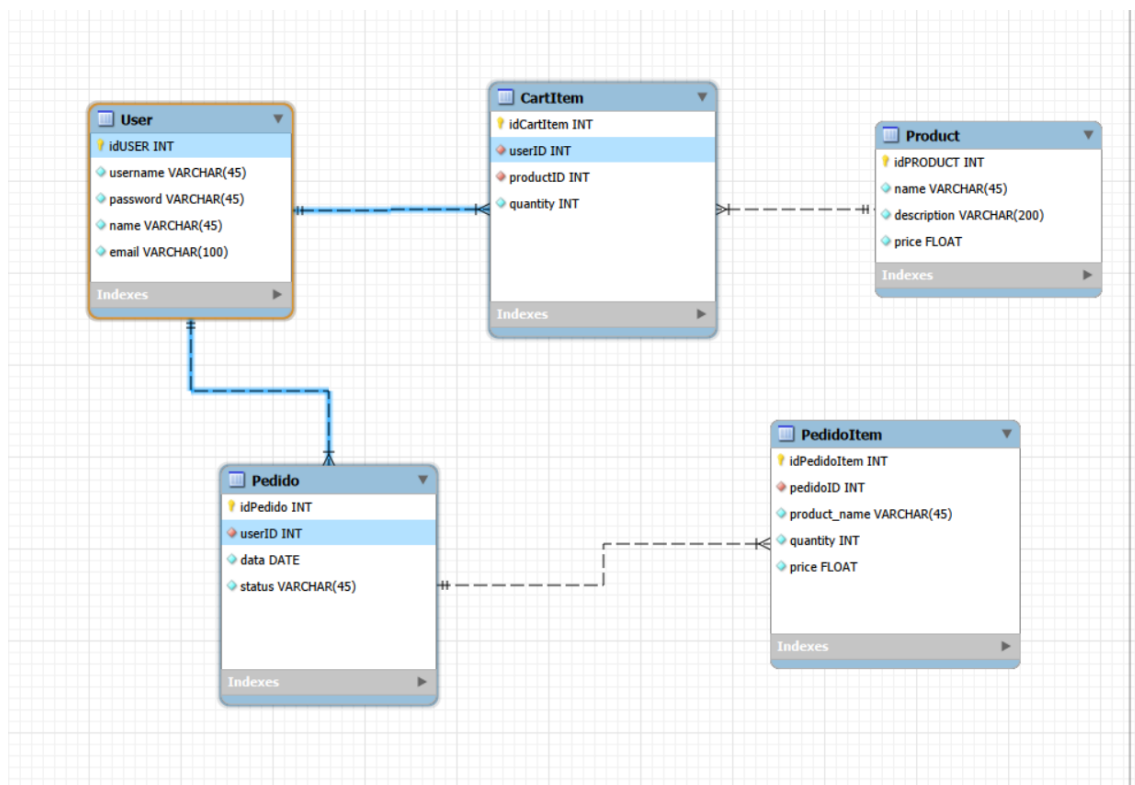
O MER traduz os conceitos do domínio em entidades de banco de dados, permitindo visualizar:

- Cada tabela existente no sistema
- Seus atributos e tipos de dados
- Suas chaves primárias e estrangeiras
- Os relacionamentos entre entidades, como 1:N ou N:N
- A correspondência direta entre os modelos ORM e as tabelas físicas

O MER é essencial para garantir organização, integridade e coerência dos dados, servindo como base para a criação e manutenção do banco de dados utilizado pela aplicação.

No caso do Venezart, o modelo contempla entidades como **Usuário**, **Produto**, **Item de Carrinho**, **Pedido** e **Item de Pedido**, refletindo exatamente como os dados são armazenados e consultados durante o funcionamento da loja virtual.

A seguir, apresenta-se o **Modelo Entidade-Relacionamento (MER)**:



## Conclusão

O **Venezart** é um sistema completo de comércio eletrônico voltado para materiais de arte, unindo **Flask** e **React** em uma aplicação robusta que permite compras, geração de comprovantes, gerenciamento do carrinho, consulta de histórico e administração total do catálogo e dos pedidos.

Sua estrutura de modelos ORM é bem definida e garante organização e eficiência no fluxo de dados entre usuários, produtos, carrinho e pedidos.