

**GRUPO DE ESTUDO DE EDUCAÇÃO A DISTÂNCIA
CETEC - MODALIDADE EAD
TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS**

**Rodrigo Moro Batalha
Thamires Santos Valerio Silva
Wilker Branquinho Cunha
Wilson da Silva Castro
Vagner Fidelis**

Sistema de Gestão de Estoque para Empresas de Serviços

**São Paulo
2025**

Rodrigo Moro Batalha
Thamires Santos Valerio Silva
Wilker Branquinho Cunha
Wilson da Silva Castro
Vagner Fidelis

Sistema de Gestão de Estoque para Empresas de Serviços

Trabalho de Conclusão de Curso apresentado ao Curso Técnico de Análise e Desenvolvimento de Sistemas, como parte dos requisitos necessários à obtenção do título de Técnico de Desenvolvimento de Sistemas.

Orientador: Rodrigo Bressan

São Paulo
2025

Dedicatória. . .

- Dedico este projeto à minha família e amigos que sempre estiveram presentes direta ou indiretamente em todos os momentos de minha formação.

O único lugar onde o sucesso vem antes do trabalho é no dicionário.

Resumo

O presente trabalho apresenta o desenvolvimento de um Sistema de Gestão de Estoque voltado para empresas de serviços, como salões de beleza, oficinas mecânicas, clínicas de estética e empresas de limpeza. A proposta visa otimizar o controle de insumos, evitando desperdícios, atrasos e falhas no abastecimento. O sistema foi implementado utilizando HTML, CSS e JavaScript, garantindo acessibilidade por navegadores e compatibilidade com múltiplos dispositivos. Por meio da metodologia ágil SCRUM, o projeto foi dividido em sprints com entregas incrementais, abrangendo funcionalidades como cadastro de produtos, controle de estoque em tempo real, geração de alertas automáticos, relatórios detalhados e controle de acesso por permissões. A modelagem de dados, os testes automatizados e as medidas de segurança foram planejados para assegurar a escalabilidade e confiabilidade da solução. Os resultados obtidos demonstram um sistema funcional, intuitivo e eficaz na gestão de recursos em ambientes de serviços.

Palavras-chave: gestão de estoque, sistema web, SCRUM, controle de insumos, empresas de serviços.

Abstract

This study presents the development of an Inventory Management System tailored for service companies such as beauty salons, auto repair shops, aesthetic clinics, and cleaning services. The goal is to optimize supply control, avoiding waste, delays, and replenishment failures. The system was developed using HTML, CSS, and JavaScript, ensuring browser accessibility and multi-device compatibility. Through the agile SCRUM methodology, the project was structured in sprints with incremental deliveries, covering functionalities like product registration, real-time inventory control, automatic alerts, detailed reporting, and permission-based user access. Data modeling, automated testing, and security protocols were planned to ensure system scalability and reliability. The final results demonstrate a functional, intuitive, and efficient system for managing resources in service-based environments.

Keywords: inventory management, web system, SCRUM, supply control, service companies.

Lista de ilustrações

Figura 1 – Fluxograma do Sistema	17
Figura 2 – Legenda	19
Figura 3 – Tela de Login	21
Figura 4 – Tela de cadastro dos produtos	22
Figura 5 – Tela de Relatório dos Produtos Cadastrados	23
Figura 6 – Burndown Chart - Sprint de 14 dias	24

Sumário

1	INTRODUÇÃO	9
1.1	Justificativa	9
1.2	Objetivo	9
1.2.1	Objetivo Geral	9
1.2.2	Objetivos Específicos	10
2	METODOLOGIA	11
2.1	Identificação do Software	11
2.2	Tecnologias Utilizadas	11
2.3	Requisitos Funcionais e Não Funcionais	11
2.3.1	Requisitos Funcionais	11
2.3.2	Requisitos Não Funcionais	12
2.4	Descrição do Software	12
2.5	Equipe e Partes Interessadas	12
2.5.1	Equipe de Desenvolvimento	12
2.5.2	Partes Interessadas	13
2.6	Recursos Utilizados	13
2.6.1	Equipamentos e Ferramentas	13
2.7	Metodologia e Processos	13
2.7.1	Estrutura do SCRUM	14
2.7.2	Planejamento das Sprints	14
2.7.3	Artefatos do SCRUM	14
3	DESENVOLVIMENTO	16
3.1	Diagrama de casos de uso	16
3.1.1	Diagrama de Casos de Uso	16
3.1.2	Diagrama de Classes	17
3.2	Desenvolvimento das Estruturas de Banco de Dados	17
3.2.1	Principais Tabelas	17
3.2.2	Relacionamentos	18
3.2.3	Regras de Integridade e Constraints	18
3.2.4	Dicionário de Dados – Exemplo Simplificado	19
3.3	Implementação de Sistema	19
3.4	Mockup de Telas	20
3.5	Cronograma	23
4	CONSIDERAÇÕES FINAIS	25

REFERÊNCIAS BIBLIOGRÁFICAS	27
ANEXOS	28

1 INTRODUÇÃO

A gestão eficiente de estoques é um dos principais fatores que contribuem para a sustentabilidade e o bom funcionamento das empresas de serviços. A falta de controle adequado pode acarretar prejuízos financeiros, interrupções no atendimento, desperdícios e perda de competitividade. Ao contrário do setor varejista, que normalmente conta com soluções especializadas, empresas de serviços muitas vezes negligenciam essa etapa, lidando com insumos e materiais de forma manual e desorganizada.

Neste cenário, surge a necessidade de desenvolver uma solução tecnológica acessível, intuitiva e eficaz, capaz de atender às demandas operacionais dessas organizações. Este trabalho apresenta o projeto e a implementação de um sistema web de gestão de esto- que voltado especificamente para empresas de serviços, como salões de beleza, oficinas mecânicas, clínicas e prestadores autônomos. A proposta é entregar uma plataforma prática e adaptável, com recursos modernos e alinhados às boas práticas de desenvolvimento e usabilidade.

1.1 Justificativa

A escolha deste tema está fundamentada na observação recorrente de falhas na gestão de insumos em empresas de pequeno e médio porte que atuam na prestação de serviços. O uso de planilhas manuais, blocos de anotações ou mesmo a ausência total de registro sistemático compromete a organização e o controle de materiais, gerando impactos diretos na qualidade do serviço e na rentabilidade do negócio.

Segundo Garrett (2011), uma experiência de usuário bem projetada facilita a adoção de novas tecnologias e aumenta a eficiência dos processos internos. Já Nielsen (2020) reforça a importância da acessibilidade e da simplicidade das interfaces para garantir que o sistema atenda a diferentes perfis de usuários. Assim, ao considerar aspectos como usabilidade, acessibilidade e compatibilidade com múltiplos dispositivos, este projeto se propõe a preencher essa lacuna com uma ferramenta digital eficiente.

Além disso, a implementação de um sistema automatizado contribui para decisões mais assertivas, redução de perdas e ganho de produtividade, temas diretamente ligados à competitividade e à profissionalização das empresas.

1.2 Objetivo

1.2.1 Objetivo Geral

Desenvolver um sistema web de gestão de estoque voltado para empresas de serviços, com foco na automação de processos operacionais, controle eficiente de insumos e materiais, e geração de relatórios gerenciais. A solução deve apresentar uma interface

acessível e responsiva, garantir segurança dos dados, permitir múltiplos níveis de acesso e proporcionar suporte à tomada de decisões estratégicas, contribuindo para a redução de desperdícios, aumento da produtividade e melhoria da qualidade dos serviços prestados.

1.2.2 Objetivos Específicos

- Levantar os principais requisitos funcionais e não funcionais de um sistema de controle de estoque voltado para empresas de serviços;
- Elaborar a modelagem de dados e os diagramas de casos de uso e classes do sistema;
- Implementar uma plataforma responsiva utilizando tecnologias como HTML, CSS, JavaScript e banco de dados relacional;
- Garantir diferentes níveis de acesso ao sistema, com permissões específicas para administradores, gerentes e operadores;
- Criar relatórios dinâmicos com filtros e exportação de dados para PDF ou CSV;
- Realizar testes de funcionalidade e usabilidade, com foco na experiência do usuário;
- Validar o sistema com base nos critérios de desempenho, segurança e organização do estoque.

2 METODOLOGIA

A metodologia utilizada no desenvolvimento deste projeto foi a ágil SCRUM, amplamente adotada na engenharia de software devido à sua capacidade de adaptação, entrega contínua e forte colaboração entre os membros da equipe. O projeto foi dividido em sprints quinzenais, com entregas incrementais, revisões frequentes e adaptação contínua conforme o feedback obtido.

2.1 Identificação do Software

O software desenvolvido é um Sistema de Gestão de Estoque para Empresas de Serviços, voltado ao controle eficiente de produtos e insumos em estabelecimentos como salões de beleza, oficinas mecânicas, serviços de limpeza e clínicas de estética. O sistema busca facilitar o acompanhamento das movimentações de estoque, garantir alertas automáticos e gerar relatórios que apoiem a tomada de decisões.

2.2 Tecnologias Utilizadas

As tecnologias selecionadas têm como base a acessibilidade, a responsividade e a compatibilidade com múltiplos dispositivos e navegadores. A escolha foi orientada pela simplicidade de uso e facilidade de manutenção:

- Frontend: HTML5, CSS3 e JavaScript;
- Backend e Banco de Dados: Node.js (Express.js) com MySQL (v5.7+);
- Design de Interface: Figma;
- Infraestrutura: Acesso via navegador, compatível com desktops e dispositivos móveis;
- Controle de Versão: GitHub;
- Deploy: Plataforma Vercel (homologação) + banco ClearDB (MySQL na nuvem).

2.3 Requisitos Funcionais e Não Funcionais

2.3.1 Requisitos Funcionais

- Cadastro, edição e exclusão de produtos por categoria;
- Controle de movimentações de entrada e saída de insumos;
- Geração de relatórios detalhados com filtros de período;

- Alertas automáticos de estoque mínimo;
- Controle de acesso baseado em papéis (RBAC).

2.3.2 Requisitos Não Funcionais

- Interface gráfica amigável e responsiva, seguindo boas práticas de UX;
- Compatibilidade com navegadores modernos (Chrome, Firefox, Edge, Safari);
- Armazenamento seguro dos dados, com controle de acesso por autenticação;
- Arquitetura modular e escalável para futuras melhorias.

2.4 Descrição do Software

O sistema contempla os seguintes recursos principais:

- Cadastro e edição de produtos por categoria;
- Controle de estoque em tempo real com atualização automática;
- Registro completo das movimentações (entradas e saídas);
- Geração de relatórios analíticos com opção de exportação;
- Controle de usuários com permissões específicas conforme o perfil (administrador, gerente, operador).

2.5 Equipe e Partes Interessadas

2.5.1 Equipe de Desenvolvimento

- Scrum Master: Vagner de Jesus Fidelis
- Product Owner: Rodrigo Moro Batalha
- Desenvolvedor: Wilson da Silva Castro
- QA (Analista de Testes): Thamires Santos Valerio Silva
- Documentação e Negócios: Wilker Branquinho Cunha

2.5.2 Partes Interessadas

- Proprietários de empresas prestadoras de serviços;
- Gerentes e administradores;
- Funcionários operacionais (estoquistas, atendentes);
- Fornecedores;
- Clientes finais (beneficiados indiretamente por um serviço mais eficiente).

2.6 Recursos Utilizados

2.6.1 Equipamentos e Ferramentas

- Plataformas de Gerenciamento: Trello ou Jira (organização de tarefas);
- Controle de Versão: GitHub ou GitLab;
- Comunicação da Equipe: Discord (reuniões diárias/semanais);
- Documentação: Excel, Google Docs;
- Infraestrutura: Servidores em nuvem;
- Hardware:
 - Computadores com processadores Intel Core i5 ou superior, 8GB RAM e SSD;
 - Monitores Full HD para testes de interface;
 - Ambiente virtualizado para testes entre navegadores;
 - Internet estável $\geq 100\text{Mbps}$ para integração contínua.

2.7 Metodologia e Processos

Para garantir uma abordagem estruturada, colaborativa e iterativa no desenvolvimento do sistema, foi adotada a metodologia ágil SCRUM. Essa metodologia é amplamente utilizada no desenvolvimento de software por sua capacidade de promover entregas frequentes, adaptação constante e foco na satisfação do usuário.

O projeto foi dividido em quatro sprints, cada uma com duração de aproximadamente duas semanas. Em cada sprint foram definidas tarefas prioritárias, metas claras e entregas incrementais, possibilitando a evolução contínua da aplicação e a rápida identificação de ajustes necessários.

A equipe seguiu os rituais do SCRUM, como planejamento da sprint (Sprint Planning), reuniões diárias (Daily Scrum), revisão da sprint (Sprint Review) e retrospectiva (Sprint Retrospective). Essas práticas favoreceram a comunicação eficiente, a identificação de impedimentos e a melhoria contínua do processo.

2.7.1 Estrutura do SCRUM

A estrutura do SCRUM utilizada no projeto foi composta por:

- Product Backlog: lista priorizada de funcionalidades e requisitos definidos pelo Product Owner com base nas necessidades dos usuários;
- Sprint Backlog: subconjunto do Product Backlog, composto pelas tarefas que seriam executadas durante a sprint em andamento;
- Burndown Chart: gráfico de acompanhamento da produtividade da equipe, representando o progresso diário na execução das tarefas planejadas.

2.7.2 Planejamento das Sprints

A divisão do projeto em sprints seguiu a seguinte organização, conforme planejamento registrado nas reuniões de acompanhamento:

- Sprint 1: Desenvolvimento dos módulos de CRUD de Produtos e Movimentações, com rotas implementadas em `/routes/produtos.js` e modelos de dados em `/models/Produto.js`. Interface construída em `produtos.html`.
- Sprint 2: Implementação dos alertas automáticos de estoque mínimo com lógica de validação no arquivo `alertaEstoque.js`, além do controle de permissões baseado em níveis de acesso (RBAC) por meio de middleware de autenticação (`auth.js`).
- Sprint 3: Criação de relatórios dinâmicos e dashboard analítico, com rotas específicas em `/routes/relatorios.js` e visualização em `dashboard.html`.
- Sprint 4: Aplicação de testes unitários, refino da interface (UI) e ajustes finais para entrega. Foram utilizados testes com Jest, Supertest e Playwright em rotinas localizadas na pasta `/tests/`.

2.7.3 Artefatos do SCRUM

Durante a execução do projeto, foram utilizados diversos artefatos que documentaram e organizaram o processo:

- Product Backlog: disponível na plataforma Trello, com cartões representando funcionalidades e melhorias priorizadas;
- Sprint Backlog: conjunto de tarefas extraídas do backlog geral e atribuídas aos membros da equipe;
- Burndown Chart: gráficos atualizados diariamente, auxiliando no monitoramento do desempenho e previsão de conclusão da sprint.

3 DESENVOLVIMENTO

Nesta etapa são apresentados os principais aspectos técnicos da construção do sistema, desde a modelagem conceitual e lógica até a implementação prática e a prototipação das interfaces. Os anexos ao final deste trabalho apresentam as telas desenvolvidas, trechos de código e arquivos de estilo que evidenciam a implementação detalhada do projeto.

3.1 Diagrama de casos de uso

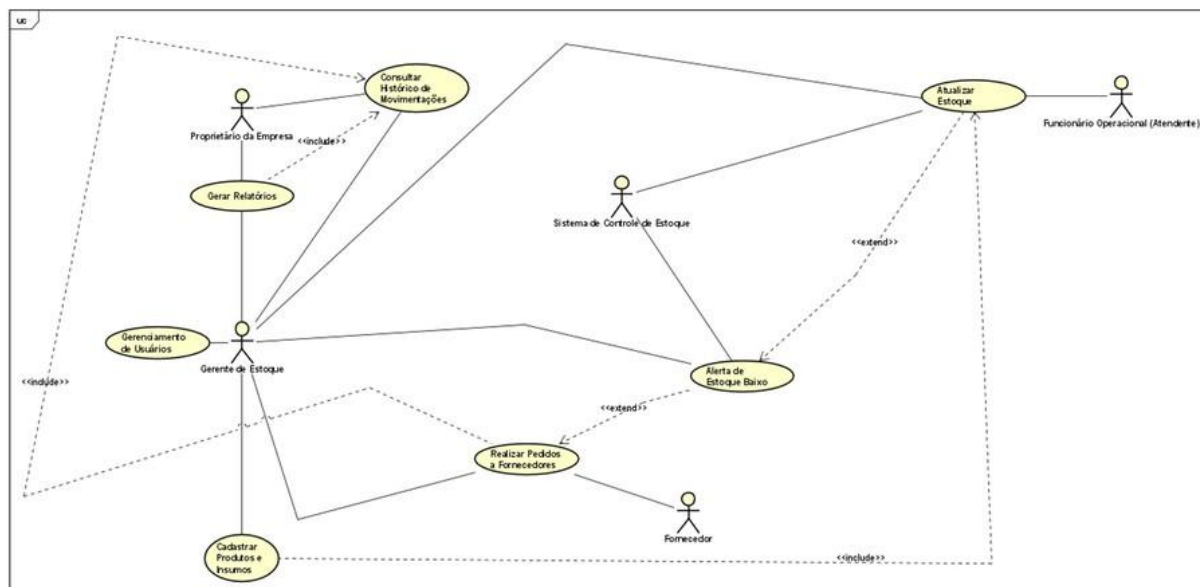
A modelagem do sistema foi essencial para garantir organização, previsibilidade e escalabilidade. Foram utilizados diagramas de caso de uso e classes, que permitiram estruturar as funcionalidades e os componentes técnicos.

3.1.1 Diagrama de Casos de Uso

O diagrama de casos de uso representou as principais funcionalidades do sistema e os atores envolvidos, como Gerente de Estoque, Funcionário Operacional, Fornecedor e Sistema. Entre os principais casos de uso estão:

- Cadastro e edição de produtos e insumos;
- Atualização automática de estoque;
- Alerta de estoque baixo;
- Geração de relatórios com filtros;
- Controle de acesso por níveis.

Figura 1 – Fluxograma do Sistema



Própria

3.1.2 Diagrama de Classes

As entidades do sistema foram mapeadas no diagrama de classes, incluindo: Produto, Movimentação, Fornecedor, Usuário, e suas respectivas relações. Essa estrutura fundamentou a criação das tabelas do banco de dados.

3.2 Desenvolvimento das Estruturas de Banco de Dados

A estrutura do banco de dados foi planejada com base em princípios de normalização e integridade referencial, visando garantir a consistência dos dados, a escalabilidade do Sistema e a facilidade de manutenção.

O banco de dados utilizado foi o MySQL 5.7+, comum modelo relacional que atende diretamente aos requisitos funcionais definidos durante as fases iniciais do projeto. A modelagem foi fundamentada nos diagramas de classes e nos casos de uso descritos na fase de análise, conforme demonstrado na Agenda 4.

3.2.1 Principais Tabelas

As tabelas foram criadas para representar de forma clara as entidades do sistema, seus atributos e suas relações. A seguir, as principais tabelas do sistema:

- produtos: armazena os dados de cada item controlado pelo estoque (nome, categoria, quantidade, estoque mínimo, valor, etc.);

- **movimentacoes**: registra todas as entradas e saídas do estoque, com data, tipo (entrada/saída), quantidade e produto relacionado;
- **usuarios**: contém informações dos usuários cadastrados no sistema, como nome, e-mail, senha (hash), e perfil de acesso (admin, gerente, operador);
- **fornecedores**: armazena os dados dos fornecedores cadastrados, com nome, contato e relação com os produtos;
- **pedido_reposicao** e **pedido_itens**: representam os pedidos de reposição de estoque gerados automaticamente pelo sistema quando o nível mínimo é atingido.

3.2.2 Relacionamentos

A modelagem do banco considera os seguintes relacionamentos principais:

- Um produto pode estar em **várias** movimentações (1:N);
- Um fornecedor pode fornecer **múltiplos** produtos (1:N);
- Um usuário pode estar vinculado a **diversas** movimentações (1:N);
- Cada pedido de reposição pode conter **múltiplos** itens (1:N).

Essas relações foram implementadas por meio de chaves estrangeiras com restrições ON DELETE RESTRICT para evitar exclusões acidentais que comprometam o histórico do sistema.

3.2.3 Regras de Integridade e Constraints

Para garantir a confiabilidade do sistema e evitar inconsistências nos dados, foram aplicadas diversas restrições (constraints):

- CHECK (estoque_atual >= 0) – impede que o estoque de um produto fique negativo;
- UNIQUE(email) – assegura que não haja usuários duplicados;
- CHECK (tipo IN ('E','S')) – restringe o tipo de movimentação às opções válidas (Entrada ou Saída);
- FOREIGN KEY (id_produto) – mantém a integridade entre as tabelas produtos e movimentacoes;
- Índices nos campos utilizados em filtros e relatórios, como data, id_produto e categoria, para melhor performance de consulta.

3.2.4 Dicionário de Dados – Exemplo Simplificado

Figura 2 – Legenda

Tabela	Campo	Tipo	Regra / Validação
produtos	id_produto	INT PK AI	>0
	nome	VARCHAR(80)	NOT NULL
	categoria	VARCHAR(40)	NOT NULL
	estoque_atual	INT	>=0
	estoque_min	INT	>=0, default 5
	valor_unit	DEC(10,2)	>=0
movimentacoes	tipo	CHAR(1)	'E' ou 'S'
	qtd	INT	>0

(Tabela completa no repositório SQL.)

Própria

3.3 Implementação de Sistema

A implementação do sistema foi realizada de forma incremental, seguindo o planejamento das sprints da metodologia SCRUM. Cada ciclo de desenvolvimento teve metas específicas e entregas funcionais, permitindo a construção progressiva do sistema e facilitando os ajustes com base no feedback do grupo e do professor orientador.

Durante o processo, foram adotadas boas práticas de organização do código, versionamento via GitHub e testes contínuos. A seguir, são descritas as principais entregas de cada sprint:

- **Sprint 1 – Cadastro de Produtos e Movimentações:**
Nesta etapa, foi implementado o módulo CRUD de produtos e o controle básico de estoque. As movimentações (entradas e saídas) passaram a ser registradas e vinculadas a produtos específicos. Os arquivos relevantes encontram-se descritos nos Anexos B.1 e B.3.
- **Sprint 2 – Alertas e Controle de Acesso:**
Foi criada a lógica de alerta automático de estoque baixo. Também foi implementado o sistema de autenticação e controle de permissões baseado em níveis de acesso.

(RBAC), com redirecionamento para áreas específicas. Os detalhes técnicos estão no Anexo B.2.

- **Sprint 3 – Relatórios e Dashboard:**
Foram desenvolvidos gráficos de movimentação e relatórios com filtros por data. A visualização desses dados permite ao gerente e ao proprietário monitorarem o desempenho do estoque e preverem necessidades futuras. A funcionalidade de exportação para PDF e CSV também foi incluída.
- **Sprint 4 – Testes, Ajustes e Homologação:**
A última sprint foi voltada à finalização do projeto. Foram realizados testes unitários (com Jest), testes de integração (com Supertest) e testes E2E (com Playwright). Além disso, foram feitos ajustes na interface com base nos princípios de usabilidade e acessibilidade (Krug, 2014; Nielsen, 2020).

A estrutura modular do código e o uso de JavaScript moderno permitiram a criação de um sistema robusto, acessível e de fácil manutenção. A aplicação foi implantada em ambiente de homologação, conforme demonstrado nas documentações técnicas complementares.

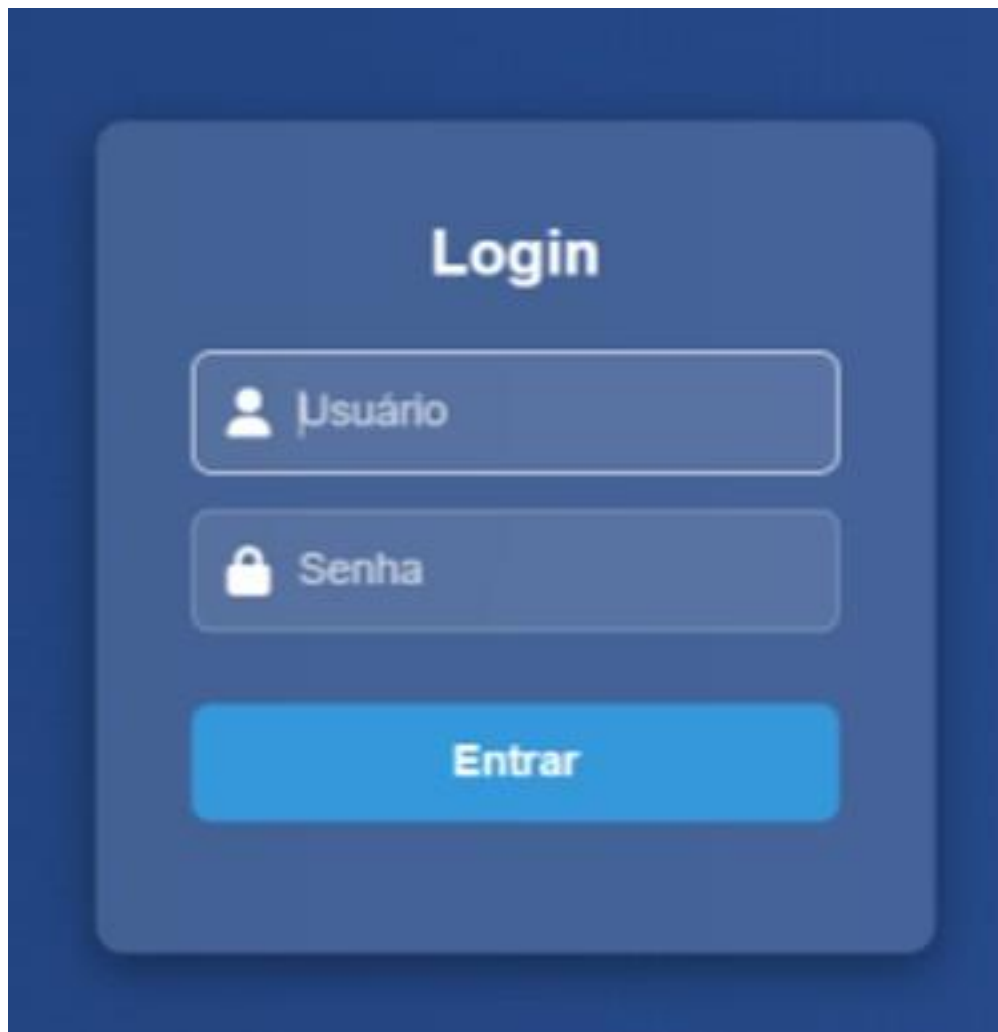
3.4 Mockup de Telas

O design das interfaces foi baseado em conceitos de usabilidade, hierarquia visual e design responsivo, com foco na simplicidade e na eficiência do uso. Os protótipos foram inicialmente desenvolvidos no Figma e depois implementados com HTML, CSS e JavaScript.

- **Tela de Login**

Permite o acesso ao sistema mediante autenticação por e-mail e senha. A interface adapta-se ao tipo de usuário autenticado, redirecionando conforme o nível de permissão (administrador, gerente ou operador).

Figura 3 – Tela de Login



Própria

- Dashboard

Painel principal do sistema, que exibe indicadores visuais como o total de produtos cadastrados, quantidade de itens com estoque crítico e movimentações recentes. Utiliza gráficos dinâmicos para facilitar a interpretação dos dados.

Figura 4 – Tela de cadastro dos produtos



A interface de usuário para o cadastro de produtos, exibida em um tema escuro. O formulário contém campos para Nome do Produto, Preço Unitário, Quantidade e Data de Inserção. Abaixo dos campos, há três botões: 'Cadastrar Produto' (verde), 'Sair' (vermelho) e 'Relatório de Produtos' (verde).

Cadastro de Produtos

Nome do Produto:

Preço Unitário:

Quantidade:

Data de Inserção:

Cadastrar Produto

Sair

Relatório de Produtos

Própria

- Tela de Relatórios

Disponibiliza relatórios de consumo e movimentações de estoque, com filtros por

período. É possível exportar os dados em formatos PDF e CSV para fins administrativos e contábeis.

Figura 5 – Tela de Relatório dos Produtos Cadastrados

Relatório de Produtos Cadastrados					
Nome do Produto	Quantidade	Valor Unitário	Valor Total	Data de Cadastro	
Bola	10	R\$1.50	R\$15.00	05/04/2025	
Bola	100	R\$12.00	R\$1200.00	01/01/2001	
Bola	10	R\$1.50	R\$15.00	17/04/2025	
Valor total do estoque: R\$1230.00					
<div>    </div>					

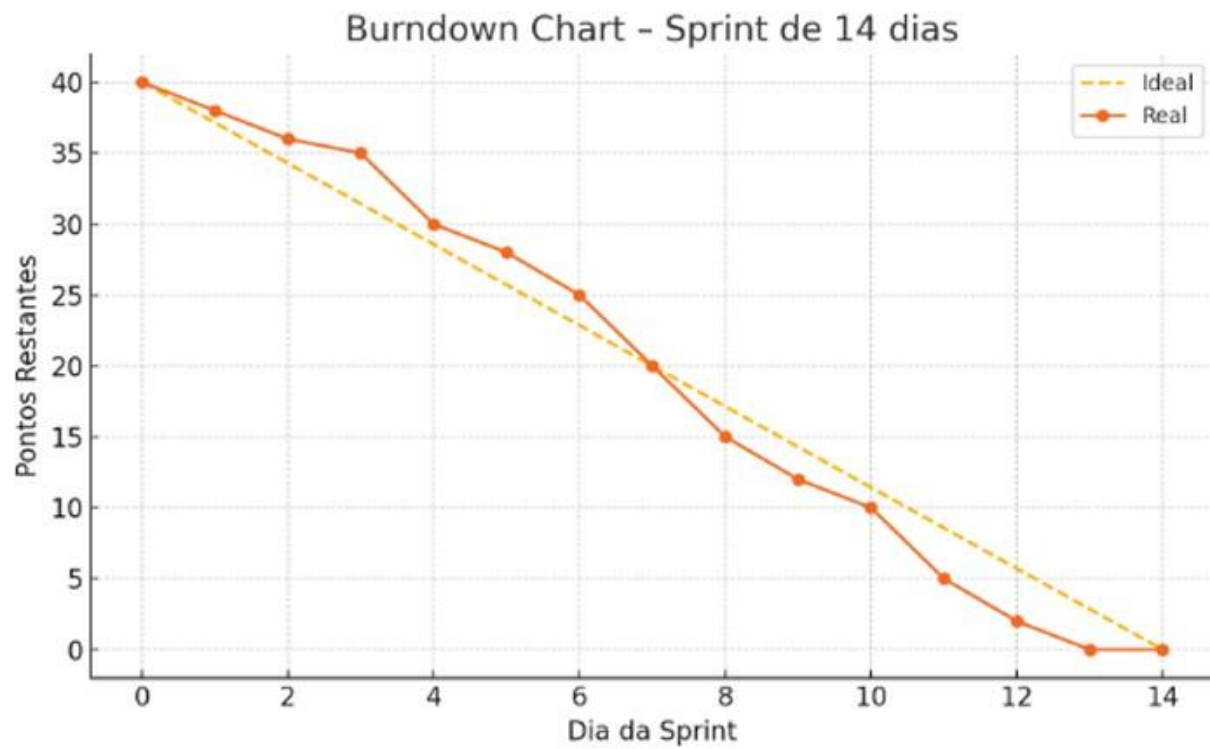
Própria

A estilização das interfaces, descrita nos Anexos C.1 a C.3, obedece aos princípios de acessibilidade definidos por Nielsen (2020) como contraste de cores, responsividade e legibilidade. O layout é compatível com dispositivos móveis e navegadores modernos.

3.5 Cronograma

Seguindo a metodologia acertada e utilizando de reuniões periódicas para revisão da Sprint a equipe conseguiu o seguinte resultado em 15 dias:

Figura 6 – Burndown Chart - Sprint de 14 dias



Própria

4 CONSIDERAÇÕES FINAIS

O desenvolvimento do Sistema de Gestão de Estoque para Empresas de Serviços demonstrou a relevância e aplicabilidade dos conhecimentos adquiridos ao longo do curso técnico em Desenvolvimento de Sistemas. Por meio de uma abordagem prática e orientada a projetos, foi possível transformar uma necessidade real do mercado — a dificuldade de controle de insumos e materiais em pequenas e médias empresas — em uma solução tecnológica funcional, eficiente e acessível.

A escolha da metodologia SCRUM se mostrou acertada, permitindo à equipe organizar o desenvolvimento em ciclos curtos e incrementais. As sprints possibilitaram entregas contínuas, validações frequentes e ajustes ágeis, promovendo maior engajamento e comprometimento com a qualidade do produto. Os artefatos produzidos (Product Backlog, Sprint Backlog, Burndown Charts), bem como as reuniões diárias e revisões, contribuíram significativamente para a fluidez do trabalho colaborativo, conforme evidenciado na Agenda 5.

Do ponto de vista técnico, a solução contemplou todos os requisitos propostos: cadastro de produtos, controle de movimentações, alertas de estoque crítico, relatórios com filtros e exportação, além de controle de acesso com níveis de permissão. A implementação com HTML, CSS, JavaScript e MySQL resultou em um sistema leve, responsivo e de fácil utilização, compatível com diversos navegadores e dispositivos. Além disso, foram aplicados princípios de usabilidade e acessibilidade para garantir uma experiência satisfatória para todos os perfis de usuários.

Destaca-se ainda a adoção de práticas de qualidade de software, como testes unitários, de integração e E2E, o que aumentou a confiabilidade do sistema e reduziu falhas em tempo de execução. A organização modular do código e a estrutura normalizada do banco de dados garantem facilidade de manutenção e futuras expansões.

Como continuidade e aprimoramento do projeto, são recomendadas as seguintes

evoluções:

- Integração com APIs de fornecedores, possibilitando pedidos automáticos de reposição;
- Desenvolvimento de um aplicativo mobile, com as principais funcionalidades do sistema web;
- Implantação de inteligência artificial para previsão de consumo, com base em dados históricos;
- Aplicação de critérios avançados de segurança, como autenticação multifator e criptografia de dados sensíveis.

Conclui-se que o sistema desenvolvido cumpre seu propósito de maneira eficaz, oferecendo uma ferramenta viável para a digitalização e automação da gestão de estoque em empresas de serviços. Além disso, o projeto consolidou o aprendizado técnico e fortaleceu habilidades como trabalho em equipe, pensamento crítico e capacidade de resolver problemas reais com soluções tecnológicas.

REFERÊNCIAS BIBLIOGRÁFICAS

TUBINO, D. F. Planejamento e Controle da Produção: Teoria e Prática. São Paulo: Atlas, 2007.

GARRETT, Jesse James. The Elements of User Experience: User-Centered Design for the Web and Beyond. 2. ed. Berkeley, CA: New Riders, 2011.

NORMAN, Donald A. The Design of Everyday Things. Revised and Expanded Edition. New York, NY: Basic Books, 2013.

KRUG, Steve. Don't Make Me Think: A Common Sense Approach to Web Usability. 3. ed. Berkeley, CA: New Riders, 2014.

Anexos

Anexo A – Interface do Sistema (Mockups Reais)

As telas abaixo representam as principais interfaces desenvolvidas no sistema de gestão de estoque. Elas foram implementadas com HTML, CSS e JavaScript e refletem a experiência de navegação pensada para usuários com diferentes níveis de acesso.

• A.1 – Tela de Login

Arquivo: Login/login.html

Descrição: Tela inicial do sistema, onde o usuário informa email e senha para autenticação. Após o login, é redirecionado conforme seu nível de permissão (administrador, gerente ou operador).

• A.2 – Dashboard

Arquivo: DashBoard/dashboard.html

Descrição: Painel principal com indicadores (KPI), gráficos gerados via Chart.js e atalhos para produtos, relatórios e movimentações. Organizado com layout responsivo.

• A.3 – Tela de Relatórios

Arquivo: Relatorio/report.html

Descrição: Tela para geração de relatórios com filtros por período, exportação para PDF/CSV e visualização detalhada das movimentações do estoque.

Anexo B – Scripts e Funções de Aplicação

A seguir, estão os principais scripts JavaScript utilizados na implementação da lógica do sistema. Cada um deles está associado a funcionalidades específicas do projeto.

• B.1 –Script do Dashboard

Arquivo: DashBoard/script.js

Função: Responsável por exibir os dados dinâmicos no painel principal, como produtos em estoque, total de movimentações, e geração de gráficos com base em dados do banco.

• B.2 – Script de Login

Arquivo: Login/scriptlogin.js

Função: Realiza a autenticação do usuário, verifica o perfil (RBAC) e redireciona para a área correspondente do sistema.

• B.3 – Script de Relatórios

Arquivo: Relatorio/script-report.js

Função: Permite ao usuário filtrar as movimentações por período, gerar relatórios analíticos e exportar os dados em formatos compatíveis como PDF e CSV.

Anexo C – Estilização do Sistema (CSS)

A interface visual do sistema foi construída com folhas de estilo CSS específicas para cada módulo, respeitando princípios de responsividade, contraste e acessibilidade.

• C.1 – Estilos do Dashboard

Arquivo: DashBoard/styles.css

Descrição: Define o layout dos cards de indicadores, grid de produtos, cores dos gráficos e organização visual do painel principal.

• C.2 – Estilos da Tela de Login

Arquivo: Login/styleslogin.css

Descrição: Define a aparência dos campos de entrada, botões e mensagens de erro da tela de autenticação. Foco em contraste e simplicidade.

• C.3 – Estilos da Página de Relatórios

Arquivo: Relatorio/styles-report.css

Descrição: Controla a disposição das tabelas de dados, filtros de período e botões de exportação. Compatível com impressão e responsivo para visualização mobile.