#### Relatório Consolidado do Projeto de Sistema de Vendas Online

## Capa

Título: Projeto de Sistema de Vendas Online

Aluno: Gustavo Fabiam

#### Índice

1. Introdução

- 2. Levantamento de Requisitos
- 3. Modelagem de Sistemas (UML)
- 4. Seleção Tecnológica
- 5. Síntese do Processo
- 6. Justificativa das Escolhas
- 7. Conclusão

## 1. Introdução

Este relatório consolida todas as etapas do projeto de desenvolvimento de um sistema de vendas online, incluindo levantamento de requisitos, modelagem UML, seleção tecnológica e decisões técnicas.

## 2. Levantamento de Requisitos

- Cliente: cadastrar conta, login, navegar produtos, adicionar ao carrinho, finalizar compra.
- Administrador: cadastrar produtos, atualizar estoque, gerar relatórios.
- Integração com sistema de pagamento externo (API).
- Atualização automática do estoque após cada venda.

# 3. Modelagem de Sistemas (UML)

## a) Diagrama de Caso de Uso

**Atores:** Cliente, Administrador, Sistema de Pagamento

Casos de uso: Cadastrar Conta, Login, Navegar Produtos, Adicionar ao Carrinho,

Finalizar Compra, Atualizar Estoque, Gerar Relatórios, Processar Pagamento

Cliente ----> Sistema

| Cadastrar Conta

| Login

| Navegar Produtos

| Adicionar Carrinho

| Finalizar Compra

Administrador ----> Sistema

| Cadastrar Produto

| Atualizar Estoque

| Gerar Relatórios

## b) Diagrama de Classes

#### Classe Atributos Métodos

Cliente nome, email, senha cadastrar(), login()

Produto nome, preço, quantidade atualizarEstoque()

Pedido itens, total, status calcularTotal(), finalizar()

Estoque listaProdutos atualizarProduto(), verificarDisponibilidade()

#### Relacionamentos:

- Pedido contém Produto
- Cliente realiza Pedido
- Estoque contém Produto

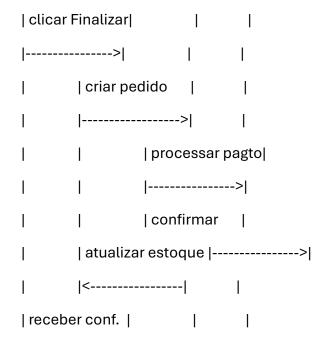
# c) Diagrama de Sequência / Fluxo

Fluxo: Finalizar Compra

- 1

Cliente Pedido SistemaPagto Estoque

1



# 4. Seleção Tecnológica

Categoria	Escolha	Justificativa Rápida
Front-end	React.js	Rápido, fácil de usar e integra com APIs
Back-end	Django	Seguro, bom desempenho e estrutura organizada
Banco de Dados	PostgreSQL	Confiável, suporta transações, fácil manutenção
API de Pagamento	Stripe	Seguro, confiável, fácil integração
Autenticação	JWT	Leve, seguro e compatível com APIs

# 5. Síntese do Processo

- Etapa 1 Requisitos: levantamento das funcionalidades do sistema.
- Etapa 2 Modelagem UML: criação de diagramas de caso de uso, classes e sequência.
- Etapa 3 Seleção Tecnológica: análise e escolha de frameworks, banco de dados e APIs baseando-se em desempenho, segurança, usabilidade e interoperabilidade.

## 6. Justificativa das Escolhas

- Cada requisito foi transformado em objetos e interações nos diagramas UML.
- As tecnologias escolhidas (React, Django, PostgreSQL, Stripe, JWT) suportam todos os requisitos, garantindo performance, segurança e integração.
- Ex.: fluxo de finalização de compra é suportado pelo Django + Stripe + JWT com atualização no PostgreSQL.

## 7. Conclusão

O relatório consolidado demonstra a coerência entre o problema, os requisitos, a modelagem do sistema e as tecnologias escolhidas. O sistema proposto atende às necessidades do cliente, garantindo confiabilidade, segurança e facilidade de uso.