

Geisiele de Oliveira - BP3053563

Thiago Oliveira - BP 3053636

Vinicius Arantes - BP3053709

Vinícius Magalhães - BP3054365

Victoria Gabriela - BP3053466

Loja – VENEZART

Bragança Paulista

2025

Geisiele de Oliveira - BP3053563

Thiago Oliveira - BP 3053636

Vinicius Arantes - BP3053709

Vinícius Magalhães - BP3054365

Victoria Gabriela - BP3053466

Loja - VENEZART

Trabalho desenvolvido para a disciplina de Análise e Orientação a Objetos,
sob orientação da Prof.^a Ana Giancoli,
do curso de Análise e Desenvolvimento de Sistemas

do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – Campus Bragança Paulista.

Bragança Paulista

2025

Sumário

1. Introdução	1
2. Desenvolvimento	2
2.1 Tecnologias Utilizadas	2
2.1.1 Front-end	2
2.1.2 Back-end e Bibliotecas Flask	3
2.2 Listagem de Requisitos	4
2.2.1 Requisitos Funcionais	4
2.2.2 Requisitos Não Funcionais	5
2.3 Diagrama de Casos de Uso	6
2.4 Diagrama de Atividades – Finalização de Compra	7
2.5 Diagrama de Atividades Geral	8
3. Conclusão	9
4. Referências	10

1. Introdução

A presente proposta descreve o desenvolvimento de uma **loja de utensílios artísticos com foco em pintura em tela**, destinada a atender tanto iniciantes quanto profissionais da área. O projeto tem como objetivo oferecer uma plataforma online que facilite o acesso a produtos de qualidade, como pincéis, tintas, solventes, solúveis e telas, permitindo que os usuários encontrem e adquiram os materiais necessários de forma prática e organizada.

O sistema foi concebido para contemplar dois perfis de acesso: o **usuário**, que representa o cliente final da loja, e o **administrador**, responsável pelo gerenciamento do catálogo de produtos e pelo controle dos pedidos realizados. O usuário pode navegar pelos produtos, adicionar itens ao carrinho, finalizar compras e acompanhar seus pedidos, recebendo confirmações por meio do sistema e por e-mail. O administrador, por sua vez, possui acesso a funcionalidades de gerenciamento, incluindo a adição, remoção e atualização de produtos, bem como a confirmação ou cancelamento de pedidos realizados pelos clientes.

Dessa forma, a loja visa proporcionar uma experiência de compra eficiente e segura, integrando funcionalidades que atendem às necessidades dos usuários e facilitam a gestão administrativa, ao mesmo tempo em que promove a disseminação de produtos artísticos para diferentes perfis de consumidores.

2. Desenvolvimento

O sistema desenvolvido consiste em uma **loja de utensílios artísticos com foco em pintura em tela**, voltada para atender tanto iniciantes quanto profissionais da área. Seu objetivo principal é apresentar os produtos da loja de forma clara e organizada, permitindo que os usuários naveguem por páginas institucionais, conheçam os produtos disponíveis e realizem compras de maneira prática.

Para os **usuários**, a plataforma oferece funcionalidades essenciais, como a visualização de produtos, adição e remoção de itens no carrinho de compras, finalização da compra e acompanhamento dos pedidos realizados. Ao concluir uma compra, o sistema gera automaticamente um **comprovante em PDF** contendo todas as informações do pedido e envia uma confirmação por e-mail ao usuário. Além disso, cada usuário possui uma página própria com o histórico de seus pedidos, permitindo acompanhar o status de cada compra individualmente.

O **administrador** possui acesso às mesmas funcionalidades disponíveis para o usuário, como pesquisa de produtos por categoria ou por nome, mas também conta com recursos exclusivos para o gerenciamento da loja. Entre essas funcionalidades estão a adição e remoção de produtos do sistema, visualização de todos os pedidos realizados por todos os usuários e a possibilidade de confirmar ou cancelar qualquer pedido. Independentemente da ação tomada pelo administrador, o sistema garante que o usuário responsável pelo pedido seja informado por e-mail sobre o status atualizado da compra. Todas as operações administrativas são autenticadas, garantindo que apenas o administrador tenha acesso a essas funcionalidades.

Dessa forma, o sistema proporciona uma experiência de uso eficiente e segura, integrando funcionalidades voltadas tanto para o cliente quanto para o gestor da loja. A plataforma permite a gestão completa de produtos e pedidos, assegurando que o processo de compra seja transparente e organizado, enquanto oferece ferramentas administrativas robustas para o gerenciamento da loja.

2.1 Tecnologias Utilizadas

O projeto foi desenvolvido utilizando uma combinação de tecnologias modernas para front-end, back-end e banco de dados, garantindo desempenho, segurança e organização do sistema.

Front-end

- **React:** biblioteca JavaScript utilizada para construção da interface do usuário de forma dinâmica e reativa. Permite criar componentes reutilizáveis e gerenciar o estado da aplicação de maneira eficiente.
- **Vite:** ferramenta de build e bundler moderna, utilizada para acelerar o desenvolvimento e otimizar o carregamento da aplicação React.

Back-end

O back-end foi implementado em **Python**, utilizando o **Flask**, um framework leve e flexível para construção de aplicações web. Diversas bibliotecas foram incorporadas para facilitar a autenticação, gerenciamento de banco de dados e envio de informações:

- **Flask:** framework principal que gerencia as rotas, requisições e respostas da aplicação web.
- **Flask-Login:** gerencia a autenticação de usuários, permitindo controle de login, logout e manutenção de sessões de forma segura.
- **Flask-JWT-Extended:** permite autenticação baseada em tokens JWT, garantindo segurança nas rotas que requerem autorização.
- **Flask-CORS:** habilita o compartilhamento de recursos entre diferentes domínios, necessário para comunicação entre front-end e back-end.
- **Flask-Migrate e Alembic:** permitem gerenciar alterações no esquema do banco de dados de forma automatizada, facilitando atualizações e versionamento.
- **Flask-SQLAlchemy:** integra o Flask com o **SQLAlchemy**, fornecendo ORM (Object-Relational Mapping) para manipulação do banco de dados de forma orientada a objetos.
- **Flask-WTF e WTForms:** auxiliam na criação e validação de formulários web, garantindo que os dados enviados pelos usuários sejam tratados corretamente.
- **Flask-Script:** facilita a execução de comandos de gerenciamento da aplicação, como migrações e execução do servidor.

Banco de Dados

- **SQLite 3:** banco de dados relacional utilizado para armazenamento de informações do sistema, incluindo produtos, usuários e pedidos. É leve, fácil de configurar e adequado para aplicações de pequeno a médio porte.

Outras Bibliotecas e Utilitários

- **SQLAlchemy:** ORM que possibilita a manipulação do banco de dados SQLite de forma orientada a objetos.

- **email_validator**: valida endereços de e-mail fornecidos pelos usuários durante o cadastro e a finalização da compra.
- **PyJWT**: manipula tokens JWT utilizados na autenticação de usuários.
- **ReportLab e Pillow**: utilizadas para gerar os comprovantes de compra em PDF e manipular imagens, garantindo que os documentos e relatórios sejam gerados de forma automatizada.
- **Jinja2, Mako e MarkupSafe**: motores de template que permitem gerar páginas HTML dinâmicas e seguras.
- **click, colorama e blinker**: bibliotecas auxiliares para facilitar a execução de scripts, exibição de mensagens no console e comunicação interna entre componentes da aplicação.
- **Werkzeug e itsdangerous**: oferecem suporte para segurança, roteamento e geração de tokens seguros dentro do Flask.
- **pytz**: permite manipulação de fusos horários nas datas e horários das transações.

Essas tecnologias combinadas garantem que o sistema seja **funcional, seguro, escalável e de fácil manutenção**, atendendo tanto às necessidades dos usuários quanto às funcionalidades administrativas da loja.

2.2 Listagem de Requisitos

Para garantir o correto funcionamento do sistema e atender às necessidades tanto dos usuários quanto do administrador, foram definidos **requisitos funcionais e não funcionais**.

Os **requisitos funcionais** representam as funcionalidades que o sistema deve executar, ou seja, descrevem **o que a aplicação deve fazer**. Eles especificam as ações realizadas pelo usuário, como navegar pelos produtos, adicionar ou remover itens do carrinho, finalizar compras e receber confirmações, bem como as operações do administrador, incluindo gerenciamento do catálogo de produtos e controle dos pedidos.

Por outro lado, os **requisitos não funcionais** definem **as características de qualidade do sistema**, como desempenho, segurança, confiabilidade e usabilidade, garantindo que a aplicação não apenas execute suas funções, mas o faça de maneira eficiente e segura.

A seguir, serão apresentados os **requisitos funcionais do sistema**, detalhando as principais funcionalidades implementadas, permitindo uma compreensão clara do escopo do projeto e das interações entre usuários e administrador.

2.2.1 Requisitos Funcionais (RF)

Código	Descrição	Atores Envolvidos
RF01	O sistema deve permitir que o usuário visualize a página institucional da loja.	Usuário
RF02	O sistema deve permitir que o usuário visualize os produtos disponíveis.	Usuário
RF03	O sistema deve permitir que o usuário adicione produtos ao carrinho de compras.	Usuário
RF04	O sistema deve permitir que o usuário remova produtos do carrinho.	Usuário
RF05	O sistema deve permitir que o usuário finalize a compra.	Usuário
RF06	O sistema deve enviar a confirmação de compra por e-mail ao usuário.	Usuário
RF07	O sistema deve gerar um PDF do pedido com informações de produtos, quantidades e valores.	Usuário
RF08	O sistema deve permitir que o usuário visualize seus pedidos realizados.	Usuário
RF09	O sistema deve permitir que o usuário pesquise produtos por categoria.	Usuário
RF10	O sistema deve permitir que o administrador visualize todos os pedidos realizados pelos usuários.	Administrador
RF11	O sistema deve permitir que o administrador confirme ou cancele pedidos.	Administrador
RF12	O sistema deve permitir que o administrador adicione novos produtos ao sistema.	Administrador
RF13	O sistema deve permitir que o administrador remova produtos do sistema.	Administrador
RF14	O sistema deve permitir que o administrador pesquise produtos por nome e categoria.	Administrador

2.2.2 Requisitos Não Funcionais (RNF)

Código	Descrição
RNF01	O sistema deve possuir uma interface amigável e responsiva.
RNF02	O sistema deve garantir a segurança das informações dos usuários.
RNF03	O sistema deve enviar os e-mails de confirmação automaticamente após a finalização do pedido.
RNF04	O sistema deve ser acessível por meio de navegadores modernos (Chrome, Edge, Firefox).
RNF05	O tempo de resposta das páginas deve ser inferior a 3 segundos.
RNF06	O sistema deve armazenar os dados de forma persistente em um banco de dados.

2.3 Diagrama de Caso de Uso (descrição textual).

O diagrama de casos de uso é uma representação visual das funcionalidades oferecidas pelo sistema e das interações entre os atores e o sistema. Ele permite compreender de forma clara quem pode realizar cada ação dentro da aplicação e quais funcionalidades estão disponíveis para cada tipo de usuário.

No projeto da loja de utensílios artísticos, foram identificados dois atores principais: o usuário e o administrador. O usuário representa o cliente da loja, que tem acesso às funcionalidades básicas, como visualizar produtos, adicionar itens ao carrinho, remover produtos do carrinho, finalizar compras, visualizar seus pedidos e pesquisar produtos por categoria. Já o administrador possui todas essas funcionalidades, mas também tem acesso a operações de gerenciamento, incluindo a adição e remoção de produtos, visualização de todos os pedidos de todos os usuários e pesquisa avançada de produtos por nome e categoria.

Dessa forma, o diagrama de casos de uso evidencia a separação de responsabilidades e privilégios entre os diferentes atores, garantindo que cada funcionalidade esteja claramente associada ao ator correto, permitindo maior organização, segurança e entendimento do sistema como um todo.



Figura 1: Diagrama de Casos de Uso do sistema de loja de utensílios artísticos, mostrando as funcionalidades disponíveis para os atores Usuário e Administrador.

2.4 Diagrama de Atividades

O diagrama de atividades é utilizado para representar o **fluxo de ações** de um processo específico dentro do sistema, mostrando as atividades realizadas pelos atores e pelo sistema, bem como as decisões que podem ocorrer durante o fluxo.

No caso do processo de **finalização de compra**, o diagrama ilustra detalhadamente todas as etapas desde a solicitação do usuário até a confirmação da compra. Inicialmente, o **usuário solicita finalizar a compra**, e o sistema realiza a **validação das informações**, verificando se o usuário está logado, se possui e-mail cadastrado e se o carrinho contém itens.

Caso os dados sejam válidos, o sistema exibe a **tela de confirmação**, permitindo que o usuário confirme a compra. Após a confirmação, o sistema registra o pedido no banco de dados, gera um **comprovante em PDF**, salva o documento localmente no dispositivo do usuário e envia um **e-mail de confirmação** com o PDF em anexo.

Se, no entanto, os dados não forem válidos ou algum problema for identificado durante a validação, o sistema exibe uma **mensagem de erro** e retorna o usuário ao carrinho, permitindo que as correções necessárias sejam realizadas.

Dessa forma, o diagrama de atividades evidencia o **fluxo detalhado de interação entre usuário e sistema**, demonstrando os passos sequenciais, as decisões críticas e as ações automáticas do sistema, garantindo **transparência e clareza** sobre o funcionamento do processo de finalização de compra.

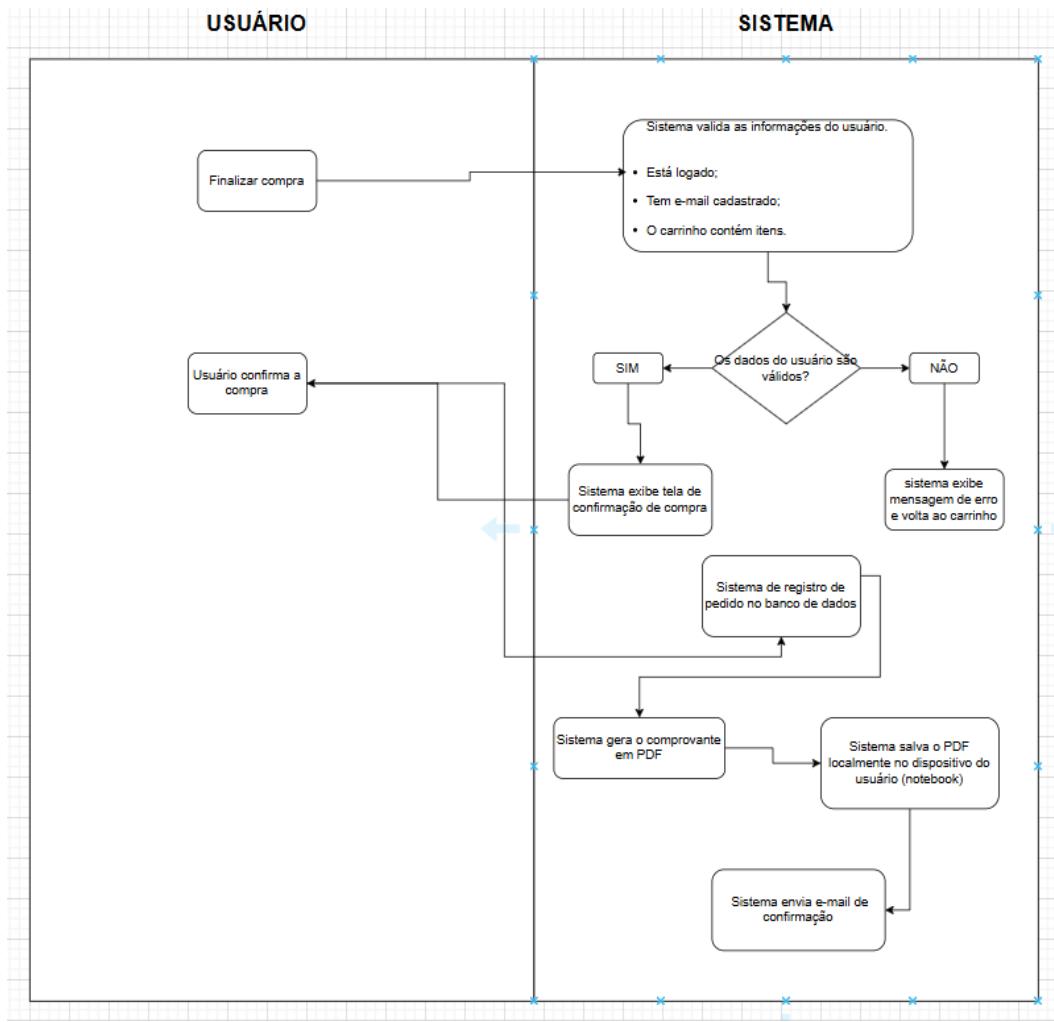


Figura 2: Diagrama de Atividades do processo de finalização de compra, ilustrando o fluxo de ações realizadas pelo usuário e pelo sistema, incluindo validação de dados, confirmação da compra, registro no banco de dados, geração do PDF e envio do e-mail de confirmação.

2.5 Diagrama de atividades – Geral

Além do processo específico de finalização de compra, foi elaborado um **Diagrama de Atividades Geral**, apresentado na Figura 3, que fornece uma visão simplificada de todas as principais funcionalidades do sistema. Esse diagrama ilustra as ações realizadas tanto pelo **usuário** quanto pelo **administrador**, incluindo visualização e pesquisa de produtos, gerenciamento do carrinho de compras, finalização de pedidos, acompanhamento do histórico de compras e operações administrativas, como adição, remoção e atualização de produtos e controle de pedidos. A representação permite compreender de forma geral a interação entre os atores e o sistema, evidenciando os fluxos de atividades e a separação de responsabilidades entre os diferentes perfis de acesso.

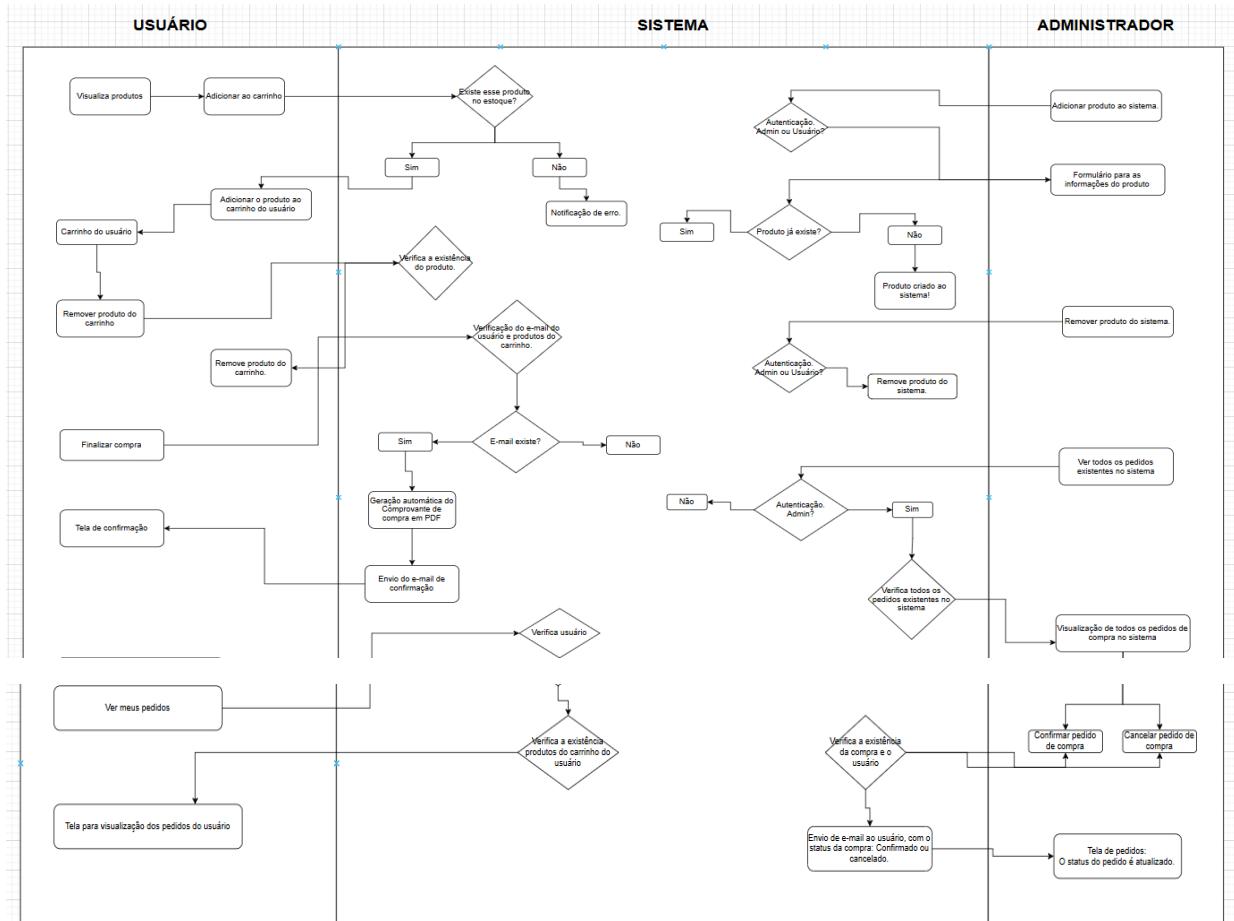


Figura 3: Diagrama de Atividades Geral do sistema de loja de utensílios artísticos, mostrando de forma simplificada todas as principais ações realizadas pelos atores Usuário e Administrador, incluindo visualização de produtos, gerenciamento de carrinho, finalização de compras, acompanhamento de pedidos e operações administrativas.

Conclusão

O desenvolvimento do sistema da loja de utensílios artísticos para pintura em tela permitiu criar uma plataforma **flexível, acessível e eficiente**, que atende tanto iniciantes quanto profissionais da área. O sistema oferece ferramentas para que os usuários explorem, adquiram e gerenciem produtos de maneira prática, proporcionando uma experiência de compra simples e organizada.

Além disso, o projeto evidencia a importância de **tecnologias modernas** e de uma arquitetura bem planejada para garantir a **segurança, o desempenho e a confiabilidade** da aplicação. A separação de perfis entre usuário e administrador assegura que as operações sejam realizadas de forma controlada, facilitando o gerenciamento dos produtos e pedidos.

Dessa forma, a loja torna-se mais do que um simples espaço de vendas: é um **ambiente pensado para quem ama a arte**, oferecendo materiais de qualidade e incentivando o desenvolvimento de novas técnicas e habilidades. O sistema cumpre o objetivo de unir **praticidade, eficiência e paixão pela arte**, contribuindo para que cada usuário possa aprimorar sua experiência artística de forma completa e satisfatória.

Referências

- PALLETS PROJECTS. *Flask Documentation (3.1.x)*. Disponível em: <https://flask.palletsprojects.com/>. Acesso em: 02 nov. 2025. [flask.palletsprojects.com+1](#)
- REACTJS. *React – A JavaScript library for building user interfaces*. Disponível em: <https://react.dev/>. Acesso em: 02 nov. 2025. [react.dev+1](#)
- DAMON, Craig Larman. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*. 3. ed. Boston: Pearson /Addison-Wesley, 2005. (Capítulo 6: Use Case Diagrams) [oreilly.com](#)