

IFSP - Instituto Federal de Educação, Ciência e Tecnologia
Câmpus São Paulo

BEATRIZ MUNIZ DE BARROS	SP3161315
GEAN CARLOS DE SOUSA BANDEIRA	SP3030075
KHALIL KHALID ABOU ANCHE	SP3121925
MARCELO FLORES VALDEZ	SP3039056
MATHEUS PRANDO APPOLINARIO BARBOSA	SP3121747
RAFAEL VALVERDE ZANATA DA SILVA	SP3119866
VITOR DA SILVA OLIVEIRA	SP3020589

VIP PENHA

São Paulo - SP - Brasil

2025

IFSP - Instituto Federal de Educação, Ciência e Tecnologia
Câmpus São Paulo

BEATRIZ MUNIZ DE BARROS	SP3161315
GEAN CARLOS DE SOUSA BANDEIRA	SP3030075
KHALIL KHALID ABOU ANCHE	SP3121925
MARCELO FLORES VALDEZ	SP3039056
MATHEUS PRANDO APPOLINARIO BARBOSA	SP3121747
RAFAEL VALVERDE ZANATA DA SILVA	SP3119866
VITOR DA SILVA OLIVEIRA	SP3020589

VIP PENHA

Projeto desenvolvido como parte das atividades da disciplina Projeto Integrado, apresentado ao Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, Câmpus São Paulo, curso de Tecnologia em Análise e Desenvolvimento de Sistemas.

IFSP - Instituto Federal de Educação, Ciência e Tecnologia
Câmpus São Paulo
Tecnologia em Análise e Desenvolvimento de Sistemas

São Paulo - SP - Brasil

2025

Resumo

Este projeto descreve a criação de um sistema de controle de estoque para a loja de eletrônicos VIP PENHA. Em um cenário de rápido avanço tecnológico e crescente competitividade de mercado, a gestão eficiente de recursos é essencial para empreendimentos. O sistema foi desenvolvido para otimizar o controle de produtos, tornando a operação mais eficiente.

A problemática central é a falta de controle de estoque e os erros de processos manuais na "VIP PENHA". A solução implementada é um sistema unificado que permite o cadastro de produtos, o controle de entradas e saídas de mercadorias, e a geração automática de relatórios gerenciais. Isso resultou na redução de erros manuais, na diminuição do tempo gasto na gestão do estoque e na melhoria da tomada de decisões estratégicas, oferecendo uma perspectiva clara sobre o desempenho do inventário.

Para o desenvolvimento do sistema, foram realizadas pesquisas sobre gestão de estoque e sistemas de informação, utilizando uma metodologia de trabalho ágil, o Kanban. O projeto abrangeu o detalhamento de funcionalidades, requisitos técnicos, arquitetura do sistema (front-end, back-end, banco de dados e infraestrutura), tecnologias empregadas, planos de testes e planejamentos financeiros.

Palavras-chave: Sistema de estoque. Gestão de produtos. Automação. Relatórios gerenciais. Kanban.

Abstract

This project describes the creation of an inventory control system for the VIP PENHA electronics store. In a scenario of rapid technological advancement and increasing market competitiveness, efficient resource management is essential for businesses. The system was developed to optimize product control, making operations more efficient.

The central problem addressed is the lack of inventory control and manual process errors at "VIP PENHA". The implemented solution is a unified system that allows product registration, control of goods entry and exit, and automatic generation of management reports. This resulted in the reduction of manual errors, decreased time spent on inventory management, and improved strategic decision-making, offering a clear perspective on inventory performance.

For the system's development, research on inventory management and information systems was conducted, utilizing an agile methodology, Kanban. The project covered the detailing of functionalities, technical requirements, system architecture (front-end, back-end, database, and infrastructure), technologies employed, test plans, and financial planning.

Keywords: Inventory management system. Product management. Automation. Management reports. Kanban.

Lista de ilustrações

Figura 1 – QR Code para o repositório no GitHub	21
Figura 2 – Diagrama de Componentes	32
Figura 3 – Diagrama de Implantação	33
Figura 4 – Resultado do SSL Labs	38
Figura 5 – Resultado do Security Headers	40
Figura 6 – Modelo de Entidade-Relacionamento	44
Figura 7 – Diagrama de Entidade-Relacionamento	45

Lista de quadros

Quadro 1 – Comparação entre Sistemas de Gerenciamento de Estoque	14
Quadro 2 – Regras de Negócio (RN01 a RN04)	22
Quadro 3 – Regras de Negócio (RN05 a RN09)	23
Quadro 4 – Requisitos Funcionais (RF01 a RF02)	23
Quadro 5 – Requisitos Funcionais (RF3 a RF10)	24
Quadro 6 – Requisitos Funcionais (RF11 a RF13)	25
Quadro 7 – Requisitos Não Funcionais (RNF01 a RNF05)	25
Quadro 8 – Histórias de Usuário (US01 a US13)	26
Quadro 9 – Resumo Consolidado Estoque.Domain	37
Quadro 10 – Resumo Consolidado Estoque.Infrastructure	37
Quadro 11 – Resumo Consolidado Estoque.Web	38
Quadro 12 – Resultado SSL Labs	39
Quadro 13 – Resultado Security Headers	40

Lista de tabelas

Tabela 1 – Composição da equipe e funções	18
Tabela 2 – Atribuição de Papéis da Equipe no Kanban	20
Tabela 3 – Detalhamento dos Custos de Desenvolvimento.	47
Tabela 4 – Estrutura Mensal da Empresa.	48

Sumário

1	INTRODUÇÃO	9
1.1	Objetivos	10
1.1.1	Objetivo Geral	10
1.1.2	Objetivos Específicos	10
1.2	Problema e Solução Proposta	11
1.2.1	Problema	11
1.2.2	Solução Proposta	11
1.3	Justificativa	12
1.4	Análise de Concorrência	13
1.4.1	Concorrente 1: Bling ERP	13
1.4.2	Concorrente 2: Tiny ERP	13
1.4.3	Concorrente 3: Nex	14
1.4.4	Concorrente 4: MarketUP	14
1.4.5	Quadro Comparativo	14
2	REVISÃO DA LITERATURA	15
2.1	Histórico do Gerenciamento de Estoque	15
2.2	Atualidades do Gerenciamento de Estoque	16
3	GESTÃO DO PROJETO	18
3.1	Organização da Equipe	18
3.1.1	Responsabilidades/Papéis	18
3.2	Metodologia de Gestão	19
3.2.1	Kanban	19
3.2.2	Funções da Equipe no Kanban	20
3.2.3	Responsabilidades de cada papel	20
3.3	Repositório da Aplicação	20
3.3.1	Definição do Repositório	20
3.3.2	Link e Acessos	21
4	DESENVOLVIMENTO DO PROJETO	22
4.1	Escopo do projeto	22
4.1.1	Regras do Negócio	22
4.1.2	Requisitos Funcionais	23
4.1.3	Requisitos Não Funcionais	25
4.2	Histórias de Usuário	26

4.2.1	Descrição das Histórias	26
4.3	Arquitetura	31
4.3.1	Definições da Arquitetura	31
4.3.2	Diagramas da Arquitetura	32
4.4	Tecnologias	33
4.4.1	Front-end	33
4.4.2	Back-end	34
4.4.3	Banco de Dados	34
4.4.4	Infraestrutura	34
4.5	Estilos de Codificação e Padrões de Projeto	35
4.6	Cobertura de Código e Testes	35
4.6.1	Metas e Escopo da Cobertura	36
4.6.2	Implementação e Validação	36
4.7	Resultados dos Testes	37
4.7.1	Cobertura de Classes	37
4.7.2	Segurança	38
4.7.3	Integração	41
4.7.4	Usabilidade	42
4.8	Segurança, Privacidade, Legislação	42
4.8.1	Critérios de Segurança e Privacidade	42
4.8.2	Legislação	43
4.9	Modelo de Banco de Dados	43
4.10	Cronograma	46
4.10.1	Cronograma de Atividades – Primeiro Semestre	46
5	VIABILIDADE FINANCEIRA	47
5.1	Custos	47
6	CONSIDERAÇÕES FINAIS	49
	APÊNDICE A – MAPEAMENTO DE PROCESSOS	50
A.1	Processo AS IS e TO BE	50
A.1.1	Processo AS IS (Situação Atual)	50
A.1.2	Processo TO BE (Situação Proposta com Sistema)	51
	REFERÊNCIAS	52

1 Introdução

Em um mercado cada vez mais competitivo, a gestão eficiente de recursos tornou-se essencial para o sucesso das empresas, especialmente em lojas de artigos eletrônicos como a VIP PENHA. O setor de eletrônicos tem crescido de forma constante, impulsionado pelo avanço tecnológico e pela alta demanda por dispositivos inteligentes, acessórios e equipamentos de informática. De acordo com dados da [Associação Brasileira da Indústria Elétrica e Eletrônica \(ABINEE\) \(2024\)](#), o mercado brasileiro de eletrônicos movimenta bilhões de reais por ano, sendo um dos segmentos mais dinâmicos e inovadores da economia. Esse crescimento, entretanto, também traz desafios, principalmente para pequenas e médias empresas, que precisam lidar com grande variedade de produtos, constantes atualizações tecnológicas e margens de lucro cada vez mais apertadas.

A transformação digital tem impulsionado mudanças significativas no ambiente corporativo, exigindo das organizações a adoção de práticas tecnológicas que aprimorem seu desempenho. Ferramentas digitais possibilitam operações mais estruturadas, reduzem falhas e embasam decisões estratégicas por meio de dados confiáveis, tornando-se um diferencial competitivo indispensável. Segundo [Laudon e Laudon \(2014\)](#), os sistemas de informação constituem a base para conduzir os negócios na era atual, permitindo às empresas alcançar excelência operacional e desenvolver novos produtos e serviços.

No caso específico da gestão de estoque, a integração tecnológica garante maior visibilidade do fluxo de produtos, minimiza erros operacionais e perdas de informação, além de permitir um planejamento mais eficiente de reposição e armazenamento. O acesso a dados precisos e atualizados fortalece o processo de decisão e amplia a capacidade de expansão da empresa. Para uma loja como a VIP PENHA, que trabalha com produtos de diferentes categorias e faixas de preço, a organização do estoque é um fator determinante para manter a disponibilidade dos itens e evitar prejuízos com produtos parados ou desatualizados.

Além disso, o uso de um sistema de gerenciamento de estoque contribui diretamente para a melhoria do atendimento ao cliente, pois permite prever demandas sazonais, identificar os produtos mais vendidos e antecipar tendências de consumo. Isso possibilita que a empresa mantenha um portfólio de produtos mais alinhado às preferências do público, aumentando sua competitividade no mercado local.

Dessa forma, a digitalização deixou de ser uma opção e passou a representar um pilar estratégico para empresas que buscam eficiência, segurança e competitividade. A implementação de um sistema de gerenciamento de estoque na VIP PENHA alinha-se a essa realidade, contribuindo para a sustentabilidade e o crescimento do negócio diante das

demandas do mercado moderno

1.1 Objetivos

Nesta seção serão apresentados os objetivos do projeto, ele estão divididos em objetivo geral e objetivos específicos. O objetivo geral descreve a meta principal do projeto, já os objetivos específicos descrevem as etapas e funcionalidades para o alcance da meta final.

1.1.1 Objetivo Geral

Auxiliar a gestão da loja VIP Penha por meio do desenvolvimento de um sistema de gerenciamento de estoque para a loja VIP PENHA, visando transformar a gestão de inventário em um processo automatizado e estratégico para otimizar o controle de produtos, diminuir erros manuais e reduzir o tempo gasto na gestão, ao possibilitar o cadastro detalhado de itens, o monitoramento de entradas e saídas em tempo real e a configuração de alertas de estoque mínimo.

1.1.2 Objetivos Específicos

Para alcançar o objetivo geral proposto, torna-se necessário estabelecer metas específicas que orientem o desenvolvimento do sistema de forma organizada e funcional. Esses objetivos buscam garantir que a solução proposta atenda às principais demandas de controle, eficiência e confiabilidade na gestão de estoque da empresa.

- Desenvolver uma plataforma de cadastro de produtos, permitindo o registro e atualização dos produtos no estoque.
- Permitir a análise do desempenho do estoque por meio de relatórios gerenciais, identificando produtos mais vendidos, vendas realizadas e necessidades de reposição.
- Reduzir os erros manuais ao implementar processos automatizados.
- Facilitar a tomada de decisão através de indicadores e gráficos que permitam visualizar o comportamento das vendas e a movimentação dos produtos.
- Garantir a integridade e segurança das informações, assegurando que os dados cadastrados e relatórios gerados sejam armazenados e acessados de forma confiável.
- Proporcionar uma interface intuitiva e acessível, de modo que usuários com diferentes níveis de familiaridade tecnológica consigam operar o sistema sem dificuldades.

1.2 Problema e Solução Proposta

Nessa seção serão apresentados os problemas enfrentados pela loja VIP PENHA em relação ao gerenciamento do seu estoque, assim como a solução proposta para resolver esses problemas. A seguir, são descritos os principais desafios e como o sistema visa solucioná-los de maneira eficiente.

1.2.1 Problema

Devido a recentes expansões, a loja VIP PENHA vem enfrentando problemas como a falta de controle do seu estoque devido a ausência de um sistema automatizado. Essa lacuna tem gerado uma série de complicações que comprometem o funcionamento eficiente da loja. A principal delas é a falta de controle em tempo real das mercadorias, dificultando o acompanhamento preciso da quantidade de produtos disponíveis e resultando em faltas ou excessos de estoque.

Além disso, o processo manual de registro das entradas e saídas de produtos tem sido fonte constante de erros, como registros duplicados, extravios e divergências entre o estoque físico e o registrado. A inexistência de relatórios gerenciais confiáveis impede uma análise clara do desempenho da loja, dificultando a tomada de decisões estratégicas.

Outro ponto preocupante é a falta de rastreabilidade do histórico de movimentações dos produtos, o que inviabiliza a realização de auditorias e a identificação de padrões de consumo. Além disso, a ausência de integração entre os setores de compras, vendas e estoque gera desorganização e atrasos operacionais. Todos esses fatores combinados impactam negativamente a eficiência, o atendimento ao cliente e o potencial de crescimento da loja.

1.2.2 Solução Proposta

Para resolver esses problemas, propõe-se o desenvolvimento de um sistema de gerenciamento de estoque voltado às necessidades específicas da loja VIP PENHA. A implementação desse sistema permitirá centralizar e automatizar os principais processos envolvidos no controle de mercadorias, trazendo mais eficiência, organização e confiabilidade às operações da empresa.

O sistema contará com funcionalidades essenciais como o cadastro de produtos, controle automatizado das entradas e saídas de mercadorias e geração de relatórios gerenciais. Com isso, será possível acompanhar em tempo real o status do estoque, identificar rapidamente produtos em falta ou com excesso, e tomar decisões mais claramente com base nos relatórios fornecidos pelo sistema.

Diferente das soluções genéricas disponíveis no mercado, o sistema VIP PENHA será

totalmente adaptado à realidade e ao fluxo de trabalho da loja, oferecendo uma interface simples, intuitiva e de fácil uso. Além disso, o sistema terá baixo custo de implementação e manutenção, tornando-se uma alternativa acessível e eficiente para pequenos comércios que desejam modernizar sua gestão de estoque sem recorrer a plataformas complexas e de alto custo.

Com a adoção dessa solução, a loja VIP PENHA poderá reduzir drasticamente os erros operacionais, melhorar o fluxo de trabalho, aumentar a produtividade da equipe e oferecer um atendimento mais ágil e eficiente aos clientes. Dessa forma, o sistema contribuirá significativamente para o crescimento sustentável e a consolidação da loja no mercado.

1.3 Justificativa

No ambiente de negócios atual, a gestão de estoque deixou de ser uma tarefa puramente operacional para se tornar um pilar estratégico, essencial para a sobrevivência e competitividade de qualquer empresa varejista. Conforme aponta [Ballou \(2006\)](#) em seus estudos sobre logística empresarial, o controle preciso do inventário impacta diretamente os custos, o nível de serviço ao cliente e, por consequência, a lucratividade do negócio. Ignorar a modernização desta área significa expor a empresa a riscos financeiros e operacionais que podem comprometer sua sustentabilidade.

Para dimensionar a urgência dessa modernização, é crucial analisar o panorama das micro e pequenas empresas no Brasil. Segundo o estudo Mapa de Digitalização das Micro e Pequenas Empresas Brasileiras de 2024, desenvolvido pela Fundação Getúlio Vargas (FGV) em parceria com a [Agência Brasileira de Desenvolvimento Industrial \(ABDI\) \(2024\)](#), o índice médio de maturidade digital dos pequenos negócios é de 35 pontos, em uma escala de 0 a 80, indicando um nível de 43,75% de maturidade média. A pesquisa aponta também que apenas 27% das empresas possuem um sistema de gestão que integra as bases de dados de todas as áreas do negócio .

Nesse cenário, a implementação de um sistema de gestão digital na loja VIP PENHA representa uma ação estratégica, não apenas para acompanhar a modernização do mercado, mas principalmente para solucionar gargalos que hoje prejudicam o desempenho e o crescimento do negócio. Na prática, a implementação deste sistema gera impactos positivos e mensuráveis na operação, destacando-se:

- **Redução de erros operacionais:** O sistema eliminará divergências entre o estoque físico e o digital, garantindo registros precisos e confiáveis.
- **Acompanhamento do sistema:** Com dados digitalizados, é possível acompanhar todas as movimentações de entradas e saídas de produtos com uma extrema facilidade

e confiabilidade.

- **Agilidade nos processos:** As movimentações de produtos serão registradas com mais velocidade e precisão, reduzindo o tempo gasto com conferências e atualizações manuais.
- **Otimização de recursos:** Com um sistema de gerenciamento tecnológico, é possível fazer uma melhor utilização do espaço físico disponível, além de um melhor controle dos recursos financeiros, evitando o excesso e a falta de produtos.
- **Geração de relatórios:** As informações poderão ser analisadas rapidamente por meio de relatórios personalizados, auxiliando na tomada de decisões estratégicas.

Digitalizar o sistema de estoque é, portanto, um passo essencial para empresas que desejam crescer de forma organizada, segura e competitiva. Ao automatizar esse setor, os gestores têm mais controle e previsibilidade, elementos indispensáveis para a sobrevivência e sustentabilidade de qualquer negócio, especialmente no cenário atual de constantes mudanças e demandas cada vez mais dinâmicas.

1.4 Análise de Concorrência

Nesta seção, foi realizada uma análise dos principais sistemas de gerenciamento de estoque disponíveis no mercado, com foco em soluções utilizadas por lojas de pequeno e médio porte. Assim, demonstrando quais as vantagens de usar o sistema que produzimos.

1.4.1 Concorrente 1: Bling ERP

O Bling é um sistema ERP completo que oferece controle de estoque, vendas, emissão de notas fiscais e integração com plataformas de e-commerce. É bastante utilizado por empresas que também vendem online, oferecendo funcionalidades robustas. No entanto, seu uso pode ser complexo para iniciantes, além de exigir pagamento mensal.

1.4.2 Concorrente 2: Tiny ERP

O Tiny ERP oferece funcionalidades similares ao Bling, como controle de estoque, pedidos, emissão de notas fiscais e integração com o setor financeiro. É conhecido por sua interface amigável, mas ainda assim exige uma curva de aprendizado e também é um serviço pago.

1.4.3 Concorrente 3: Nex

O Nex é um sistema gratuito e simples, ideal para pequenos comércios. Permite o cadastro de produtos, controle de estoque e de vendas. É bastante intuitivo, mas possui limitações em relação à integração com outras plataformas e funcionalidades avançadas.

1.4.4 Concorrente 4: MarketUP

O MarketUP é uma solução gratuita e bastante completa, oferecendo controle de estoque, vendas, financeiro e emissão de notas fiscais. No entanto, a interface pode ser confusa, especialmente para usuários menos experientes, e o suporte técnico é limitado.

1.4.5 Quadro Comparativo

Quadro 1 – Comparação entre Sistemas de Gerenciamento de Estoque

Sistema	Funcionalidades Principais	Preço	Observações
Bling ERP	Controle de estoque, vendas, emissão de notas fiscais, integração com e-commerce	Pago	Funcional, mas complexo para iniciantes
Tiny ERP	Estoque, pedidos, notas fiscais, controle financeiro	Pago	Interface moderna, porém exige curva de aprendizado
Nex	Cadastro de produtos, estoque e vendas	Gratuito	Intuitivo, ideal para pequenos comércios, porém limitado
MarketUP	Estoque, vendas, financeiro, notas fiscais	Gratuito	Completo, mas com interface confusa e suporte limitado
"Sistema VIP PENHA"	Controle de estoque em tempo real, alerta de estoque mínimo, cadastro técnico de produtos, controle de garantias e geração de relatórios personalizados	Gratuito/Personalizado	Sistema desenvolvido especificamente para a realidade da loja VIP PENHA, com interface simples e fácil aprendizado, garantindo maior eficiência e redução de erros.

2 Revisão da Literatura

O presente capítulo tem como objetivo apresentar estudos, teorias e contribuições acadêmicas relacionados à gestão de estoque em empresas.

2.1 Histórico do Gerenciamento de Estoque

A gestão de estoque é uma prática que acompanha a humanidade há milênios, realizar uma armazenagem inteligente dos recursos se mostrou essencial para a raça humana desde o seu primórdio. Um exemplo é o Período Uruk, nele foram desenvolvidas várias técnicas de gestão, como o uso de imagens e símbolos para administrar a estocagem de grãos, frutas e produtos, o que impulsionou essa sociedade a grandes avanços e ao desenvolvimento de um dos primeiros sistemas de escrita da história ([CONFERENCE, 1996](#)).

Durante a revolução industrial, com a produção em larga escala, houve um grande aumento na necessidade de melhores práticas de gerenciamento de estoque. O aumento da demanda de abastecimento contínuo do mercado levou ao desenvolvimento de melhores técnicas de controle e armazenamento dos produtos ([DIAS; OUTROS, 2021](#)).

Em meados do século XX, com o avanço da computação e o surgimento de sistemas informatizados, ocorreu uma verdadeira revolução na gestão de estoque. O desenvolvimento de softwares específicos para administração de materiais, como o Material Requirements Planning (MRP), criado na década de 1960, permitiu que empresas passassem a planejar suas necessidades de produção com base em dados de consumo e previsão de demanda. Esse modelo evoluiu posteriormente para o Manufacturing Resource Planning (MRP II), que integrou outras áreas da empresa, como produção, compras e finanças, ampliando o controle sobre os processos internos ([SLACK; CHAMBERS; JOHNSTON, 2013](#)).

Já nos anos 1990, surge o Enterprise Resource Planning (ERP), sistema que consolidou os conceitos anteriores e os ampliou para toda a organização. O ERP unificou informações de diferentes setores em uma única plataforma, permitindo maior integração entre compras, estoque, vendas e contabilidade. Essa inovação marcou uma nova era na administração de estoques, com base em dados, indicadores e processos informatizados. Paralelamente, práticas como o Just in Time (JIT), desenvolvidas inicialmente no Japão pela Toyota, também influenciaram o modo como as empresas passaram a lidar com seus estoques, priorizando a redução de desperdícios e a reposição conforme a necessidade real de produção ([OHNO, 1997](#)).

Nas décadas seguintes, com o avanço da globalização e da tecnologia da informação,

a gestão de estoques passou a incorporar conceitos de cadeia de suprimentos (supply chain management), conectando empresas, fornecedores e clientes em tempo real. As ferramentas digitais se tornaram indispensáveis para lidar com grandes volumes de dados, prever demandas e otimizar o fluxo de mercadorias.

Dessa forma, o percurso histórico ressalta a complexidade e a necessidade de soluções de gestão de estoque modernas, capazes de integrar processos e informações de maneira eficiente. A evolução dos métodos, desde o controle manual até os sistemas ERP, demonstra que a tecnologia tornou-se essencial para a administração eficaz dos estoques. No entanto, também evidencia que muitas das soluções existentes no mercado não são adequadas à realidade de pequenos comércios, por demandarem altos investimentos e infraestrutura avançada.

Sendo assim, o desenvolvimento do sistema proposto para a VIP PENHA se apoia nesse entendimento histórico e tecnológico: trata-se de uma solução digital sob medida, que busca aliar os princípios da gestão moderna, precisão, integração e às necessidades práticas de um negócio local.

2.2 Atualidades do Gerenciamento de Estoque

Atualmente, o processo de gerenciamento de estoque encontra-se em níveis elevados de integração tecnológica, com diversas ferramentas modernas desempenhando papéis fundamentais nessa tarefa. A utilização de sistemas automatizados, softwares de gestão integrada (ERP) e tecnologias de rastreamento tem sido amplamente adotada pelas empresas com o objetivo de aumentar o controle, a acuracidade e a eficiência do estoque. Sistemas de gestão empresarial, como ERPs, permitem centralizar e padronizar os dados relacionados às operações, conectando departamentos como compras, vendas e logística, o que resulta em atualizações em tempo real e redução de erros humanos ([Cast4IT, 2024](#)).

Os softwares de gestão integrada consolidam dados operacionais e financeiros em uma única plataforma, permitindo que decisões estratégicas sejam tomadas com base em informações confiáveis e atualizadas. A automação dos processos, aliada à padronização e integração promovidas pelos sistemas ERP, contribui para a redução de desperdícios, otimização de recursos e aumento da produtividade das equipes. Dessa forma, empresas que aplicam essas soluções conseguem alinhar o gerenciamento de estoque às necessidades reais do mercado, ajustando os níveis de produtos de maneira mais eficiente e estratégica ([SILVA et al., 2020](#)).

Dessa forma, a gestão de estoque moderna deixa de ser apenas uma função operacional e passa a desempenhar um papel estratégico dentro das organizações. A adoção de tecnologias avançadas permite não apenas maior precisão e controle, mas também fortalece a capacidade competitiva das empresas, tornando o gerenciamento de estoque

um elemento essencial para o sucesso e a sustentabilidade dos negócios.

3 Gestão do Projeto

Este capítulo apresenta como a equipe foi estruturada, quais papéis e responsabilidades cada membro assumiu, além da metodologia de gerenciamento adotada ao longo do desenvolvimento. Também são descritas as ferramentas utilizadas para organização das tarefas e o repositório da aplicação.

3.1 Organização da Equipe

A equipe do projeto foi composta por membros com múltiplas atribuições, garantindo flexibilidade e colaboração entre as áreas de desenvolvimento, testes e gestão. Cada integrante desempenhou funções combinadas de análise, desenvolvimento e suporte, de modo a cobrir todas as etapas do projeto sem sobrecarregar nenhum membro

Essa abordagem favoreceu a comunicação, a integração das atividades e a eficiência do trabalho, permitindo identificar problemas rapidamente e otimizar recursos. Além disso, a equipe teve papel ativo na orientação de funcionários da VIP PENHA quanto ao uso do sistema, garantindo que as novas práticas de gestão de estoque fossem incorporadas de forma eficaz.

Membro	Função(ões)
Vitor	Gestor, DBA
Matheus	Front-end, QA
Beatriz	Front-end, Back-end
Rafael	Front-end, Back-end
Khalil	Back-end, QA
Gean	QA
Marcelo	QA

Tabela 1 – Composição da equipe e funções

3.1.1 Responsabilidades/Papéis

Os papéis desempenhados pela equipe foram definidos com base em suas competências e nas necessidades do projeto. A seguir, os principais papéis e suas responsabilidades:

- **Gestor:** Responsável pelo planejamento, organização, acompanhamento das atividades, definição de cronogramas e mediação da comunicação interna e externa.

- **DBA (Administrador de Banco de Dados):** Responsável pela modelagem do banco de dados, criação e manutenção das estruturas de dados, performance e integridade das informações.
- **Desenvolvedor Back-end:** Responsável pela implementação da lógica de negócio, criação de APIs, segurança e integração com o banco de dados.
- **Desenvolvedor Front-end:** Responsável pela criação da interface visual da aplicação, usabilidade, responsividade e interação com o usuário.
- **QA (Quality Assurance):** Responsável pela garantia da qualidade da aplicação, planejamento e execução de testes funcionais, validação dos requisitos e registro de bugs.

3.2 Metodologia de Gestão

3.2.1 Kanban

A equipe optou pela utilização do método **Kanban** para o gerenciamento das atividades do projeto. Essa metodologia permite o acompanhamento visual focando na entrega contínua e em tempo real das tarefas.

O quadro Kanban utilizado possui as seguintes colunas:

- **Backlog – Código:** Armazena ideias e funcionalidades relacionadas à implementação do código que ainda não foram iniciadas.
- **Backlog – Documentação:** Armazena tarefas de documentação que ainda não foram iniciadas.
- **Design:** Etapa dedicada à elaboração de pesquisas.
- **A Fazer:** Tarefas já priorizadas e planejadas, aguardando início.
- **Em Andamento:** Tarefas em desenvolvimento.
- **Revisão de Código:** Etapa de verificação e revisão do código antes da finalização.
- **Fase de Teste:** Validação e testes das funcionalidades desenvolvidas.
- **Concluído:** Tarefas finalizadas, revisadas e testadas com sucesso.

As atividades são constantemente avaliadas e realocadas entre as colunas conforme seu progresso, promovendo transparência e melhoria contínua do fluxo de trabalho.

3.2.2 Funções da Equipe no Kanban

A seguir, serão apresentadas as funções da equipe no projeto, com os respectivos membros atribuídos e as responsabilidades de cada papel:

PO	Flow manager	Team Member
Vitor	Beatriz	Matheus
		Khalil
		Gean
		Marcelo
		Rafael

Tabela 2 – Atribuição de Papéis da Equipe no Kanban

3.2.3 Responsabilidades de cada papel

- **PO (Product Owner):** Responsável por representar os interesses do cliente e das partes interessadas. No contexto Kanban, o PO prioriza as tarefas no backlog e garante que o trabalho mais valioso seja entregue primeiro, alinhando as entregas com os objetivos do projeto.
- **Flow Manager:** Responsável por acompanhar e otimizar o fluxo de trabalho da equipe. Atua identificando gargalos, promovendo a melhoria contínua, monitorando métricas como lead time e WIP, além de incentivar a transparência, colaboração e boas práticas no uso do Kanban.
- **Team Members:** São os membros da equipe de desenvolvimento que executam o trabalho técnico. Isso inclui análise, implementação, testes e revisão de código. Eles colaboram continuamente para manter o fluxo de trabalho saudável e entregar valor com qualidade.

3.3 Repositório da Aplicação

Nesta seção, definimos o repositório da aplicação, assim como seus links e necessidades para acesso.

3.3.1 Definição do Repositório

O repositório escolhido foi o Github, usado para armazenar, versionar e compartilhar o código-fonte do projeto. Assim, facilitando a colaboração entre os membros da equipe. Este repositório é público, o que significa que qualquer pessoa com o link pode visualizá-lo, navegar pelo código-fonte e acompanhar o histórico de alterações sem a necessidade de autenticação.

3.3.2 Link e Acessos

O repositório está hospedado no GitHub e pode ser acessado pelo seguinte link:



Figura 1 – QR Code para o repositório no GitHub

<https://github.com/VitorDaSilvaOliveira/Projeto-Integrado-IFSP>

4 Desenvolvimento do Projeto

Esta seção detalha o desenvolvimento do projeto, abrangendo desde a definição do escopo, regras de negócio e requisitos até as tecnologias utilizadas e a arquitetura adotada. O objetivo é descrever de forma clara como o sistema foi concebido, implementado e validado. Fases do desenvolvimento do projeto

4.1 Escopo do projeto

O projeto tem como objetivo o desenvolvimento de um sistema de controle de estoque voltado para um estabelecimento comercial. O sistema será acessado via navegador e terá como foco a organização e o gerenciamento de produtos, fornecedores e movimentações de entrada e saída de estoque.

4.1.1 Regras do Negócio

Quadro 2 – Regras de Negócio (RN01 a RN04)

Código	Nome	Descrição	Requisito Relacionado
RN01	Cadastro de Produtos	Produtos devem ter: nome, código, categoria, preço, garantia e estoque mínimo obrigatoriamente.	RF01
RN02	Atualização de Estoque	Qualquer entrada/saída deve atualizar automaticamente o estoque e recalcular o valor total em estoque. Produtos abaixo do estoque mínimo devem gerar notificações no sistema.	RF01, RF05
RN03	Registro de Movimentações	Toda movimentação deve gerar um registro contendo: tipo, data, responsável, produto(s), quantidade.	RF04
RN04	Controle de Acesso	O sistema deve ter diferentes níveis de acesso para perfis de Administrador e outros tipos de perfil.	RF07, RF08, RF11

Quadro 3 – Regras de Negócio (RN05 a RN09)

Código	Nome	Descrição	Requisito Relacionado
RN05	Garantia	Devoluções só podem ser aceitas caso os itens ainda estejam dentro do prazo de garantia correspondente.	RF02, RF06
RN06	Validação de Venda	Vendas só podem ser registradas se houver estoque suficiente para todos os itens.	RF01, RF02
RN07	Informações da Venda	Toda venda deve ter informações relevantes como valor, desconto, forma de pagamento e dados do cliente, registrada nos dados do pedido, e na nota fiscal eletrônica.	RF02, RF12, RF13
RN08	Auditoria	Todas as exclusões e alterações de preço/estoque devem registrar IP, usuário, data/hora e valores antes/depois.	RF07
RN09	Criação de Relatórios	O sistema deve gerar relatórios com informações relevantes sobre produtos, pedidos, clientes, correspondendo ao estado atual do estoque, e esses relatórios devem poder ser exportados em formato adequado.	RF05

4.1.2 Requisitos Funcionais

Quadro 4 – Requisitos Funcionais (RF01 a RF02)

Código	Atores	Nome	Descrição
RF01	Gerente de estoque	Controle de Produtos	O sistema deve permitir ao usuário cadastrar, alterar, consultar e deletar produtos. Deve registrar movimentações de estoque e o histórico de alterações, garantindo que produtos tenham categoria e quantidade mínima ideal.
RF02	Gerente de estoque, Vendedor/Funcionário	Controle de Pedidos	O sistema deve permitir cadastrar, consultar e alterar pedidos, vinculando produtos e quantidades. Pedidos aceitos geram movimentações de estoque, reduzindo saldos. Pedidos podem ter alteração de estado, mas não podem ser deletados.

Quadro 5 – Requisitos Funcionais (RF3 a RF10)

Código	Atores	Nome	Descrição
RF03	Gerente de estoque	Controle de Fornecedores	O sistema deve permitir o cadastro de fornecedores. Após criado, o registro não pode ser deletado, apenas inativado, e deve ser relacionado a devoluções e movimentações.
RF04	Gerente de estoque	Controle de Movimentação	O sistema deve gerar movimentações a partir de pedidos e devoluções, alterando a quantidade em estoque. O registro é de segurança e não pode ser deletado após a criação.
RF05	Gerente de estoque	Relatórios de Estoque	O sistema deve gerar relatórios consolidados para auxiliar a tomada de decisão, identificando dados interessantes referentes a produtos, pedidos, clientes, fornecedores, movimentações e os relacionando de forma útil para a análise do negócio.
RF06	Vendedor, Gerente de estoque	Registrar Devoluções	O sistema deve permitir o registro de devoluções de produtos (de cliente ou para fornecedor), vinculadas a pedidos e dentro da garantia (para clientes). Deve registrar data, origem, justificativa e impactar relatórios e movimentações.
RF07	Administrador do sistema	Gestão de Usuários e Permissões	O sistema deve ter gestão de perfis (Admin/Gerente e Vendedor/Funcionário) com permissões distintas. Ações críticas (como deletar produtos) devem ser restritas ao Administrador e registradas em log.
RF08	Administrador do sistema, Usuário do sistema	Cadastro e Autenticação	O sistema deve permitir o cadastro e login com e-mail e senha, aplicando controle de acesso. As senhas devem ser armazenadas de forma segura (criptografia/hashing).
RF09	Usuário do sistema	Redefinição de Senha	O sistema deve permitir que usuários solicitem a redefinição de senha via e-mail. Deve enviar um link seguro e temporário, exigindo critérios mínimos de segurança para a nova senha.
RF10	Todos os usuários do sistema	Internacionalização	O sistema deve oferecer suporte a internacionalização, estando disponível em Português e Inglês. O idioma deve ser configurável pelo usuário ou ajustado pelo navegador.

Quadro 6 – Requisitos Funcionais (RF11 a RF13)

Código	Atores	Nome	Descrição
RF11	Administrador, Vendedor	Suporte a Múltiplas Lojas	O sistema deve permitir a gestão de estoque em múltiplas lojas. Usuários devem ser vinculados a uma loja, e o Administrador deve poder alternar entre elas.
RF12	Administrador, Vendedor	Controle de Clientes	O sistema deve permitir o gerenciamento e registro completo de clientes. Todo cliente registrado deve ter pedidos atrelados a ele.
RF13	Administrador, Vendedor	Geração de Nota Fiscal	O sistema deve, após a finalização de um pedido, gerar a Nota Fiscal eletrônica correspondente à venda, validando os dados fiscais do cliente e do produto, e registrando o status da emissão..

4.1.3 Requisitos Não Funcionais

Quadro 7 – Requisitos Não Funcionais (RNF01 a RNF05)

Código	Módulo	Descrição
RNF01	Desempenho	O sistema deve suportar no mínimo 10 usuários simultâneos sem queda de desempenho.
RNF02	Segurança	O sistema deve ter todas as suas rotas protegidas por um mecanismo de autenticação robusto.
RNF03	Disponibilidade	O sistema deve estar completamente disponível a todo momento, com uma tolerância máxima de queda de 0,15 ao mês.
RNF04	Multiusuário	O sistema deve ser operável por múltiplos usuários ao mesmo tempo sem que ocorram inconsistências nos dados transacionais.
RNF05	Qualidade de Código	O sistema deve poder receber manutenção de código com facilidade, seguindo padrões de desenvolvimento e possuindo documentação extensiva.
RNF06	Usabilidade	O sistema deve ter uma interface intuitiva e uma curva de aprendizagem rápida, permitindo que um novo usuário realize as principais operações com pouco treinamento.
RNF07	Tolerância a Falhas	O sistema deve possuir um mecanismo de registro (logs) de erros e falhas, que permita a rastreabilidade e identificação da causa raiz em um tempo curto.

4.2 Histórias de Usuário

Quadro 8 – Histórias de Usuário (US01 a US13)

Código	História de Usuário	Requisito Funcional Relacionado
US01	Cadastrar, Consultar, Alterar e Deletar Produtos.	RF01
US02	Cadastrar e consultar pedidos.	RF02
US03	Gerenciar fornecedores.	RF03
US04	Registrar movimentações de estoque.	RF04
US05	Gerar relatórios consolidados de estoque.	RF05
US06	Registrar devoluções.	RF06
US07	Gerenciar usuários e permissões.	RF07
US08	Cadastro e autenticação segura.	RF08
US09	Redefinição de senha segura.	RF09
US10	Configurar o idioma do sistema.	RF10
US11	Suporte a múltiplas lojas.	RF11
US12	Gerenciar e registrar clientes.	RF12
US13	Gerar a Nota Fiscal eletrônica.	RF13

Fonte: Elaborado com base nos Requisitos Funcionais (RF01 a RF12)

4.2.1 Descrição das Histórias

Neste tópico será detalhada a descrição das Histórias de Usuário, incluindo o ator principal, a descrição e os critérios de aceitação.

1. US01: Cadastrar, Consultar, Alterar e Deletar Produtos

Descrição: Como um **Gerente de Estoque**, eu quero ter a capacidade de gerenciar o ciclo de vida dos produtos no sistema (cadastrar, consultar, alterar e deletar), para que o catálogo de estoque esteja sempre atualizado e com a configuração da quantidade mínima ideal.

Requisito Funcional Relacionado: RF01 – Controle de Produtos

Critérios de Aceitação:

- Deve existir uma interface acessível para o gerenciamento de produtos.
- O sistema deve permitir **registrar movimentações de estoque** e o **histórico de alterações** do produto.
- Deve ser obrigatório vincular o produto a uma **categoria** e definir a **quantidade mínima ideal**.
- O processo de exclusão deve garantir que o produto não tenha vínculos ativos.

2. US02: Cadastrar e Consultar Pedidos

Descrição: Como **Vendedor**, eu quero cadastrar, consultar e alterar pedidos, vinculando produtos e quantidades, para iniciar o processo de venda e garantir que o estoque seja reduzido após a aceitação do pedido.

Requisito Funcional Relacionado: RF02 – Controle de Pedidos

Critérios de Aceitação:

- O sistema deve permitir vincular o pedido a um cliente (RF12) e a múltiplos produtos com suas quantidades.
- A **aceitação de um pedido** deve gerar automaticamente uma **movimentação de saída** no estoque (RF04).
- Pedidos podem ser **aceitos ou recusados**, mas **não devem ser deletados** para manter a rastreabilidade.

3. US03: Gerenciar Fornecedores

Descrição: Como **Gerente de Estoque**, eu quero cadastrar fornecedores com suas informações (nome, documento, localização), permitindo apenas a inativação, para que o histórico de devoluções e movimentações seja sempre rastreável.

Requisito Funcional Relacionado: RF03 – Controle de Fornecedores

Critérios de Aceitação:

- O sistema deve permitir o cadastro de dados como CNPJ/Documento e Contatos.
- O fornecedor deve ser **relacionado a devoluções e movimentações** de entrada de produtos.
- Após a criação, o registro do fornecedor **não pode ser deletado**, apenas **inativado**.

4. US04: Registrar Movimentações de Estoque

Descrição: Como **Gerente de Estoque**, eu quero registrar movimentações de entrada ou saída (produto, quantidade, tipo, usuário), para que a quantidade em estoque seja atualizada de forma precisa.

Requisito Funcional Relacionado: RF04 – Controle de Movimentação

Critérios de Aceitação:

- O sistema deve permitir registrar o **tipo de movimentação** (Ex: Entrada por Compra, Saída por Venda, Perda).
- A quantidade em estoque deve ser alterada automaticamente após o registro.

- O registro de movimentação é de **segurança** e **não pode ser deletado** após a criação.

5. US05: Gerar Relatórios Consolidados de Estoque

Descrição: Como **Gerente de Estoque**, eu quero gerar relatórios consolidados, com filtros de loja, para identificar produtos abaixo do mínimo ideal, perdas e analisar o saldo total considerando todas as movimentações.

Requisito Funcional Relacionado: RF05 – Relatórios de Estoque

Critérios de Aceitação:

- O relatório deve ser capaz de **identificar produtos abaixo do mínimo ideal** (RF01).
- Deve incluir informações sobre perdas, entradas, saídas e devoluções.
- O relatório deve permitir filtro por **loja** (RF11).

6. US06: Registrar Devoluções

Descrição: Como **Vendedor ou Gerente de Estoque**, eu quero registrar devoluções (de cliente ou para fornecedor), vinculadas a pedidos, para que o estoque e os relatórios sejam corretamente impactados.

Requisito Funcional Relacionado: RF06 – Registrar Devoluções

Critérios de Aceitação:

- Deve ser obrigatório registrar a **origem, justificativa** e a **data da devolução**.
- Devoluções de cliente devem ser validadas se estão **dentro do período de garantia**.
- O registro deve gerar uma **movimentação de entrada** no estoque (RF04) ou de saída para o fornecedor.

7. US07: Gerenciar Usuários e Permissões

Descrição: Como **Administrador do Sistema**, eu quero ter gestão de perfis (Admin/Gerente e Vendedor/Funcionário) com permissões distintas, para que ações críticas sejam restritas e registradas em log.

Requisito Funcional Relacionado: RF07 – Gestão de Usuários e Permissões

Critérios de Aceitação:

- O sistema deve garantir que ações críticas (ex: exclusão de registros) sejam **restritas ao perfil Administrador**.
- Todas as alterações de permissão devem ser **registradas em log** de auditoria.

- A interface deve ser ajustada para exibir apenas as funcionalidades permitidas ao perfil logado.

8. US08: Cadastro e Autenticação Segura

Descrição: Como **Usuário do Sistema**, eu quero poder me cadastrar e efetuar o login usando e-mail e senha, para ter acesso controlado e seguro ao sistema.

Requisito Funcional Relacionado: RF08 – Cadastro e Autenticação

Critérios de Aceitação:

- O sistema deve aplicar **controle de acesso** e redirecionar para a tela inicial após o login.
- As senhas devem ser armazenadas de forma segura (utilizando **criptografia/hashing**).
- Deve ser verificada a existência do usuário para evitar duplicidade.

9. US09: Redefinição de Senha Segura

Descrição: Como **Usuário do Sistema**, eu quero solicitar a redefinição de senha, recebendo um link seguro e temporário via e-mail, para recuperar meu acesso quando necessário.

Requisito Funcional Relacionado: RF09 – Redefinição de Senha

Critérios de Aceitação:

- O link de redefinição enviado por e-mail deve ter **validade limitada** (temporário).
- O sistema deve exigir **critérios mínimos de segurança** (ex: tamanho, caracteres especiais) para a nova senha.
- Deve haver uma confirmação na tela após a solicitação e após a redefinição bem-sucedida.

10. US10: Configurar o Idioma do Sistema

Descrição: Como **Qualquer Usuário do Sistema**, eu quero que o sistema ofereça suporte a internacionalização, estando disponível em, no mínimo, Português e Inglês, para configurar a interface no meu idioma.

Requisito Funcional Relacionado: RF10 – Internacionalização

Critérios de Aceitação:

- Deve haver uma opção para o usuário **configurar o idioma preferencial** no seu perfil.

- Alternativamente, o sistema deve **ajustar o idioma** com base na configuração do navegador.
- Todos os textos da interface de usuário devem estar traduzidos para os idiomas suportados.

11. US11: Suporte a Múltiplas Lojas

Descrição: Como **Administrador**, eu quero que o sistema permita a gestão de estoque em, no mínimo, duas lojas distintas, para que os usuários sejam vinculados a uma loja e eu possa alternar entre elas.

Requisito Funcional Relacionado: RF11 – Suporte a Múltiplas Lojas

Critérios de Aceitação:

- Deve ser possível **vincular usuários a uma loja específica** (exceto o Administrador).
- O Administrador deve ter uma interface para **alternar o contexto** entre as lojas.
- Todas as consultas e relatórios de estoque (RF05) e movimentações (RF04) devem respeitar o contexto da loja ativa.

12. US12: Gerenciar e Registrar Clientes

Descrição: Como **Vendedor**, eu quero gerenciar o cadastro completo de clientes (nome, documento, contatos), para que todo pedido de venda esteja atrelado a um cliente registrado.

Requisito Funcional Relacionado: RF12 – Controle de Clientes

Critérios de Aceitação:

- O sistema deve permitir o cadastro de dados como CPF/CNPJ, e-mail e telefone.
- O sistema **deve exigir um cliente registrado** para a criação de um novo pedido (RF02).
- Deve ser possível consultar o histórico de pedidos e devoluções (RF06) atrelado a cada cliente.

13. US13: Gerar a Nota Fiscal Eletrônica

Descrição: Como **Vendedor**, eu quero que o sistema **gere a Nota Fiscal eletrônica** automaticamente após a aceitação ou finalização de um pedido (RF02), para cumprir as obrigações fiscais e disponibilizar o documento ao cliente.

Requisito Funcional Relacionado: RF13 – Geração de Nota Fiscal

Critérios de Aceitação:

- A geração da NF-e deve ser disparada após a **confirmação de que o pedido foi aceito e o pagamento efetuado**.
- O sistema deve **validar os dados fiscais** antes da emissão.
- Deve ser registrado o **status da emissão** e a chave de acesso da NF-e.
- O sistema deve **enviar a NF-e (XML e DANFE)** ao e-mail do cliente automaticamente.

4.3 Arquitetura

A arquitetura do sistema representa a estrutura organizacional fundamental da aplicação, incluindo seus principais componentes, suas relações, ambientes de execução e os padrões adotados para garantir a escalabilidade, manutenibilidade e segurança. Esta seção descreve a forma como o sistema foi planejado e dividido, visando facilitar tanto o desenvolvimento quanto a futura evolução da solução. São considerados aqui os aspectos lógicos e físicos do sistema, incluindo frameworks, tecnologias e infraestrutura.

4.3.1 Definições da Arquitetura

A arquitetura do projeto foi desenvolvida como uma aplicação Web, apoiando-se em tecnologias modernas para garantir robustez, escalabilidade e segurança. O sistema foi concebido com uma abordagem em camadas, utilizando principalmente tecnologias da plataforma .NET.

No lado do servidor, foi adotada a linguagem C# com o framework ASP.NET MVC, facilitando a separação das responsabilidades e promovendo uma melhor organização do código através do padrão Model-View-Controller (MVC).

Para o acesso a dados, foi utilizado Entity Framework com LINQ, que permite uma comunicação eficiente com o banco de dados relacional e facilita a execução de consultas através de expressões C#. A base de dados principal é gerida por Microsoft SQL Server, garantindo a persistência e segurança das informações.

No front-end, a aplicação utiliza Razor Views (.cshtml), combinadas com HTML, CSS e JavaScript, para renderização dinâmica das páginas diretamente no servidor. O uso de Bootstrap assegura uma interface responsiva e compatível com diferentes dispositivos. Para manipulação do DOM e criação de interações dinâmicas, foi utilizado o jQuery, ampliando a usabilidade da aplicação.

Para aprimorar a interação do usuário, foi utilizada a biblioteca jQuery, que facilita o tratamento de eventos e a manipulação do DOM, permitindo a criação de funcionalidades dinâmicas e responsivas.

A autenticação e autorização dos usuários são tratadas por meio do ASP.NET Identity, que integra recursos robustos de controle de acesso, gerenciamento de permissões e segurança na autenticação. Esse serviço está destacado no diagrama como um componente separado e essencial para a integridade e confiabilidade do sistema.

Além disso, o sistema utiliza o framework JJMasterData, uma solução que permite a criação dinâmica de formulários e estruturas de dados em tempo de execução. Com ele, é possível configurar cadastros e regras de negócio diretamente pela interface, sem necessidade de mudanças no código, trazendo mais flexibilidade e agilidade na evolução do sistema.

4.3.2 Diagramas da Arquitetura

Foram construídos diagramas de Componentes e de Implantação para visualizar a estrutura do projeto. O diagrama de componentes descreve a estrutura modular da aplicação, evidenciando os elementos lógicos e suas interações internas. Já o diagrama de implantação representa a distribuição física desses componentes em ambientes de execução, detalhando a infraestrutura utilizada e as conexões entre os ambientes.

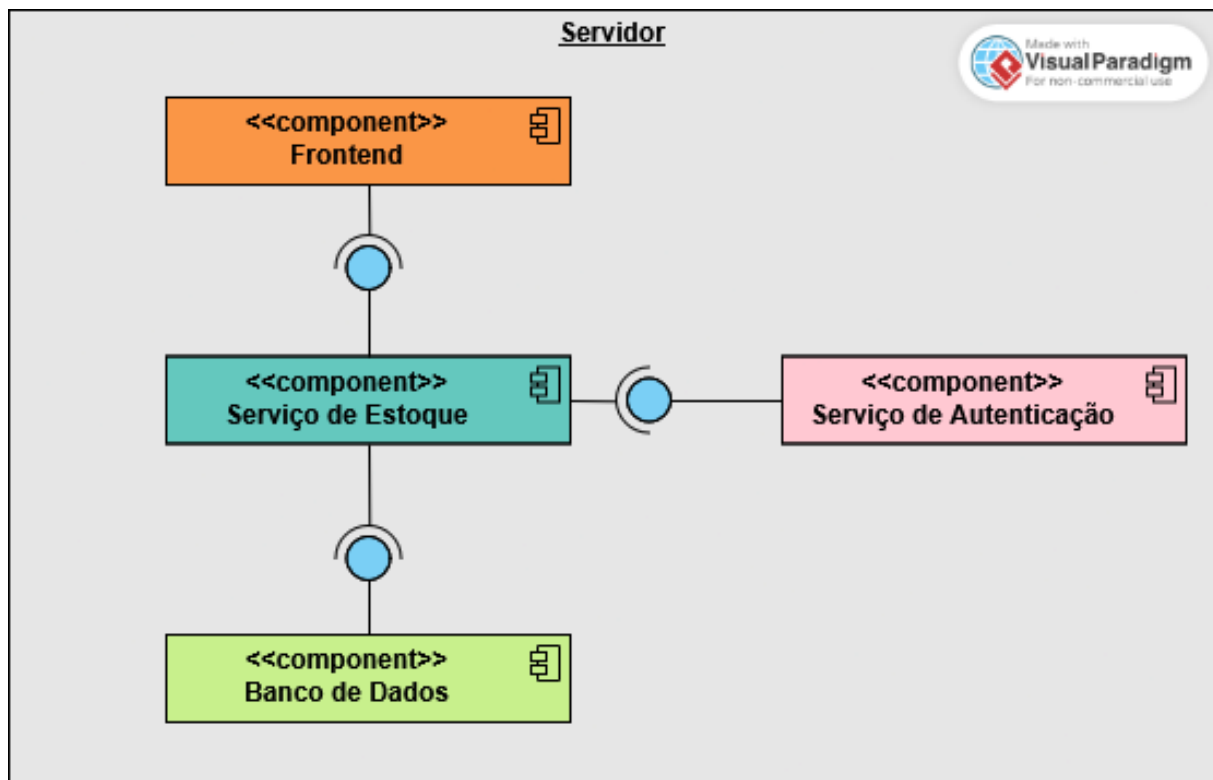


Figura 2 – Diagrama de Componentes

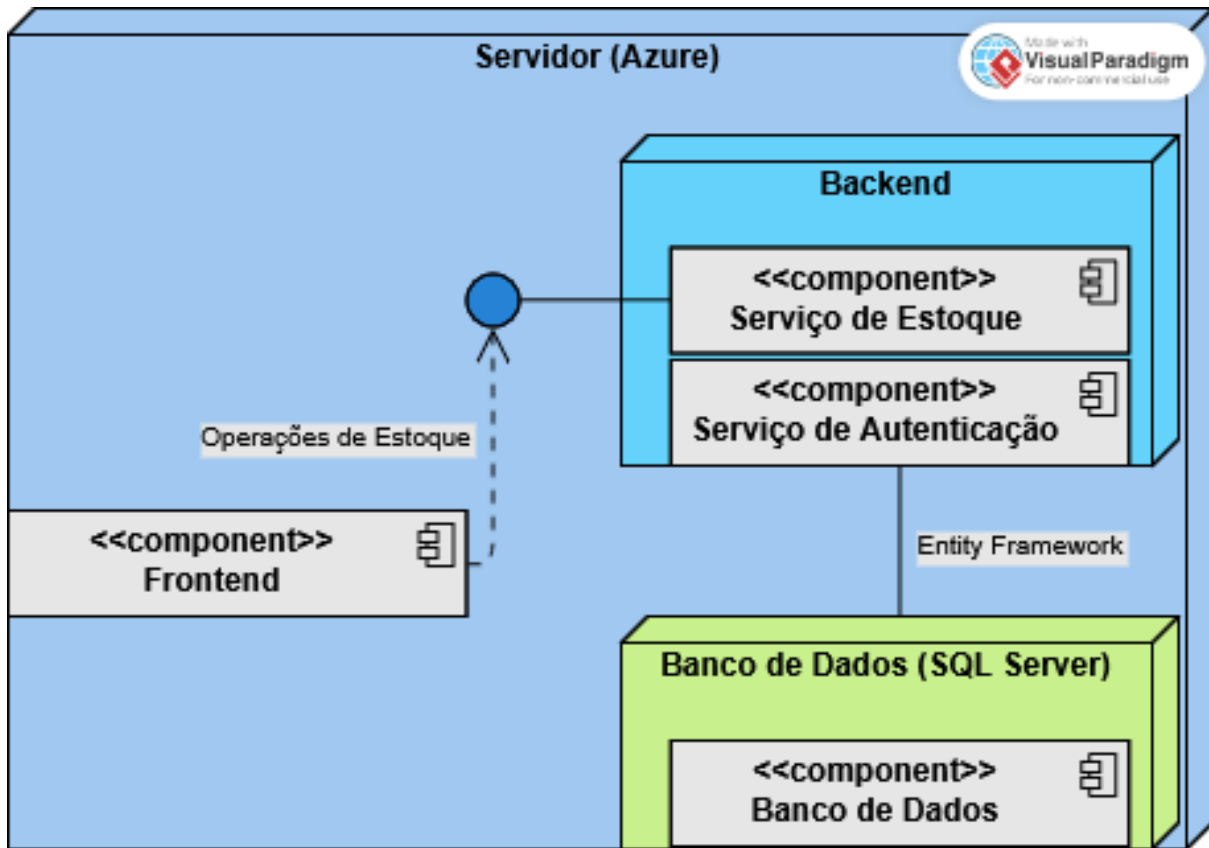


Figura 3 – Diagrama de Implantação

4.4 Tecnologias

Esta seção detalha as tecnologias escolhidas para o desenvolvimento do sistema, organizadas em categorias. A seleção foi baseada em critérios de desempenho, escalabilidade e compatibilidade com os requisitos do projeto.

4.4.1 Front-end

- **Razor Views:** Arquivos .cshtml que utilizam a engine Razor do ASP.NET para gerar páginas HTML dinâmicas. Essa abordagem permite mesclar código C# com marcação HTML de forma fluida, facilitando a renderização de conteúdo no lado do servidor e a reutilização de componentes visuais (Microsoft, 2024a).
- **HTML, CSS e JavaScript:** Tecnologias fundamentais para a construção da camada de apresentação do sistema. O HTML estrutura o conteúdo das páginas, o CSS define o estilo visual e o JavaScript adiciona interatividade ao front-end (Alura, 2024).
- **Bootstrap:** Framework front-end baseado em HTML, CSS e JS que oferece componentes prontos e responsivos, facilitando a criação de interfaces modernas, padronizadas e adaptáveis a diferentes dispositivos (Bootstrap, 2024).

- **jQuery:** Biblioteca JavaScript que simplifica a manipulação do DOM, o tratamento de eventos e requisições AJAX. Foi utilizada para agilizar o desenvolvimento de funcionalidades interativas no front-end da aplicação (jQuery, 2025).

4.4.2 Back-end

- **C# com ASP.NET MVC:** Linguagem e framework utilizados na construção do back-end da aplicação, seguindo o padrão arquitetural Model-View-Controller (MVC), que organiza o código de forma modular, separando lógica de negócios, visualização e controle (Microsoft, 2025b).
- **Entity Framework com LINQ:** Conjunto de tecnologias para acesso a dados em .NET. O Entity Framework permite o mapeamento objeto-relacional (ORM), e o LINQ facilita a realização de consultas ao banco de dados de forma legível e integrada ao C# (Microsoft, 2024b).
- **ASP.NET Identity:** Sistema de autenticação e controle de acesso da Microsoft utilizado para gerenciar usuários, perfis e permissões de forma segura e integrada ao projeto (Microsoft, 2025a).
- **JJMasterData:** Ferramenta de administração e modelagem de dados que permite a criação dinâmica de formulários e telas de cadastro com base nas entidades do sistema, otimizando o desenvolvimento da interface administrativa (JJConsulting, 2025).

4.4.3 Banco de Dados

- **Microsoft SQL Server:** O Microsoft SQL Server é um sistema de gerenciamento de banco de dados relacional (RDBMS) robusto e amplamente utilizado no mercado, responsável pelo armazenamento seguro das informações da aplicação (Microsoft, 2025d).

4.4.4 Infraestrutura

- **Microsoft Azure:** Plataforma de computação em nuvem utilizada para hospedar a aplicação e seus serviços relacionados. A utilização do Azure proporciona escalabilidade, segurança e alta disponibilidade (Microsoft, 2025c). O Azure está sendo utilizado como plataforma principal para a hospedagem de todos os componentes do sistema, abrangendo o banco de dados, o back-end e o front-end.

4.5 Estilos de Codificação e Padrões de Projeto

Para manter a consistência e legibilidade do código-fonte, foram adotados os seguintes padrões:

- **C# (Back-end) - Padrões .NET:** Utilização das diretrizes oficiais do Microsoft .NET, que incluem:
 - **Convenções de Nomenclatura:** Aplicação de *PascalCase* para nomes de classes, métodos, propriedades (e.g., `ProdutoController`) e *camelCase* para variáveis locais (e.g., `faturamentoAcumulado`).
 - **Nullability:** Uso mandatório de *nullable reference types* (`string?`, `DateTime?`) para mitigar erros de referência nula. O tipo `decimal?` foi utilizado para campos opcionais de valores monetários.
 - **Tipagem e Performance:** Uso de construtores primários em Services (e.g., `PedidosService(EstoqueDbContext context)`) e utilização de *Records* ou *Value Tuples* para retorno eficiente de múltiplos valores (e.g., `Task<(List<string> labels, List<int> vendas, List<int> trocas)>`).
- **JavaScript/Front-end:** Aplicação de linter (*ESLint*) para impor padrões como *camelCase* e garantir a sintaxe moderna e consistente. Foi adotado o padrão de injetar variáveis de tradução do C# (`@Localizer["Chave"]`) em um objeto JS (`textTranslated`) para garantir a localização do conteúdo dinâmico do *Chart.js*.
- **Arquitetura e Convenções MVC (ASP.NET Core):** Separação estrita de responsabilidades, conforme as convenções:
 - **Controle e Roteamento:** Utilização de *Areas* (`/Areas/Estoque/`) para organizar módulos grandes e *Controllers* para roteamento e orquestração.
 - **Lógica de Negócio (Services):** O *Controller* delega a lógica de negócio aos *Services* (Injeção de Dependência). Foi imposto que toda consulta complexa do Entity Framework Core (EF Core), como `GroupBy` e `Sum`, seja encapsulada dentro do *Service*.

4.6 Cobertura de Código e Testes

A Cobertura de Código é uma métrica essencial que estabelece o nível de **confiabilidade e robustez** do código-fonte. E a definição estratégica de teste é fundamental para garantir essa qualidade no código e no funcionamento do sistema como um todo.

4.6.1 Metas e Escopo da Cobertura

A estratégia de cobertura e qualidade foi delineada para garantir a máxima confiabilidade nas áreas de maior risco do sistema. O escopo foi rigorosamente focado em três pilares:

- **Cobertura de Classes:** Foi estabelecida uma meta mínima de 80% de cobertura de classes para todos os módulos que encapsulam a lógica de negócio e complexidade algorítmica.
- **Segurança:** O ambiente web foi submetido a testes de segurança para garantir que o sistema esteja de acordo com os padrões de segurança, recebendo no mínimo uma nota A.
- **Integração:** Os testes de integração são essenciais para garantir que todos os ambientes necessários para que a aplicação funcione, estejam devidamente integrados, sem a existência de erros e falhas de segurança.
- **Usabilidade:** O esforço destes testes foi concentrado em fornecer a melhor experiência para o usuário, garantindo que todas as telas e processos sejam de fácil entedimento e interação.

4.6.2 Implementação e Validação

A qualidade do código e a aderência à meta de cobertura são asseguradas por uma combinação estratégica de testes manuais e automatizados, integrados ao fluxo de desenvolvimento.

- **Testes Unitários e de Integração: (C#):**

A cobertura de classes é implementada pelos *Unit Tests* (C#, que indicam a porcentagem de classes cobertas, e a integração dos ambientes é feita através de *Integration Tests* (C#), que indicam se a conexão entre os sistemas está funcional ou falha.

- **Testes de Ponta-a-Ponta (*End-to-End*, E2E):**

A validação dos fluxos completos de usuário, incluindo a interface gráfica, é realizada com **testes automatizados usando Selenium**, garantindo que o processo, sob a perspectiva do usuário, esteja funcional. E também foram feitos testes manuais para validar a usabilidade do sistema.

- **Teste de Segurança:**

O ambiente web do projeto foi submetido a testes de segurança em duas ferramentas: SSL Labs e Security Headers, ferramentas que avaliam se o ambiente web está de acordo com os padrões de segurança mais modernos.

4.7 Resultados dos Testes

Após definidas as metas de cobertura e os métodos de implementação e validação, a seguir consta os resultados dos testes propostos.

4.7.1 Cobertura de Classes

O projeto da VIP Penha é formado por diversas classes com diferentes funções, como mapeamento de tabelas do banco de dados, classes controladoras de requisições e classes que executam as regras de negócio. Foram Mapeadas 85 classes em nosso sistema, analisando o conteúdo e a relevância de cada classe, decidimos por testar 83,7% delas, totalizando 71 classes. As classes escolhidas para a testagem foram:

- As que envolviam entidades (classes mapeadoras de tabelas do banco de dados) que executam papéis de grande importância nos processos do negócio.
- As classes de serviço que executam métodos das regras de negócio
- As classes que contêm as rotas que aceitam requisições (controladoras)
- As classes que geram as telas do sistema com base na biblioteca JJ Masterdata

Quadro 9 – Resumo Consolidado Estoque.Domain

Tipo	Total	Candidatas a teste	Cobertura esperada
Entities	22	20	91%
Enums	7	4	57%
Models	13	9	69%
Total Geral	42	33	72%

Fonte: Elaborado pelos autores

Quadro 10 – Resumo Consolidado Estoque.Infrastructure

Tipo	Total	Candidatas a teste	Cobertura esperada
Data	1	1	100%
Factory	2	1	50%
Services	18	17	94%
Total Geral	26	24	86%

Fonte: Elaborado pelos autores

Quadro 11 – Resumo Consolidado Estoque.Web

Tipo	Total	Candidatas a teste	Cobertura esperada
Admin (Controllers)	6	6	100%
Estoque (Controllers)	10	10	100%
Identity (Controllers)	4	4	100%
Configuration	14	8	73%
Total Geral	34	28	93%

Fonte: Elaborado pelos autores

Todas as classes estão mapeadas e disponíveis no arquivo do projeto em:

<https://github.com/VitorDaSilvaOliveira/Projeto-Integrado-IFSP/tree/master/docs/Anexos>.

4.7.2 Segurança

O ambiente web do projeto foi submetido a testes de segurança em duas ferramentas: SSL Labs e Security Headers obtivemos resultados positivos em ambos, o que demonstra a conformidade das práticas adotadas com os padrões modernos de segurança da web.

- **Avaliação SSL Labs — Nota A** O domínio do projeto recebeu nota A no teste do SSL Labs, o que indica uma implementação sólida e segura do protocolo HTTPS/TLS. Os critérios avaliados apresentaram os seguintes resultados:

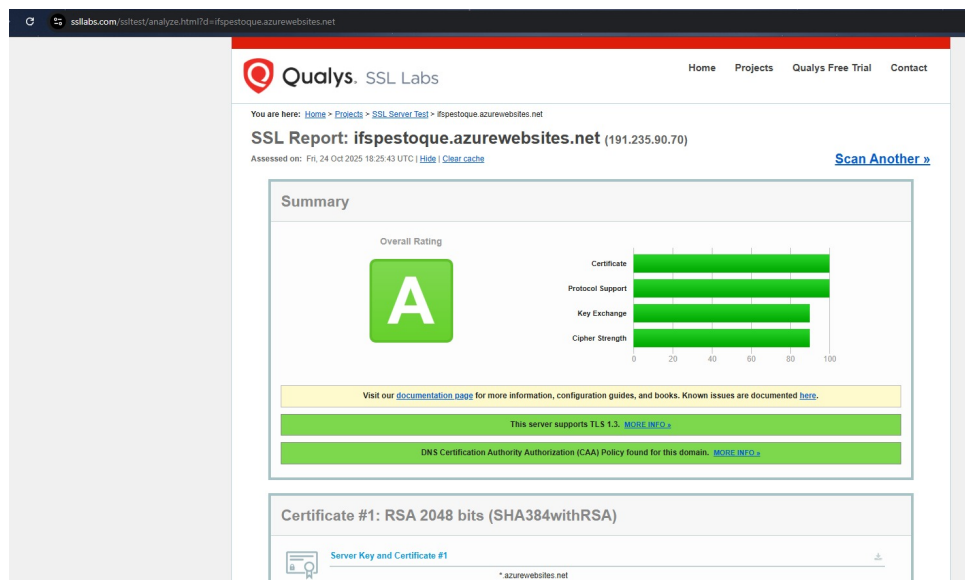


Figura 4 – Resultado do SSL Labs

Quadro 12 – Resultado SSL Labs

Critério	Pontuação	Explicação
Certificate	100/100	O certificado SSL/TLS está corretamente configurado, é válido, confiável e emitido por uma autoridade certificadora reconhecida. Garante autenticidade e confiança na comunicação.
Protocol Support	100/100	Apenas versões seguras dos protocolos TLS estão habilitadas, com desativação de versões obsoletas (como SSLv3 e TLS 1.0/1.1), evitando vulnerabilidades conhecidas.
Key Exchange	90/100	O mecanismo de troca de chaves usa algoritmos modernos (como ECDHE) que fornecem sigilo perfeito (Perfect Forward Secrecy), garantindo que sessões passadas não possam ser descriptografadas mesmo em caso de comprometimento futuro da chave privada.
Cipher Strength	90/100	Os conjuntos de cifras (ciphers) utilizados são fortes, com chaves de 256 bits e algoritmos seguros (AES-GCM, SHA-256), protegendo os dados contra ataques de força bruta e descriptografia não autorizada.

Fonte: Elaborado pelos autores

- **Avaliação Security Headers** O projeto também foi analisado pela ferramenta Security Headers, que verifica a presença e configuração de cabeçalhos HTTP responsáveis por mitigar ataques comuns (como XSS, clickjacking e data leaks).

O site obteve todas as diretivas essenciais ativas, conforme o checklist abaixo:

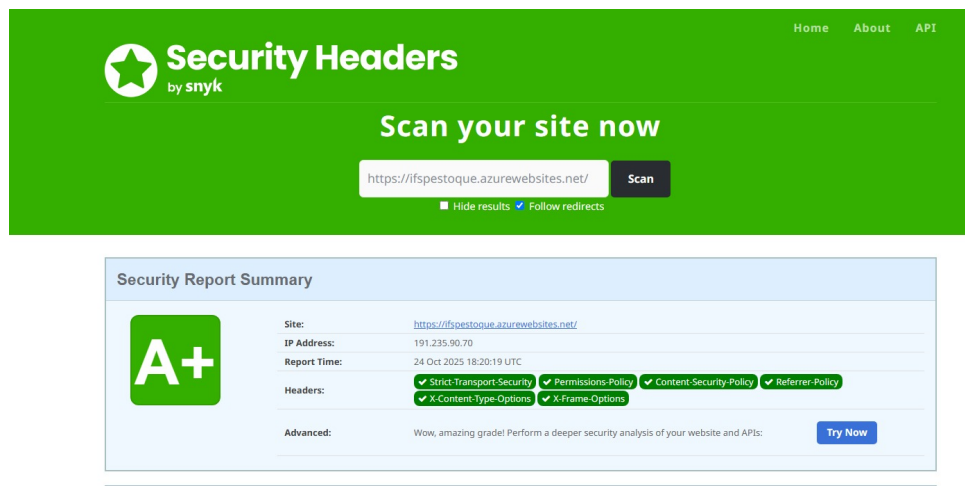


Figura 5 – Resultado do Security Headers

Quadro 13 – Resultado Security Headers

Cabeçalho	Status	Função
Strict-Transport-Security (HSTS)	Positivo	Obriga o navegador a se conectar sempre via HTTPS, evitando downgrade attacks.
Permissions-Policy	Positivo	Restringe o acesso a APIs e recursos sensíveis (como câmera, microfone e geolocalização).
Content-Security-Policy (CSP)	Positivo	Impede a execução de scripts não autorizados, protegendo contra Cross-Site Scripting (XSS).
Referrer-Policy	Positivo	Controla o envio de informações de origem, reduzindo o vazamento de dados de navegação.
X-Content-Type-Options	Positivo	Evita que navegadores interpretem arquivos com tipos MIME incorretos, prevenindo ataques de injeção.
X-Frame-Options	Positivo	Bloqueia o carregamento do site em iframes externos, prevenindo ataques de clickjacking.

Fonte: Elaborado pelos autores

Conclusões a respeito de segurança Os resultados das análises indicam que o site está altamente seguro, contendo boas práticas como

- Comunicação criptografada e confiável.
- Cabeçalhos HTTP de segurança corretamente configurados.
- Mitigação de ataques comuns de rede e navegador.

4.7.3 Integração

A aplicação da VIP Penha necessita de duas integrações para funcionar devidamente, sendo elas a conexão com um banco de dados SQL Server e a integração com um ambiente de emissão de notas fiscais.

- **Integração com o Banco de Dados**

O projeto realiza sua integração com o banco de dados *SQL Server* por meio do *Entity Framework Core* (EF Core), um ORM (Object-Relational Mapper) que permite mapear as entidades de domínio diretamente para tabelas do banco, garantindo abstração da camada de persistência e manutenção simplificada. A configuração dessa integração é feita na classe *ConfigurationExtensions*, onde o contexto *EstoqueDbContext* é registrado com a string de conexão *DefaultConnection*. Essa abordagem permite que o projeto: Utilize injeção de dependência para gerenciar o contexto de dados, centralize a migração de esquemas dentro do assembly *Estoque.Infrastructure* e mantenha consistência entre as entidades de domínio (*Estoque.Domain.Entities*) e suas representações no banco. Além disso, o uso de repositórios e serviços garante que as operações de CRUD e as transações sejam tratadas de forma controlada, assegurando integridade e rastreabilidade dos dados.

- **Integração com o Ambiente da Nota Fiscal**

O ambiente da Nota Fiscal foi integrado ao sistema por meio do serviço *NotaFiscalService*, localizado na camada *Estoque.Infrastructure.Services*. Esse serviço tem como responsabilidade principal o gerenciamento dos processos de emissão, registro e comunicação das notas fiscais, integrando o sistema a um ambiente fiscal externo (seja de homologação ou produção). A integração segue princípios de arquitetura limpa, sendo mediada por:

- **Serviços especializados responsáveis por comunicação HTTP segura com a API fiscal.**
- **Modelos de dados padronizados que representam os documentos fiscais eletrônicos.**
- **Tratamento de exceções e logs detalhados para auditoria e rastreabilidade.**

Essa integração é configurada e injetada automaticamente pelo método *AddEstoqueServices()*, que registra o *NotaFiscalService* e outros componentes auxiliares: *services.AddScoped<NotaFiscalService>()*; Dessa forma, o ambiente de notas fiscais é completamente integrado ao fluxo do sistema, permitindo:

- Automatização da emissão de notas após o registro de vendas ou movimentações.
- Sincronização de status fiscais (autorização, cancelamento, inutilização)
- Segurança no tráfego de dados por meio de HTTPS e certificados digitais válidos.

- **Conclusão sobre Integração**

A integração entre o sistema, o banco de dados e o ambiente da nota fiscal foi projetada de forma modular e segura. Enquanto o *Entity Framework Core* assegura persistência confiável e manutenção simples dos dados internos, o *NotaFiscalService* garante interoperabilidade com os sistemas fiscais externos, proporcionando automação, conformidade legal e rastreabilidade completa das operações.

4.7.4 Usabilidade

Testes de usabilidade do sistema foram executados almejando a melhor experiência possível para o usuário. Foram apresentadas as visões gerais do sistema e definidos objetivos específicos a serem alcançados durante a utilização por um usuário externo ao desenvolvimento do projeto. Inicialmente, recebemos feedbacks negativos quanto à clareza dos processos e à disposição de algumas funcionalidades na interface.

Com base nessas observações, foram revisados os fluxos de navegação e reorganizados elementos visuais para tornar o sistema mais intuitivo. O participante relatou dificuldades principalmente na compreensão da sequência correta de ações, o que indicou a necessidade de uma revisão nos fluxos dos processos e das interfaces gráficas.

Após as melhorias, uma nova rodada de testes foi realizada, o usuário conseguiu realizar as tarefas propostas com muito mais fluidez e clareza. Desta forma, foi constatada a eficácia das modificações, auxiliando na evolução do produto final.

4.8 Segurança, Privacidade, Legislação

Este tópico é dedicado a explicar sobre as questões de segurança e legislação relevantes para o projeto.

4.8.1 Critérios de Segurança e Privacidade

O sistema de gerenciamento de estoque foi projetado com critérios de segurança e privacidade para garantir a integridade das informações e proteger os dados dos usuários. As principais medidas adotadas no projeto incluem:

- **Autenticação e Autorização:** O acesso ao sistema é controlado por meio de autenticação de usuários, utilizando o *ASP.NET Identity*. Cada usuário precisa estar autenticado para acessar funcionalidades sensíveis, como cadastro de produtos, movimentação de estoque ou relatórios.
- **Criptografia de Senhas:** As senhas dos usuários são armazenadas de forma criptografada (hash) no banco de dados, conforme práticas recomendadas pelo *Entity Framework*.
- **Boas Práticas de Privacidade:** Os dados pessoais dos usuários, como nome e e-mail, são utilizados apenas para fins de autenticação e gerenciamento interno, e não são compartilhados com terceiros.

4.8.2 Legislação

A Lei Geral de Proteção de Dados (LGPD - Lei nº 13.709/2018) garante transparência diante ao uso de dados pessoais garantindo que os dados coletados sejam utilizados exclusivamente para fins de autenticação, autorização e controle de acesso ao sistema. Sendo assim, o sistema segue as diretrizes da LGPD, adotando medidas técnicas e administrativas adequadas para proteger os dados pessoais contra acessos não autorizados, perda ou vazamento.

Além da LGPD, o sistema também considera princípios estabelecidos no Marco Civil da Internet (Lei nº 12.965/2014), que estabelece garantias, direitos e deveres para o uso da internet no Brasil. Nesse sentido, o projeto reflete esses princípios ao promover a proteção da privacidade dos usuários, a segurança das informações e a transparência nas operações de tratamento de dados, assegurando que o uso da aplicação respeite os direitos fundamentais de liberdade e privacidade no ambiente digital.

4.9 Modelo de Banco de Dados

O modelo de banco de dados é responsável por organizar e estruturar as informações utilizadas no sistema, garantindo integridade, consistência e facilidade no acesso aos dados. O MER (Modelo Entidade-Relacionamento) e o DER (Diagrama Entidade-Relacionamento) representam graficamente as entidades, atributos e relacionamentos do sistema, enquanto o dicionário de dados descreve detalhadamente cada campo presente no banco, incluindo tipo, tamanho e função. O modelo foi projetado para gerenciar de forma eficiente o controle de estoque, permitindo registrar entradas, saídas, usuários, produtos, fornecedores e categorias.

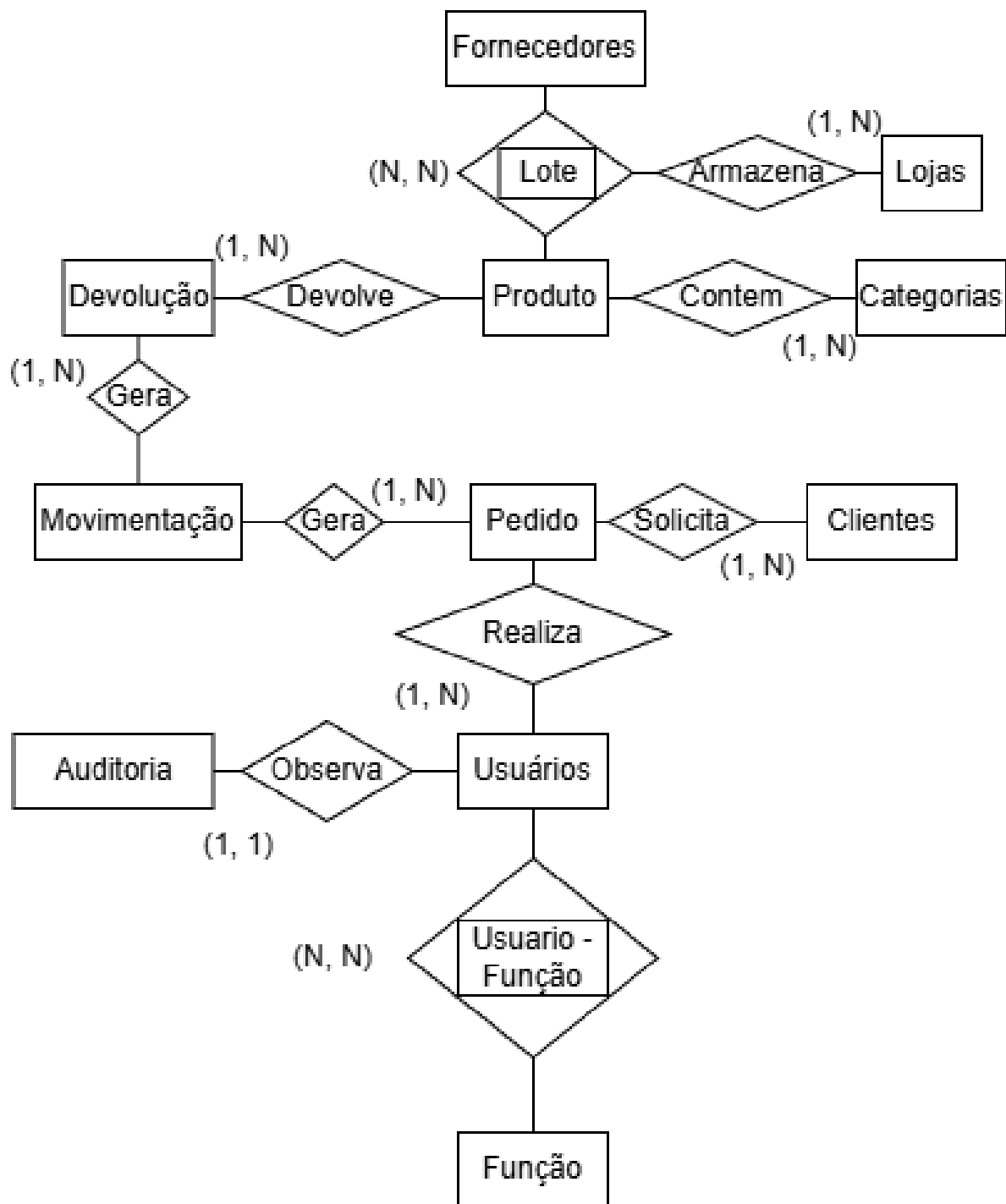


Figura 6 – Modelo de Entidade-Relacionamento

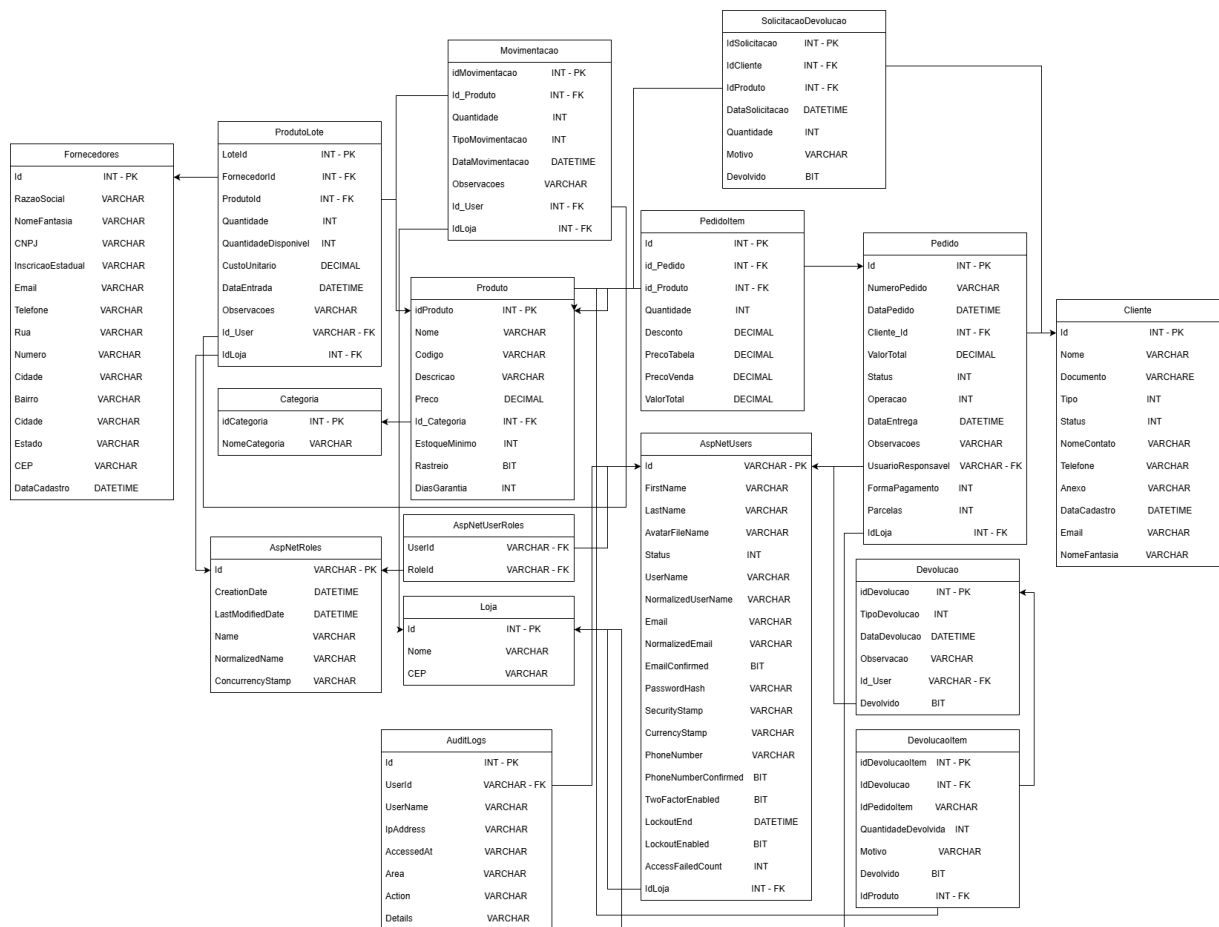


Figura 7 – Diagrama de Entidade-Relacionamento

4.10 Cronograma

Esta seção apresenta uma visão detalhada das principais fases do projeto, focando nas atividades e entregáveis cruciais para o desenvolvimento.

4.10.1 Cronograma de Atividades – Primeiro Semestre

O desenvolvimento do sistema de gerenciamento de estoque será realizado ao longo de dois semestres letivos, estando o presente cronograma correspondente às atividades planejadas para o primeiro semestre. Neste período inicial, o foco principal está na fundamentação e planejamento do projeto, bem como na construção da prova de conceito e definição do escopo mínimo viável.

- No início do semestre, tem-se a fase de Desenvolvimento do Tema (Entrega do marco dia 08/04/2025), que contempla a escolha do assunto central do projeto e a definição do parceiro institucional. Essa etapa foca no levantamento detalhado dos requisitos (funcionais e não funcionais) e na análise de viabilidade do sistema proposto.
- Posteriormente, desenvolve-se a fase de Desenho da Aplicação (Entrega do marco dia 29/04/2025), com foco na estruturação técnica da solução. Aqui, são entregues o projeto do banco de dados (MER e DER), os diagramas de arquitetura (UML) e a modelagem dos processos de negócio (AS IS e TO BE).
- A seguir, passa-se à etapa de Prova de Conceito (Entrega do marco dia 20/05/2025), na qual se realiza a escolha das tecnologias mais adequadas. O resultado é o desenvolvimento dos protótipos iniciais de interface, dos scripts de criação do banco de dados e o início da implementação do *back-end* do sistema.
- Por fim, inicia-se a fase da Criação do MVP (Produto Mínimo Viável) (Entrega do marco dia 17/06/2025). Essa etapa inclui a definição clara do problema e das prioridades, delimitando o escopo da aplicação com suas restrições. O produto final é a identificação das funcionalidades mínimas essenciais e das integrações necessárias.
- Paralelamente, é elaborada a fase de Análise e Documentação (Entrega do marco dia 18/07/2025), cujo objetivo é consolidar os dados obtidos. Esta seção inclui a elaboração da introdução, revisão da literatura, gestão do projeto e análise de viabilidade financeira, encerrando com as considerações finais sobre o processo.

5 Viabilidade Financeira

5.1 Custos

A Tabela 9 apresenta uma análise detalhada dos custos de desenvolvimento, com foco no investimento em mão de obra, que representa o principal ativo do projeto. A metodologia de cálculo é simples e transparente: o esforço previsto (em horas) é multiplicado pela taxa horária específica de cada especialista (desenvolvedores, analistas, gestores), garantindo uma apuração precisa e fiel aos valores de mercado.

Esta abordagem, portanto, transcende um simples levantamento numérico. Ela não só oferece clareza sobre os custos inerentes a cada fase do desenvolvimento, mas, ao ser projetada sobre a duração total do desenvolvimento, transforma-se em uma poderosa ferramenta de gestão estratégica e preditiva. A projeção temporal permite uma análise aprofundada da distribuição dos custos mensais e globais, oferecendo previsibilidade e permitindo a antecipação de picos de investimento. Com isso, a gestão pode mitigar riscos financeiros, como desvios orçamentários, antes que se concretizem.

Consequentemente, a tabela constitui um pilar para um planejamento financeiro robusto, que justifica o orçamento perante stakeholders e estabelece uma linha de base para o controle rigoroso das despesas, viabilizando a alocação otimizada de recursos e assegurando a sustentabilidade operacional do projeto.

Tabela 3 – Detalhamento dos Custos de Desenvolvimento.

Função	Quant.	Horas	Valor/Hora (R\$)	Total (R\$)
Product Owner (PO)	1	400	35,00	14.000,00
DBA	1	400	25,50	10.200,00
Desenvolvedor Front-end	3	250	18,00	13.500,00
Desenvolvedor Back-end	3	250	20,00	15.000,00
QA (Testes)	4	100	22,00	8.800,00
Total	7	-	-	61.500,00

Fonte: Elaborado pelos autores.

Na tabela 10 é representada a estrutura mensal da empresa, detalhando os custos associados às instalações, equipamentos e serviços de TI. No item 2.1, destacam-se os gastos com aluguel de espaço e mobiliário, totalizando 0 reais por mês. Já no item 2.2, referente à aquisição de equipamentos, observa-se que não há custos neste momento. No entanto, no item 2.3, estão listados os custos mensais de serviços de TI.

Tabela 4 – Estrutura Mensal da Empresa.

Item	Quantidade	Valor (R\$)
2. Estrutura da empresa (mensal)		
2.1. Instalações - Aluguel	-	0,00
2.2. Equipamentos (TI e Outros)		
- Computadores/notebooks	0	0,00
- Servidores proprietários	0	0,00
- Outros (mobiliário, etc.)	0	0,00
2.3. Serviços de TI		
- Ferramentas de desenvolvimento	-	0,00
- Hospedagem e banco de dados	-	100,00
Total mensal		R\$ 100,00

Fonte: Elaborado pelos autores.

6 Considerações Finais

Ao longo do desenvolvimento deste projeto, que visa a criação de um sistema de controle de estoque para um estabelecimento comercial, foi possível constatar de soluções tecnológicas para otimizar processos e garantir a eficiência operacional em ambientes comerciais. A iniciativa de desenvolver este sistema surgiu da necessidade de modernizar a gestão em um cenário de mercado cada vez mais competitivo. O sistema proposto representa uma ferramenta estratégica para a loja, Vip Penha permitindo o cadastro detalhado de produtos, o controle preciso de entradas e saídas de mercadorias, e a geração de relatórios gerenciais. Essas funcionalidades são necessárias para reduzir erros manuais, tempo dedicado à organização do estoque e, conseqüentemente, aprimorar a tomada de decisões. A implementação de uma metodologia ágil, como o Kanban, foi de grande ajuda para a organização do projeto. Conclui-se que a aplicação deste sistema resulta em uma melhora substancial na gestão do estoque do estabelecimento, contribuindo para a redução de perdas, a otimização do fluxo de trabalho e por consequência ganhos econômicos. Além disso, este projeto demonstra o potencial das tecnologias da informação para resolver desafios práticos do cotidiano de empreendimentos. A contínua evolução e aprimoramento de tais sistemas é essencial para que empreendimentos e empresas possam manter sua competitividade e prosperar no mercado atual.

APÊNDICE A – Mapeamento de Processos

A.1 Processo AS IS e TO BE

A.1.1 Processo AS IS (Situação Atual)

Descrição Geral

A loja faz controle de estoque manualmente, por anotações em papel feitas pelo dono.

Etapas do Processo

1. **Recebimento de produtos (entrada):** O dono recebe os produtos de fornecedores. Ele anota em um caderno: nome do produto, quantidade recebida, data e fornecedor.
2. **Venda de produtos (saída):** Ao vender um produto, o dono anota à mão no mesmo caderno: nome do produto, quantidade vendida, data e valor da venda.
3. **Controle de estoque:** O dono consulta o caderno para verificar a quantidade de produtos disponíveis. Não há atualização automática; o controle depende da leitura e interpretação das anotações.
4. **Relatório mensal:** O dono folheia o caderno, soma manualmente as entradas e saídas e tenta fazer um resumo no final do mês. Processo demorado e sujeito a erros.

Problemas Identificados

- Alto risco de erro humano.
- Dificuldade em rastrear movimentações específicas.
- Falta de relatórios precisos.
- Perda ou danos ao caderno comprometem todo o controle.
- Nenhuma visibilidade em tempo real do estoque.

A.1.2 Processo TO BE (Situação Proposta com Sistema)

Descrição Geral

Será implementado um sistema informatizado para registrar entradas, saídas e gerar relatórios mensais automaticamente.

Etapas do Processo com o Sistema

1. **Recebimento de produtos (entrada):** O dono ou funcionário acessa o sistema. Registra os produtos recebidos: nome, quantidade, fornecedor, data, nota fiscal. O estoque é atualizado automaticamente.
2. **Venda de produtos (saída):** No momento da venda, o produto é registrado como “vendido” no sistema. O sistema subtrai automaticamente a quantidade vendida do estoque.
3. **Controle de estoque:** O sistema exibe em tempo real o estoque disponível de cada item. Alertas de baixo estoque podem ser configurados.
4. **Relatório mensal de movimentações:** Ao fim do mês (ou sob demanda), o sistema gera automaticamente um relatório com:
 - Total de entradas por produto.
 - Total de saídas por produto.
 - Saldo atual.
 - Produtos mais vendidos.
 - Histórico por período.

Benefícios Esperados

- Redução de erros manuais.
- Facilidade no rastreamento de movimentações.
- Economia de tempo.
- Maior controle e visibilidade sobre o estoque.
- Dados organizados e acessíveis a qualquer momento.

Referências

Agência Brasileira de Desenvolvimento Industrial (ABDI). *Mapa de Maturidade Digital 2024: cerca de 50% dos pequenos negócios usam mídias digitais para vender produtos ou serviços*. 2024. Acesso em: 3 jun. 2025. Disponível em: <https://www.abdi.com.br/mapa-de-maturidade-digital-2024-cerca-de-50-dos-pequenos-negocios-usam-midias-digitais-para-vender-produtos-ou-servicos/>. Citado na página 12.

Alura. *HTML, CSS e Javascript, quais as diferenças?* 2024. Acessado em: 9 jun. 2025. Disponível em: <https://www.alura.com.br/artigos/html-css-e-js-definicoes?srsltid=AfmBOoqiW-sINTyk-x29HrxcgaO4i9guj4vwCxNVSc98Oucs8RYIK3L5>. Citado na página 33.

Associação Brasileira da Indústria Elétrica e Eletrônica (ABINEE). *Desempenho do Setor Elétrico e Eletrônico – Indicadores 2024*. 2024. Acesso em: 10 out. 2025. Disponível em: <https://www.abinee.org.br/faturamento-do-setor-cresce-11-em-2024/>. Citado na página 9.

BALLOU, R. H. *Gerenciamento da Cadeia de Suprimentos: Logística Empresarial*. Porto Alegre: Bookman, 2006. Citado na página 12.

Bootstrap. *Comece com Bootstrap*. 2024. Acessado em: 9 jun. 2025. Disponível em: <https://getbootstrap.com.br/docs/4.1/getting-started/introduction/>. Citado na página 33.

Cast4IT. *Título do Documento ou Relatório*. 2024. <https://www.cast4it.com/gestao-de-estoque/>. Acesso em: 26 out. 2025. Citado na página 16.

CONFERENCE, W. F. A. C. The study of the ancient near east in the twenty-first century. In: *The William Foxwell Albright Centennial Conference*. [S.l.]: Eisenbrauns, 1996. p. 14–15. ISBN 9780931464966. Citado na página 15.

DIAS, J. D. S.; OUTROS. Uma breve análise sobre a evolução da logística. In: _____. *Logística: contribuições para melhorias na produção e nos resultados*. Editora Científica, 2021. v. 1, n. 1, p. 64–81. Acesso em: 10 maio 2025. Disponível em: <https://www.editoracientifica.com.br/books/chapter/210303726>. Citado na página 15.

JJConsulting. *Welcome to the JJ MasterData documentation!* 2025. Acessado em: 9 jun. 2025. Disponível em: <https://md.jjconsulting.tech/>. Citado na página 34.

jQuery. *jQuery*. 2025. Acessado em: 9 jun. 2025. Disponível em: <https://jquery.com/>. Citado na página 34.

LAUDON, K. C.; LAUDON, J. P. *Sistemas de Informação Gerenciais*. São Paulo: Pearson Education do Brasil, 2014. Citado na página 9.

Microsoft. *Razor Referência de sintaxe para ASP.NET Core*. 2024. Acessado em: 9 jun. 2025. Disponível em: <https://learn.microsoft.com/pt-br/aspnet/core/mvc/views/razor?view=aspnetcore-9.0>. Citado na página 33.

Microsoft. *Visão geral do Entity Framework Core – EF Core*. 2024. Acessado em: 9 jun. 2025. Disponível em: <<https://learn.microsoft.com/pt-br/ef/core/>>. Citado na página 34.

Microsoft. *Introdução ao Identity no ASP.NET Core*. 2025. Acessado em: 9 jun. 2025. Disponível em: <<https://learn.microsoft.com/pt-br/aspnet/core/security/authentication/identity?view=aspnetcore-9.0>>. Citado na página 34.

Microsoft. *O que é o AspNet*. 2025. Acessado em: 2 jun. 2025. Disponível em: <<https://dotnet.microsoft.com/pt-br/learn/aspnet/what-is-aspnet>>. Citado na página 34.

Microsoft. *O que é o Azure?* 2025. Acessado em: 2 jun. 2025. Disponível em: <<https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-azure>>. Citado na página 34.

Microsoft. *O que é o SQLServer*. 2025. Acessado em: 2 jun. 2025. Disponível em: <<https://learn.microsoft.com/pt-br/sql/sql-server/what-is-sql-server?view=sql-server-ver17>>. Citado na página 34.

OHNO, T. *O Sistema Toyota de Produção: além da produção em larga escala*. Porto Alegre: Bookman, 1997. Citado na página 15.

SILVA, N. d. C. et al. Gestão de estoques com inventário físico: Um estudo de caso de impactos na acuracidade de estoque de uma rede de material de construção. *Revista Mythos*, v. 14, n. 2, p. 7–20, 2020. Citado na página 16.

SLACK, N.; CHAMBERS, S.; JOHNSTON, R. *Administração da Produção*. São Paulo: Atlas, 2013. Citado na página 15.