



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
CAMPUS TERESINA CENTRAL
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

KAIRON WILLYAN RIBEIRO DE MELO

EASY SALES: software para gestão de vendas e relatórios inteligentes

TERESINA, PIAUÍ
2025

KAIRON WILLYAN RIBEIRO DE MELO

EASY SALES: software para gestão de vendas e relatórios inteligentes

Trabalho de Conclusão de Curso apresentado como exigência parcial para obtenção do diploma do Curso de Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Teresina Central.

Orientador: Prof. M^º. Ely Miranda

TERESINA, PIAUÍ
2025

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD

Melo, Kairon Willyan Ribeiro de
M528e Easy Sales : software para gestão de vendas e relatórios inteligentes /
Kairon Willyan Ribeiro de Melo. - 2025.
85 f.: il. color.

Trabalho de Conclusão de Curso (Tecnologia em Análise e
Desenvolvimento de Sistemas) - Instituto Federal de Educação, Ciência e
Tecnologia do Piauí, Campus Teresina Central, 2025.

Orientador : Prof Me. Ely da Silva Miranda .

1. Gestão de Vendas . 2. Pequenas e médias empresas . 3. Automação
de processos . 4. Desenvolvimento de software. I. Título.

CDD - 004.21

Elaborado por Denise de Paula Veras Aquino CRB 3/962

KAIRON WILLYAN RIBEIRO DE MELO

EASY SALES: software para gestão de vendas e relatórios inteligentes

Trabalho de Conclusão de Curso apresentado como exigência parcial para obtenção do diploma do Curso de Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Teresina Central.

Aprovada em 13 de Fevereiro de 2025.

BANCA EXAMINADORA:

Prof. M^e. Ely Miranda(Orientador)
Instituto Federal do Piauí (IFPI)

Prof. M^e. Fernando Castelo Branco Gonçalves Santana
Instituto Federal do Piauí (IFPI)

Prof. Dr. Thiago Alves Elias da Silva
Instituto Federal do Piauí (IFPI)

TERESINA, PIAUÍ
2025

Esta dedicatória de TCC é para Toda a minha família, que me deu apoio durante todo o curso, para minha namorada Marisa, que esteve comigo durante todo o curso no IFPI e ao meu grande e querido Professor Ely Miranda, que mais do que um orientador, foi um verdadeiro guia na selva acadêmica. Sua habilidade de encontrar erros onde eu só via 'trabalho finalizado' foi impressionante. Obrigado por ensinar mais do que esperava aprender.

AGRADECIMENTOS

Agradeço às pessoas as quais convivi a longo desses anos de curso, que me incentivaram e que certamente tiveram impacto na minha formação acadêmica. Agradeço ao meu Deus, que me deu força quando mais precisava. Agradeço a minha família, por todo apoio e paciência nas minhas ausências enquanto me dedicava a esse trabalho. Por fim, agradeço a todos os professores e servidores do IFPI do Campus Teresina Central, pelas correções e ensinamentos que me permitiram apresentar um melhor desempenho no meu processo de formação profissional ao longo do curso.

*A tarefa não é tanto ver aquilo que ninguém viu, mas pensar o que ninguém ainda
pensou sobre aquilo que todo mundo vê.”*
Arthur Schopenhauer

RESUMO

Este projeto propõe o desenvolvimento do Easy Sales, um sistema de gestão empresarial voltado para pequenas e médias empresas, com o objetivo de otimizar o gerenciamento de vendas, estoque, clientes e fornecedores. Utilizando tecnologias modernas como Ionic e Spring Boot, o sistema está sendo estruturado de maneira modular e escalável; no entanto, essas características ainda precisam ser validadas em testes práticos. Além disso, com base nas entrevistas realizadas com potenciais usuários, os feedbacks obtidos têm sido fundamentais para ajustar as funcionalidades às necessidades reais do mercado. Entre as principais propostas do sistema estão a futura integração com plataformas de e-commerce e contabilidade, além da implementação de uma versão offline.

Palavras-chaves: gestão de vendas; pequenas e médias empresas; automação de processos; desenvolvimento de software.

ABSTRACT

This project proposes the development of Easy Sales, a business management system aimed at small and medium-sized companies, with the aim of optimizing the management of sales, inventory, customers and suppliers. Using technologies modern technologies such as Ionic and Spring Boot, the system is being structured in a modular and calable; however, these characteristics still need to be validated in practical tests. Furthermore, based on interviews conducted with potential users, the feedback obtained has been fundamental in adjusting the functionalities to the real needs of the market. Among the main proposals of the system are the future integration with e-commerce and accounting platforms, in addition to implementing an offline version.

Keywords: sales management; small and medium enterprises; process automation; software development.

LISTA DE ILUSTRAÇÕES

Figura 1 – Sale Easy	20
Figura 2 – Sale Easy Vendas	21
Figura 3 – Egestor	21
Figura 4 – Egestor Relatórios	22
Figura 5 – Bling venda	22
Figura 6 – Bling Cadastrar Vendas	23
Figura 7 – Software Trello.	26
Figura 8 – Software Jira.	26
Figura 9 – Monday.	27
Figura 10 – Backlog do produto	29
Figura 11 – Sprint	30
Figura 12 – Protótipos de Papel	31
Figura 13 – Protótipos de Alta Fidelidade	31
Figura 14 – Protótipos Funcionais	32
Figura 15 – Diagrama de Classe	33
Figura 16 – Diagrama de Caso de Uso	34
Figura 17 – Vue e Ionic	39
Figura 18 – Java e Spring Boot	40
Figura 19 – Supabase e PostgreSQL	41
Figura 20 – Keycloak	41
Figura 21 – Diagrama de Caso de Uso	44
Figura 22 – Diagrama de Classes	46
Figura 23 – Diagrama de Sequência do Registro de Usuário	47
Figura 24 – Diagrama de Sequência de Login	48
Figura 25 – Diagrama de Sequência de Cadastro de Produto	49
Figura 26 – Diagrama de Sequência de Cadastro de Venda	50
Figura 27 – Diagrama de Implantação do Sistema	51
Figura 28 – Arquitetura da Aplicação	52
Figura 29 – Relatório de Segurança no SonarQube	63
Figura 30 – Resultados do teste no JMeter	64
Figura 31 – Sprint 1	65
Figura 32 – Sprint 2	66
Figura 33 – Sprint 3	66
Figura 34 – Sprint 4	67
Figura 35 – Fluxo de Cadastro de Empresas	69
Figura 36 – Listagem de Empresas	70

Figura 37 – Alternância entre Empresas	71
Figura 38 – Tela Inicial do Sistema	72
Figura 39 – Menu de Navegação Lateral	73
Figura 40 – Cadastro de Fornecedores	74
Figura 41 – Cadastro de Produtos	75
Figura 42 – Listagem e Detalhamento de Produtos	76
Figura 43 – Listagem e Detalhamento de Compras	77
Figura 44 – Interface de Listagem e Detalhamento de Vendas	78
Figura 45 – Dashboard com Indicadores de Desempenho	79
Figura 46 – Dashboard de Produtos	80
Figura 47 – Dashboard de Vendas	81
Figura 48 – Interface do Relatório de Empresas	82
Figura 49 – Interface de Negociação de Dívidas	83

LISTA DE ABREVIATURAS E SIGLAS

IFPI	Instituto Federal de Educação, Ciência e Tecnologia do Piauí
PME	Pequena e Média Empresa
ABComm	Associação Brasileira de Comércio Eletrônico
Sebrae	Serviço Brasileiro de Apoio às Micro e Pequenas Empresas
CNI	Confederação Nacional da Indústria
ERP	Enterprise Resource Planning
UML	Unified Modeling Language
dto	Data Transfer Objects

LISTA DE SÍMBOLOS

%	Porcentagem
---	-------------

SUMÁRIO

1	INTRODUÇÃO	15
1.1	JUSTIFICATIVA	16
1.2	OBJETIVO GERAL	17
1.3	OBJETIVOS ESPECÍFICOS	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	IMPACTO DOS SISTEMAS DE GESTÃO DE VENDAS NAS PMES E AUTÔNOMOS	18
2.1.1	Benefícios dos Sistemas de Gestão de Vendas	19
2.1.2	Estudos de Caso em PMEs e Autônomos	19
2.1.3	Desafios Específicos e Soluções	19
2.2	TRABALHOS CORRELATOS	20
3	METODOLOGIA	24
3.1	VISÃO GERAL	24
3.2	FERRAMENTAS DE GERENCIAMENTO DE PROJETO	25
3.3	BACKLOG	28
3.4	SPRINT	29
3.5	PROTOTIPAÇÃO	31
3.6	PRINCIPAIS DIAGRAMAS	32
3.7	FORMA DE VALIDAÇÃO COM CLIENTES	34
3.7.1	Entrevistas com Empreendedores	34
3.7.2	Validações Adicionais	36
3.7.2.1	Revisão Heurística de Usabilidade	37
3.7.2.2	Testes de Simulação de Cenários de Uso	37
3.7.2.3	Testes de Compatibilidade do Sistema	37
3.7.2.4	Validação de Requisitos	38
3.7.3	Justificativa para a Quantidade de Entrevistas	38
3.8	TECNOLOGIAS A SEREM UTILIZADAS	38
3.8.1	Front-end	39
3.8.2	Back-end	40
3.8.3	Trade-offs e desafios na escolha das tecnologias	41
4	RESULTADOS	43
4.1	APRESENTAÇÃO DO SOFTWARE	43
4.2	MODELAGEM	44

4.2.1	Diagrama de casos de uso	44
4.2.2	Diagrama de Classes	45
4.3	DIAGRAMAS DE SEQUÊNCIA	47
4.3.1	Registro de Usuário	47
4.3.2	Login no Sistema	48
4.3.3	Cadastro de Produto	49
4.3.4	Cadastro de Venda	50
4.4	DIAGRAMA DE IMPLANTAÇÃO	51
4.4.1	Arquitetura da aplicação	51
4.5	API	53
4.5.1	Autenticação e Autorização	53
4.5.2	Entidades e Controladores	53
4.5.3	ClienteController	54
4.5.4	FornecedorController	54
4.5.5	EmpresaController	55
4.5.6	CompraController	55
4.5.7	VendaController	56
4.5.8	SaleController	57
4.5.9	EnterpriseController	57
4.5.10	SupplierController	57
4.5.11	PurchaseController	58
4.5.12	DebtNegotiationController	59
4.5.13	ClientController	59
4.5.14	StockController	60
4.5.15	Relatórios	60
4.5.16	Tratamento de Erros	61
4.6	TESTES E VALIDAÇÕES	62
4.6.1	Testes De Segurança	62
4.6.2	Testes De Performance	64
4.6.2.1	Resultados Obtidos	64
4.6.2.2	Interpretação dos Resultados	64
4.6.2.3	Conclusão e Limites do Teste	65
4.7	BACKLOG E SPRINTS	65
4.8	CRONOGRAMA DE EXECUÇÃO	67
4.9	INTERFACES DESENVOLVIDAS	68
5	CONCLUSÃO E TRABALHOS FUTUROS	84
	REFERÊNCIAS	85

1 INTRODUÇÃO

O setor de vendas é fundamental para a sustentabilidade e crescimento de pequenas e médias empresas (PMEs) e vendedores autônomos. A digitalização, intensificada pela pandemia da COVID-19, impulsionou essas empresas a adotarem soluções tecnológicas para otimizar processos de vendas, gerenciar estoques e manter um bom relacionamento com clientes. No Brasil, o comércio eletrônico movimentou R\$196,1 bilhões em 2023, representando um aumento de 4,8% em relação a 2022 (MINISTÉRIO..., 2024).

Apesar desse crescimento, muitas PMEs ainda enfrentam desafios na digitalização de seus processos. De acordo com a Agência Brasileira de Desenvolvimento Industrial (ABDI), 65,2% das micro e pequenas empresas brasileiras estavam no estágio de digitalização básica ou intermediária em 2023, indicando avanços, mas também desafios na adoção de tecnologias (AGÊNCIA..., 2023). No entanto, a falta de ferramentas acessíveis de gestão de vendas continua sendo um obstáculo para a competitividade desses empreendedores, que frequentemente competem com grandes empresas que possuem sistemas automatizados e bem integrados.

Ademais, a adoção de soluções tecnológicas adequadas é essencial para que essas empresas possam gerenciar estoques, clientes e pedidos de maneira eficiente. Estudos indicam que a digitalização contribui para a redução de erros operacionais e facilita a escalabilidade dos negócios (AGÊNCIA..., 2023). Além disso, a integração de processos permite uma visão mais abrangente do negócio, auxiliando na tomada de decisões estratégicas. Portanto, o Easy Sales surge como uma solução para atender a essa necessidade, oferecendo uma plataforma de gestão de vendas que auxilia pequenos empreendedores a gerenciar seus negócios de forma eficiente. Com funcionalidades que incluem controle de estoque, emissão de relatórios e acompanhamento de vendas, o sistema visa simplificar a operação comercial e aumentar a competitividade dessas empresas em um mercado altamente dinâmico.

1.1 JUSTIFICATIVA

A atividade de vendas é de suma importância no campo da dinâmica econômica, particularmente para pequenos e médios empreendedores que dependem desse setor para gerar receita e promover a criação de empregos. No entanto, a crescente complexidade do mercado, juntamente com a ampla variedade de produtos disponíveis, torna o gerenciamento de operações como estoque, pedidos e atendimento ao cliente um desafio significativo. Na ausência de recursos tecnológicos adequados, esses empreendedores frequentemente se veem utilizando sistemas manuais ou desarticulados, o que compromete sua eficiência e competitividade em comparação com grandes corporações.

Diante dessa situação, a falta de sistemas de gerenciamento eficazes resulta em integração insuficiente entre áreas essenciais de negócios, incluindo vendas, gerenciamento de estoque, faturamento e atendimento ao cliente. Essa desarticulação nos processos não apenas dificulta a capacidade de tomar decisões rápidas e precisas, mas também aumenta a probabilidade de erros operacionais, afetando a produtividade e a sustentabilidade dos negócios. Por outro lado, empresas maiores utilizam tecnologias avançadas para aprimorar seus processos e garantir uma vantagem competitiva notável, colocando, conseqüentemente, empresas menores em relativa desvantagem.

Portanto, a digitalização e implementação de sistemas integrados de gestão surgiram como estratégias vitais para aumentar a competitividade das PMEs. Conforme destacado por (Confederação Nacional da Indústria (CNI), 2022), a adoção de soluções tecnológicas pode agilizar processos, minimizar erros e impulsionar significativamente o potencial de crescimento de pequenos empreendedores. Portanto, o estabelecimento de um sistema de gestão como o Easy Sales é garantido não apenas pela perspectiva de melhorar a eficiência operacional, mas também pela oportunidade de promover maior sustentabilidade e crescimento de longo prazo para essas empresas.

Conseqüentemente, além de centralizar e automatizar funções essenciais, como vendas e gerenciamento de estoque, a proposta do sistema visa aumentar a competitividade de pequenos empreendedores em um cenário de mercado cada vez mais desafiador. Ao oferecer uma solução acessível e flexível, o sistema capacita essas empresas a não apenas sobreviver, mas a prosperar, atendendo com eficácia às demandas atuais e aos requisitos futuros de desenvolvimento sustentável.

1.2 OBJETIVO GERAL

Desenvolver um sistema de gerenciamento de vendas para pequenas e médias empresas para aprimorar as atividades comerciais e a eficiência operacional e, ao mesmo tempo, promover o crescimento sustentável. A principal proposta é oferecer controle personalizado sobre os processos de negócios para atender às necessidades dos usuários. Isso inclui recursos como integração de inventário, automação de vendas e geração de relatórios analíticos, permitindo que os empreendedores obtenham insights sobre suas operações e tomem decisões estratégicas informadas. Desenvolver um sistema de gestão de vendas que melhore a efetividade operacional de pequenos e médios empreendedores, promovendo o crescimento sustentável a longo prazo.

1.3 OBJETIVOS ESPECÍFICOS

- **Otimizar processos de vendas.**
- **Fornecer relatórios analíticos para decisões estratégicas.**
- **Facilitar a negociação de dívidas com clientes.**

2 FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica deste trabalho oferece a base conceitual necessária para compreender os temas centrais, como os sistemas de gestão de vendas e suas implicações nas pequenas e médias empresas (PMEs). Esses sistemas desempenham um papel fundamental na otimização de processos, automação do controle de estoque e gerenciamento de clientes, aspectos essenciais para a competitividade dessas empresas no mercado atual. Além disso, serão explorados estudos de caso que destacam os desafios e os benefícios da adoção de tecnologias digitais por negócios de menor porte, contribuindo para a sustentabilidade e o crescimento dessas empresas.

Ademais, serão discutidas as metodologias ágeis, como o Scrum, amplamente utilizadas no desenvolvimento de software, com foco em sua aplicação na criação de sistemas de gestão de vendas. O uso dessas metodologias permite um desenvolvimento iterativo e flexível, garantindo que o sistema proposto seja ajustado e aprimorado continuamente de acordo com o feedback dos usuários. Dessa forma, a fundamentação teórica não apenas consolida os conhecimentos existentes, mas também identifica oportunidades de inovação que podem ser exploradas para aprimorar a eficiência e a competitividade das PMEs.

2.1 IMPACTO DOS SISTEMAS DE GESTÃO DE VENDAS NAS PMES E AUTÔNOMOS

O uso de sistemas de gestão de vendas é fundamental para aumentar a eficiência e a competitividade das pequenas e médias empresas (PMEs) e autônomos. De acordo com pesquisas recentes, 76% das PMEs no Brasil ainda não utilizam sistemas de gestão adequados, continuando a depender de planilhas e controles manuais. Dessa maneira, essa abordagem aumenta a suscetibilidade a erros, retrabalho e perda de oportunidades estratégicas (CONTÁBEIS, 2024). Portanto, esse cenário limita o potencial de crescimento e impede que os negócios alcancem maior agilidade e precisão nas operações diárias.

Apesar disso, há um crescente interesse em investir em soluções tecnológicas para gestão de negócios. Segundo o (CONTÁBEIS, 2024), 90% das PMEs manifestaram intenção de adotar sistemas de gestão de vendas nos próximos anos. Esse movimento é impulsionado pela necessidade de melhorar a competitividade em um mercado cada vez mais digitalizado, onde a automação e a integração de processos são essenciais para manter uma operação eficiente e sustentável.

2.1.1 Benefícios dos Sistemas de Gestão de Vendas

A adoção de sistemas de gestão de vendas oferece diversos benefícios para PMEs e autônomos, principalmente em termos de eficiência operacional. Nesse sentido, a automação de tarefas como controle de estoque, emissão de notas fiscais e acompanhamento financeiro permite que os empreendedores se concentrem em decisões estratégicas e no crescimento do negócio, em vez de desperdiçar tempo em atividades manuais repetitivas (CONTÁBEIS, 2024). Além disso, a otimização desses processos reduz significativamente o tempo de resposta em situações críticas, como a reposição de estoque ou a gestão de pedidos.

Dessa forma, consequentemente, outro benefício importante é a melhoria na tomada de decisões. Com acesso a dados em tempo real sobre as operações, os gestores podem analisar de forma precisa o desempenho do negócio e identificar rapidamente áreas que necessitam de ajustes ou melhorias. Isso possibilita decisões mais assertivas e fundamentadas, o que, por sua vez, contribui para a sustentabilidade e o crescimento das empresas. Portanto, a automação ajuda a reduzir custos ao diminuir a margem de erro humano e eliminar redundâncias operacionais, resultando em economia de recursos e aumento da lucratividade a longo prazo (VALENTE, 2004).

2.1.2 Estudos de Caso em PMEs e Autônomos

Estudos de caso envolvendo a implementação de sistemas ERP (Enterprise Resource Planning) em PMEs destacam tanto os benefícios quanto os desafios enfrentados por empresas de menor porte. Um exemplo é uma PME do setor de construção civil que enfrentou dificuldades iniciais, como resistência à mudança e limitações de recursos, ao adotar um sistema ERP. Apesar dos desafios, o estudo mostrou que, após a implementação, a empresa obteve melhorias significativas em termos de eficiência operacional e integração entre departamentos (VALENTE, 2004).

Logo, as melhorias no fluxo de informações entre os setores possibilitaram uma comunicação mais eficaz, permitindo decisões mais rápidas e coordenadas. No entanto, a adoção de sistemas de gestão em PMEs requer um planejamento cuidadoso para superar barreiras culturais e financeiras. Esses casos demonstram que, apesar das dificuldades iniciais, os benefícios a longo prazo compensam os esforços, contribuindo para o aumento da competitividade e sustentabilidade das empresas no mercado atual.

2.1.3 Desafios Específicos e Soluções

Os principais obstáculos que as PMEs enfrentam ao adotar sistemas de gestão de vendas incluem restrições orçamentárias e limitações de infraestrutura tecnológica. Pequenas empresas geralmente têm menos recursos disponíveis para grandes investimentos tecnológicos, o que pode atrasar a adoção desses sistemas. Além disso, a falta

de conhecimento técnico pode gerar resistência à mudança entre os colaboradores (CONTÁBEIS, 2024).

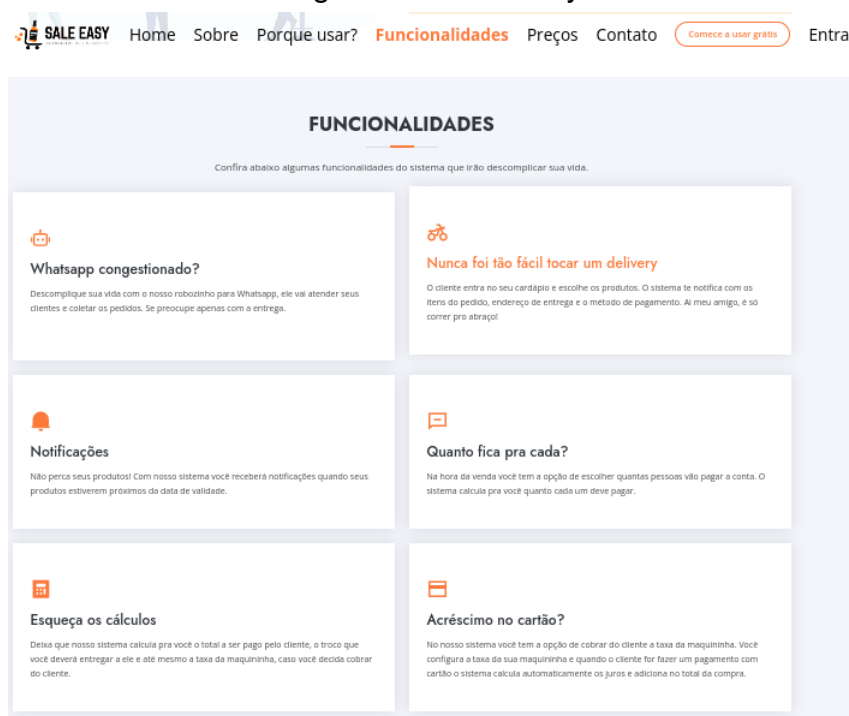
Soluções práticas para esses desafios incluem a adoção de plataformas modulares ou SaaS (Software as a Service), que permitem que as empresas paguem apenas pelo que utilizam e escalem suas operações conforme necessário. Outro caminho é a implementação de sistemas de código aberto, que podem ser mais acessíveis financeiramente (CONTÁBEIS, 2024) (VALENTE, 2004).

2.2 TRABALHOS CORRELATOS

Os sistemas de gestão de vendas *Sales Easy*, *Egestor* e *Bling* foram escolhidos para comparação devido à sua popularidade e ampla adoção por pequenos e médios vendedores, bem como autônomos. Além disso, foram selecionados por sua semelhança com o projeto proposto, permitindo uma análise mais direta e relevante das funcionalidades e características que são importantes para os usuários-alvo. Essa escolha estratégica facilitará a identificação de lacunas e oportunidades de melhoria em relação ao projeto proposto conforme tabela 1.

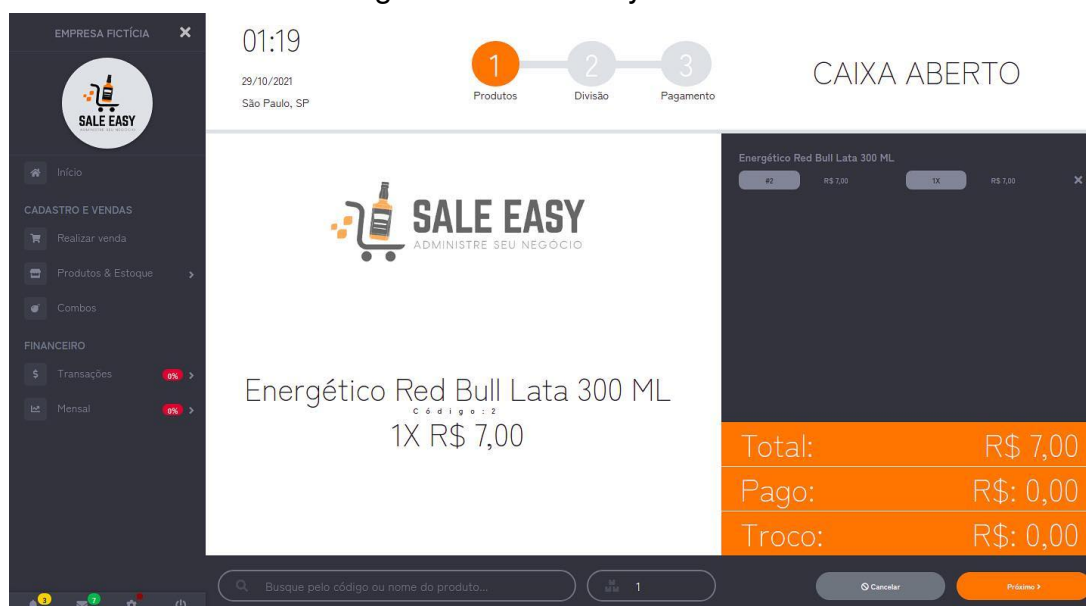
Dessa forma, o *Sales Easy*, na figura 1, é reconhecido por sua interface simplificada e foco direto na gestão de vendas online, embora ofereça recursos intuitivos para acompanhamento de pedidos e controle de clientes, sua funcionalidade pode ser considerada limitada para necessidades mais complexas de gestão de negócios.

Figura 1 – Sale Easy



Fonte: <<https://saleeasy.com.br/>>

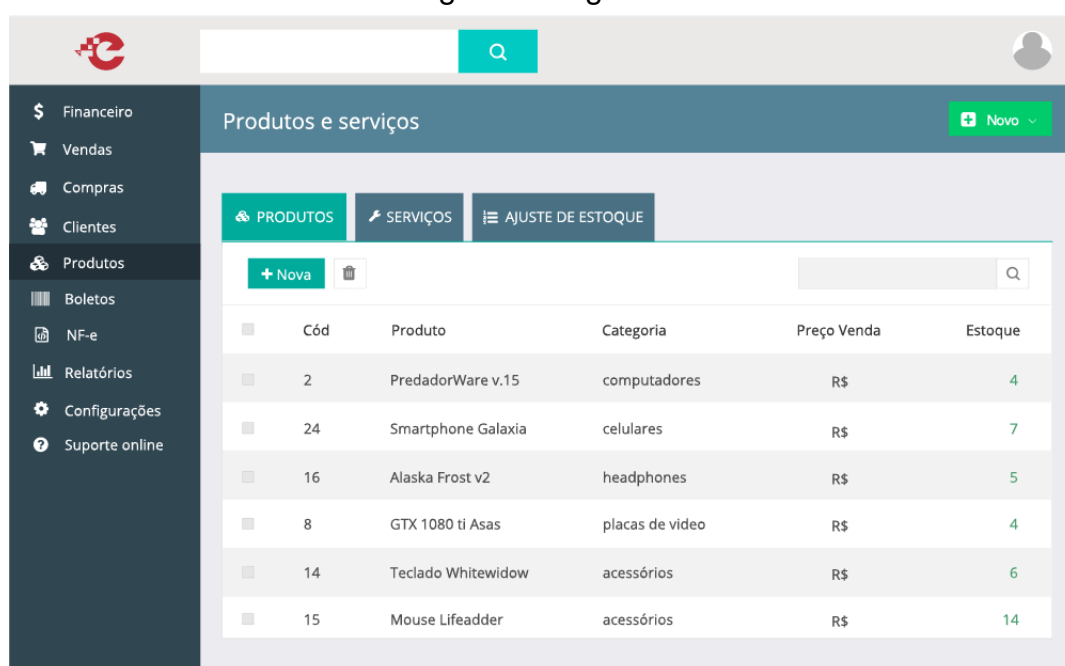
Figura 2 – Sale Easy Vendas



Fonte: <<https://saleeasy.com.br/>>

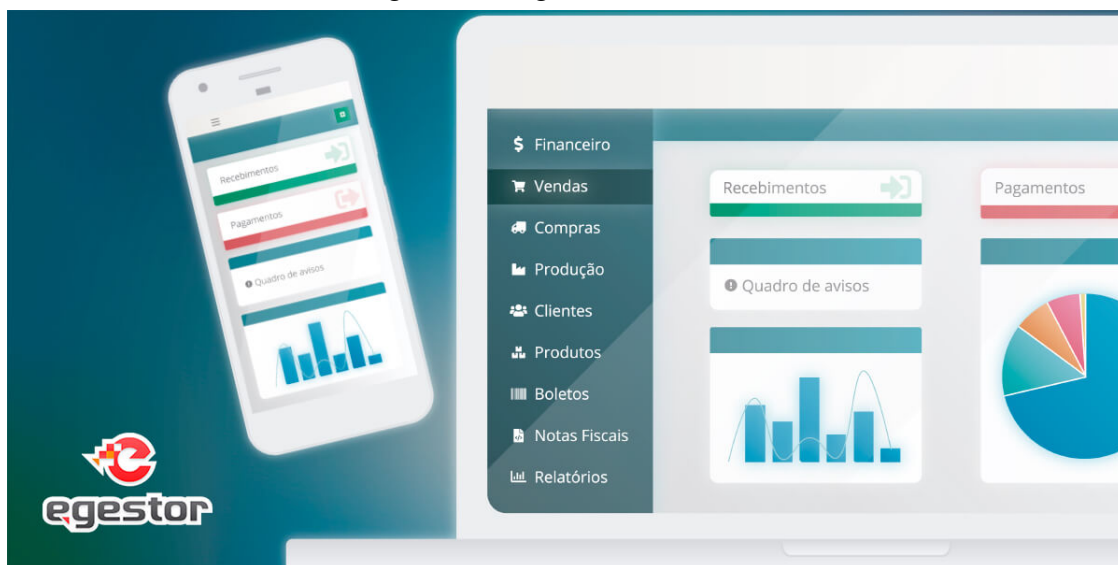
Por outro lado, o *Egestor*, apresentado na imagem 3, destaca-se por sua abordagem abrangente, fornecendo uma ampla gama de ferramentas para gestão de estoque, finanças e relacionamento com clientes. Portanto, sua capacidade de integrar múltiplas áreas de negócios em uma única plataforma oferece uma solução completa e holística, particularmente adequada para pequenas e médias empresas que buscam uma solução centralizada para suas operações.

Figura 3 – Egestor



Fonte: <<https://egestor.com.br/>>

Figura 4 – Egestor Relatórios



Fonte: <<https://egestor.com.br/>>

Por fim, o *Bling* apresentado a seguir na figura 5, se destaca por sua robustez e flexibilidade, oferecendo uma gama diversificada de recursos que incluem gestão de estoque, emissão de notas fiscais na figura 6, controle financeiro e muito mais. Ademais, sua capacidade de personalização e integração com outros sistemas torna-o uma escolha popular entre empresas de diversos portes, proporcionando uma solução adaptável às necessidades específicas de cada negócio.

Figura 5 – Bling venda

The image shows the Bling venda interface for a sales order. The header includes the Bling logo, 'Lojas Bling', a 'Nova versão' toggle, and a search bar. The main section is titled 'Venda - Pedido 349' and contains a table with columns: 'Produto', 'Qtde. (1,00)', 'Preço', 'Sub Total', and 'Ações'. The table lists one item: '- Camisa Cor:Branca,Tamanho:M' with a quantity of 1 and a price of 50,00. The 'Sub Total' is 50,00. The 'Ações' column includes icons for print, edit, and delete. On the right, there are buttons for 'Pedido 349' and 'Pedido 348'. At the bottom, there are buttons for 'Excluir venda' and 'Finalizar venda', and a 'Total 50,00' display.

Fonte: <<https://bling.com.br/>>

Figura 6 – Bling Cadastrar Vendas

The screenshot shows the Bling 'Venda - Pedido 349' interface. The top bar includes the Bling logo, 'Lojas Bling', and a 'Nova versão' toggle. The main header is 'Venda - Pedido 349' with a search bar 'Buscar venda - (Alt+P)' and a 'Nova Venda' button. The interface is divided into a sidebar and a main content area. The sidebar has tabs for 'Produto', 'Cliente', and 'Pagamento'. The main content area displays a table with columns: 'Produto', 'Qtde. (1,00)', 'Preço', 'Sub Total', and 'Ações'. A single row is visible for '- Camisa' with a price of 50,00. The bottom right shows a 'Total' of 50,00. On the left, there are sections for 'Lista de preço' and 'Leitor de código de barras'.

Fonte: <<https://bling.com.br/>>

Em contraste com os sistemas comerciais estabelecidos, o projeto proposto, denominado Easy Sales, se posiciona como uma alternativa inovadora e personalizada. A tabela comparativa 1 apresenta uma análise detalhada das principais funcionalidades dos sistemas Sales Easy, Egestor, Bling e o Easy Sales, permitindo identificar as lacunas e vantagens de cada solução. Ao focar na simplicidade de uso, acessibilidade financeira e suporte ao cliente, o Easy Sales busca preencher as deficiências encontradas nos sistemas existentes, adaptando-se melhor às necessidades de pequenos e médios vendedores e autônomos. Essa comparação ressalta o potencial do Easy Sales em oferecer uma solução mais eficiente e flexível para a gestão de vendas, alinhada aos desafios e demandas do mercado atual.

Tabela 1 – Tabela Comparativa de Softwares

Projetos	Sale Easy	Egestor	Bling	Easy Sales
Relatório completo da loja	✓	✓	✓	✓
Controle de estoque	✓	✓	✓	✓
Monitorar devedores	✗	✓	✗	✓
Alerta	✓	✓	✓	✓
Boleto	✗	✓	✓	✓
Controle de comissões	✗	✓	✗	✗
Integração facilitada	✗	✗	✓	✓
Custo-benefício	✓	✗	✗	✓

Fonte: De autoria própria

3 METODOLOGIA

A metodologia deste trabalho descreve os métodos e as ferramentas utilizadas para o desenvolvimento do sistema de gestão de vendas proposto. Nesta seção, serão apresentados os processos seguidos para o levantamento de requisitos, o planejamento, o desenvolvimento, e a validação do sistema. Para garantir a entrega de uma solução eficiente e adaptada às necessidades dos usuários finais, foram utilizadas abordagens ágeis, como o Scrum, que permite a realização de entregas incrementais e o constante refinamento do projeto com base no feedback dos stakeholders.

Dessa forma, o desenvolvimento foi dividido em etapas bem definidas, começando pelo levantamento de requisitos, seguido pela prototipação e implementação do sistema. Cada etapa foi planejada com o objetivo de assegurar que o produto final atendesse às demandas específicas das PMEs e vendedores autônomos. Ao longo do processo, foram utilizados diferentes ferramentas e tecnologias, tanto para o front-end quanto para o back-end, além de metodologias de validação com usuários para garantir a usabilidade e a eficácia da solução.

3.1 VISÃO GERAL

A metodologia de pesquisa adotada nesse trabalho integra técnicas ágeis de desenvolvimento de software e coleta de dados, organizadas em quatro fases principais: planejamento, desenvolvimento, prototipagem e validação. A metodologia Scrum permite a segmentação do projeto em sprints, aprimorando assim o fornecimento incremental de funcionalidades. As tarefas de cada sprint foram definidas e priorizadas dentro do Product Backlog, garantindo o alinhamento com as necessidades e expectativas das partes interessadas durante todo o processo.

Ademais, na fase de desenvolvimento, o foco estará na implementação de uma solução integrada, que inclui as principais funcionalidades do sistema de gestão de vendas. Tecnologias específicas, descritas posteriormente, serão empregadas para permitir que a aplicação evolua de forma ágil e flexível, atendendo aos requisitos previamente levantados. A combinação dessas ferramentas possibilita uma adaptação rápida às mudanças e a implementação eficaz das funcionalidades previstas.

Nesse sentido, antecedente a implementação, foram criados protótipos de alta fidelidade que passarão por testes com usuários selecionados, permitindo ajustes contínuos ao longo do ciclo de desenvolvimento. A validação final do software será realizada por meio de testes de usabilidade e feedbacks de clientes reais, assegurando que a solução proposta resolva de maneira eficiente os problemas identificados inicialmente, e permitindo refinamentos necessários antes da entrega final.

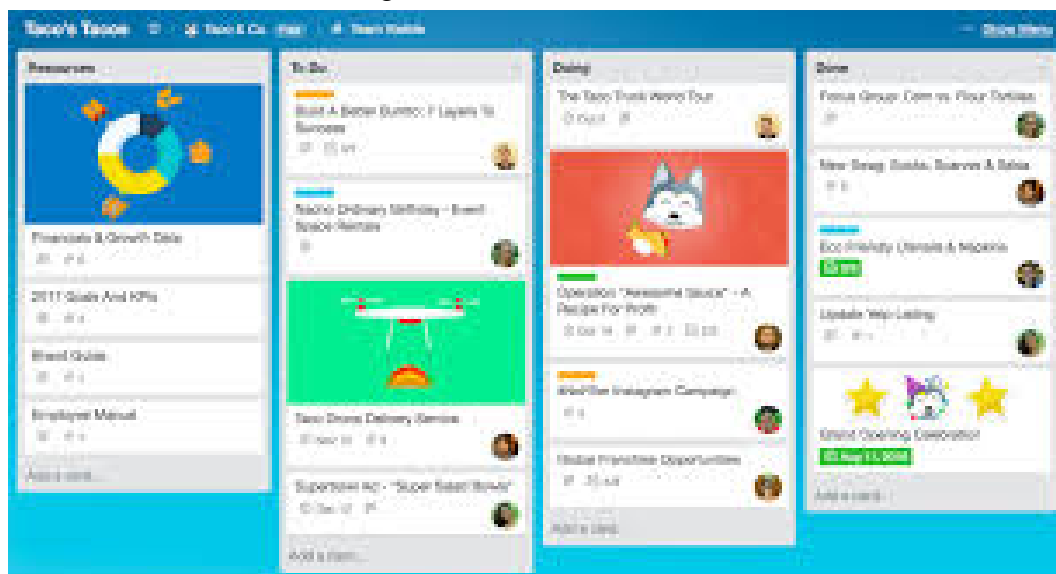
3.2 FERRAMENTAS DE GERENCIAMENTO DE PROJETO

As ferramentas de gerenciamento de projetos desempenham um papel crucial no desenvolvimento de sistemas, oferecendo uma estrutura organizacional robusta e recursos que facilitam a coordenação de tarefas, o monitoramento de progresso e a colaboração entre os membros da equipe, sendo fundamental para o controle de tarefas e bugs. Além disso, a prática de esboçar experiências do usuário, como sugerido por (BUXTON, 2007), é essencial para garantir que o design atenda às expectativas dos usuários. Ferramentas eficazes, como o *Jira*, *Trello* e *Monday* disponibilizam uma ampla gama de funcionalidades que simplificam a gestão de projetos, abrangendo desde a criação e atribuição de tarefas até o rastreamento de bugs e a gestão de sprints. Essas ferramentas também promovem a transparência no processo de desenvolvimento, permitindo que todos os envolvidos acompanhem o status do projeto em tempo real.

A seleção de um software de gerenciamento de projetos exige uma avaliação completa das várias alternativas disponíveis, cada uma apresentando vantagens e desvantagens distintas. Para os propósitos deste projeto, três ferramentas específicas foram examinadas: A seleção de um software de gerenciamento de projetos requer uma avaliação completa de várias alternativas disponíveis, cada uma apresentando vantagens e desvantagens distintas. Para os fins deste projeto, três ferramentas específicas foram examinadas:

- **Trello:** O *Trello* é uma ferramenta de gerenciamento visual que utiliza quadros e cartões para organizar tarefas. É ideal para projetos simples e equipes pequenas, permitindo a colaboração através de checklists, etiquetas e comentários. Sua interface intuitiva e a facilidade de uso tornam o Trello uma ótima escolha para equipes que buscam uma solução acessível e fácil de implementar.

Figura 7 – Software Trello.



Fonte: <<https://www.trello.com/>>

- **Jira:** O *Jira* é uma ferramenta robusta de gerenciamento de projetos voltada principalmente para equipes de desenvolvimento de software. É amplamente utilizada em ambientes que adotam metodologias ágeis, como Scrum e Kanban, oferecendo recursos avançados para planejamento de sprints, gerenciamento de backlogs e geração de relatórios detalhados. Por essas características, o Jira é ideal para projetos mais complexos.

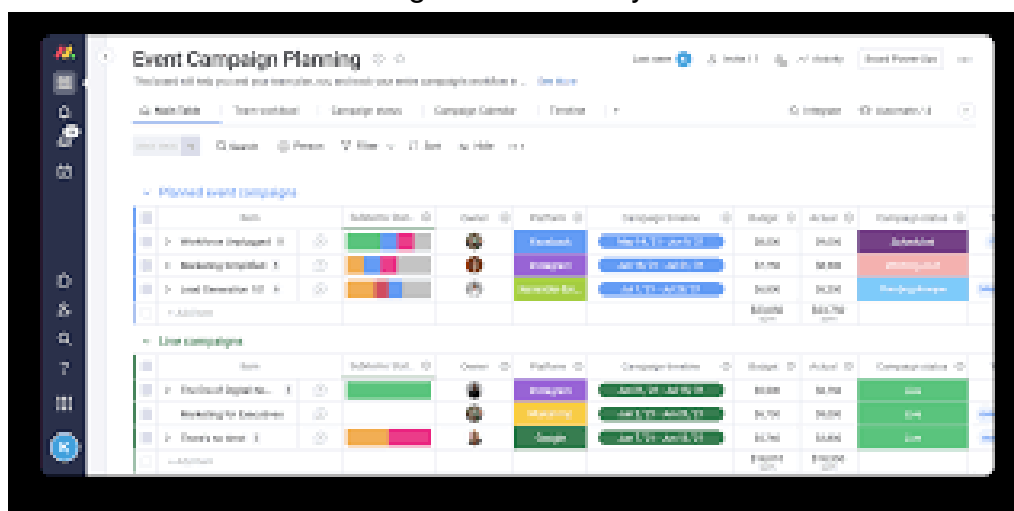
Figura 8 – Software Jira.



Fonte: <<https://www.jira.com/>>

- **Monday:** O *Monday* é uma plataforma flexível de gerenciamento de projetos que permite a personalização de fluxos de trabalho para atender às necessidades de qualquer equipe. Com recursos visuais que facilitam a gestão de tarefas e o monitoramento do progresso, o Monday oferece visualizações em quadros Kanban, cronogramas e gráficos de Gantt, além de integrações com outras ferramentas, como Slack e Google Drive. É uma solução eficaz tanto para pequenas equipes quanto para organizações maiores que buscam agilidade e colaboração em tempo real.

Figura 9 – Monday.



Fonte: <<https://www.monday.com/>>

Tabela 2 – Tabela comparativa das ferramentas de gerenciamento de projetos

Funcionalidades	Trello	Jira	Asana
Interface intuitiva e fácil de usar	✓	✗	✓
Adequado para metodologias ágeis (Scrum, Kanban)	✓	✓	✓
Visualização em quadros/Kanban	✓	✓	✓
Curva de aprendizado rápida	✓	✗	✗
Funcionalidades robustas para desenvolvimento de software	✗	✓	✗
Custo benefício para pequenas equipes	✓	✗	✓

Fonte: De Autoria Própria

A tabela 2 apresenta uma comparação detalhada das principais funcionalidades das ferramentas avaliadas. Após analisar os benefícios de cada uma, a ferramenta de gerenciamento de projetos escolhida para o desenvolvimento do Easy Sales foi o *Jira*. Essa escolha se baseia na sua robustez e na capacidade de gerenciar projetos complexos, especialmente em ambientes que utilizam metodologias ágeis, como o

Scrum. O Jira oferece recursos avançados para o planejamento de sprints, controle de backlog e geração de relatórios detalhados, o que facilita o monitoramento do progresso e o ajuste contínuo das entregas.

Embora o Jira tenha uma curva de aprendizado mais acentuada em comparação com outras opções, como o Trello, sua flexibilidade e funcionalidades voltadas ao desenvolvimento de software o tornam ideal para o Easy Sales. A ferramenta não apenas permite uma melhor organização das tarefas, mas também oferece integrações que facilitam a colaboração entre os membros da equipe. Assim, o Jira proporciona uma supervisão mais detalhada e profissional do avanço do projeto, garantindo que o Easy Sales seja desenvolvido de maneira eficiente e alinhada às necessidades do mercado.

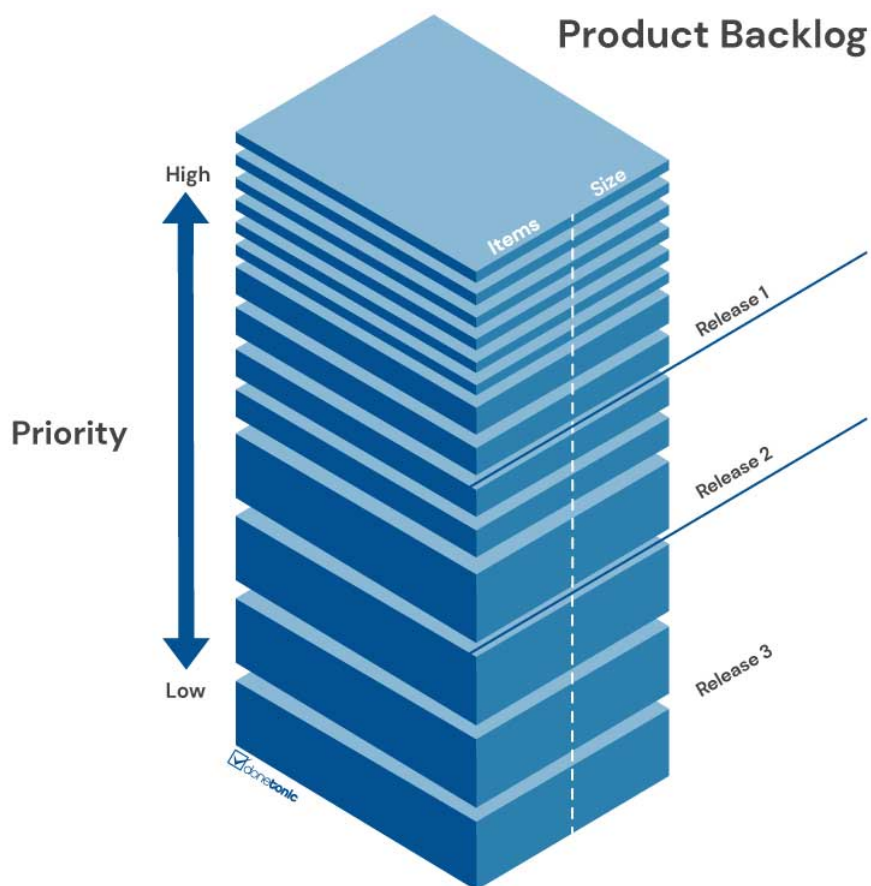
3.3 BACKLOG

O backlog, ou “lista de pendências”, é uma parte essencial do processo de gerenciamento de projetos no Jira e em muitas outras ferramentas de desenvolvimento de software. Ele funciona como um repositório centralizado de todas as tarefas, funcionalidades e requisitos que ainda precisam ser realizados ao longo do ciclo de vida do projeto, organizado de acordo com prioridades e categorias (COHN, 2005). No contexto do desenvolvimento de sistemas, o backlog é especialmente valioso para manter o foco na entrega de valor ao cliente, adicionando funcionalidades com base nas necessidades do cliente e priorizando-as de acordo com seu impacto e urgência (VERHEYEN, 2020).

Existem diferentes tipos de backlog, como o backlog do produto e o backlog de sprint, cada um com sua própria finalidade e contexto de aplicação, desempenhando um papel crucial no planejamento e na execução de projetos, garantindo que as equipes foquem nas atividades mais importantes e entreguem valor aos clientes de forma eficiente (VERHEYEN, 2020).

O product backlog é uma lista priorizada de todas as funcionalidades, requisitos e mudanças necessárias para o produto, servindo como a fonte definitiva de direcionamento para o desenvolvimento, conforme descrito no Guia Scrum (SCRUM.ORG, 2022). Ao longo do processo de desenvolvimento, o product backlog está sujeito a mudanças contínuas para garantir que o produto se mantenha relevante, competitivo e útil (SCHWABER, 2014).

Figura 10 – Backlog do produto



Fonte: <https://donetonic.com/wp-content/uploads/2022/08/Donetonic-Product-Backlog.jpg>

O Product Owner, em colaboração com outras partes interessadas, é responsável por definir os itens do product backlog e garantir sua ordenação correta por prioridade, como ilustrado na figura acima 10. Dessa forma, os itens de maior prioridade no product backlog, localizados no topo da lista, devem ser mais detalhados e claros, pois serão os primeiros a serem implementados durante o desenvolvimento (VERHEYEN, 2020).

3.4 SPRINT

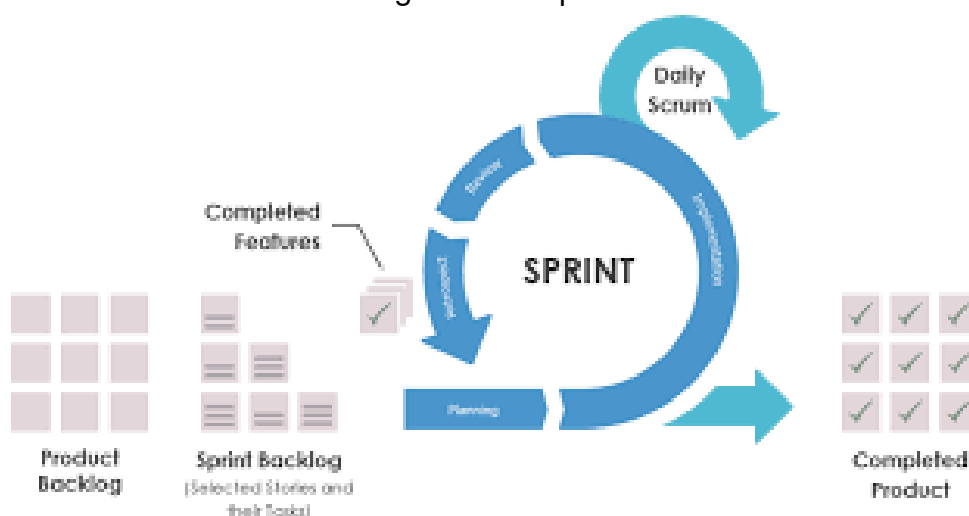
A sprint é uma iteração curta e focada no desenvolvimento de software dentro do framework Scrum, geralmente com duração de duas a quatro semanas (SCHWABER, 2014). Durante a sprint, a equipe de desenvolvimento se concentra em completar um conjunto específico de tarefas previamente selecionadas a partir do backlog do produto. O objetivo é entregar um incremento funcional do produto, pronto para ser revisado e

testado ao final da sprint (VERHEYEN, 2020). Dessa forma, durante o planejamento da sprint, a equipe revisa e refina o sprint backlog, adicionando detalhes necessários e estimativas de esforço para cada item, garantindo uma compreensão clara do trabalho a ser realizado e ajudando a equipe a se comprometer com o que pode ser realizado durante a sprint (SCHWABER, 2014).

O sprint backlog é uma lista priorizada de itens de trabalho que uma equipe de desenvolvimento planeja completar durante uma sprint específica (SCHWABER, 2014). Esses itens são retirados do backlog do produto e representam as funcionalidades, melhorias ou correções que foram selecionadas para serem implementadas na próxima iteração do ciclo de desenvolvimento (VERHEYEN, 2020). A priorização é essencial no sprint backlog, garantindo que os itens mais importantes ou de maior valor para o cliente sejam abordados primeiro, maximizando assim o retorno sobre o investimento (COHN, 2005).

Uma prática eficaz no gerenciamento do sprint backlog é manter a transparência e a colaboração, envolvendo toda a equipe no processo de refinamento e priorização dos itens. Isso não apenas aumenta o senso de propriedade e responsabilidade, mas também promove uma compreensão compartilhada dos objetivos da sprint (VERHEYEN, 2020). À medida que os itens são concluídos durante a sprint, o progresso é frequentemente revisado durante as reuniões diárias do Scrum, e qualquer ajuste necessário no sprint backlog pode ser feito durante a sprint (SCHWABER, 2014). Dessa forma, o sprint backlog serve como uma ferramenta dinâmica que guia o trabalho da equipe de desenvolvimento, proporcionando um caminho claro e adaptável para a conclusão dos objetivos.

Figura 11 – Sprint



Fonte: <<https://www.cybermedian.com/pt/wp-content/uploads/sites/11/2022/01/0mR-c0N09YvJdy5SP.png>>

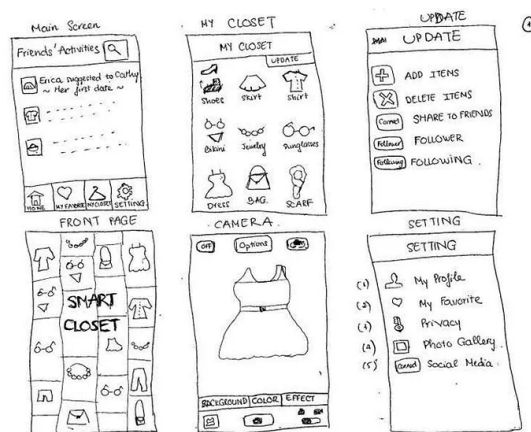
3.5 PROTOTIPAÇÃO

A prototipação é um processo essencial no desenvolvimento de sistemas e design de produtos, permitindo que as equipes criem representações preliminares de um produto ou sistema antes de sua implementação completa (GOODWIN, 2009). O objetivo principal da prototipação é validar conceitos, testar funcionalidades e obter feedback antecipado dos usuários para garantir que o produto final atenda às suas necessidades e expectativas (BUXTON, 2007).

Existem diferentes tipos de protótipos, incluindo:

- **Protótipos de Papel:** Representações simples e rápidas de um design, frequentemente utilizados para validar conceitos iniciais e fluxos de usuário.

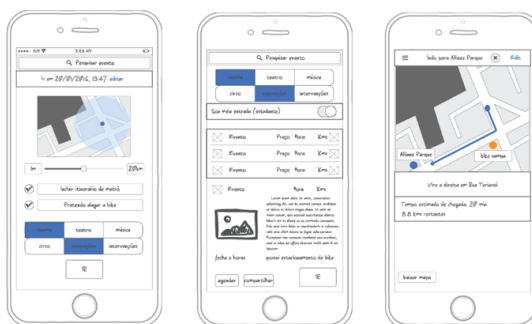
Figura 12 – Protótipos de Papel



Fonte: <<https://brasil.uxdesign.cc/hackeando-a-prototipagem-29b8abab5f01>>

- **Protótipos de Alta Fidelidade:** Versões mais detalhadas e interativas do produto, que simulam de forma mais precisa a aparência e o funcionamento do produto final.

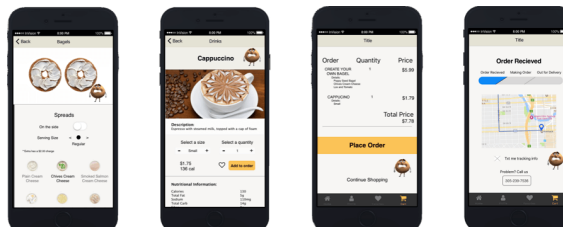
Figura 13 – Protótipos de Alta Fidelidade



Fonte: <<http://designr.com.br/como-e-o-processo-da-prototipacao-rapida/>>

- **Protótipos Funcionais:** Protótipos que são desenvolvidos com funcionalidades básicas e permitem testar a viabilidade técnica de um sistema.

Figura 14 – Protótipos Funcionais



Fonte: <<https://www.edrawsoft.com/pt/prototyping.html>>

No contexto das ferramentas de gerenciamento de projetos, muitas vezes é utilizado o software para criar protótipos digitais, o que pode incluir a criação de wireframes, mockups e protótipos interativos. Ferramentas como *Axure*, *Figma* e *Adobe XD* são amplamente utilizadas para essas finalidades, oferecendo recursos que permitem a criação e o teste de protótipos de forma eficiente e colaborativa. A prototipação eficaz pode levar a um desenvolvimento mais ágil e iterativo, reduzindo o risco de falhas e garantindo que o produto final seja bem alinhado com as expectativas dos usuários (BUXTON, 2007).

Para este projeto, utilizaremos o *Figma* como ferramenta principal de prototipação. O *Figma* é uma plataforma de design colaborativo baseada na web que permite criar protótipos interativos e compartilhá-los facilmente com membros da equipe e stakeholders. Sua interface intuitiva e recursos avançados facilitam a criação de protótipos detalhados e a realização de testes de usabilidade.

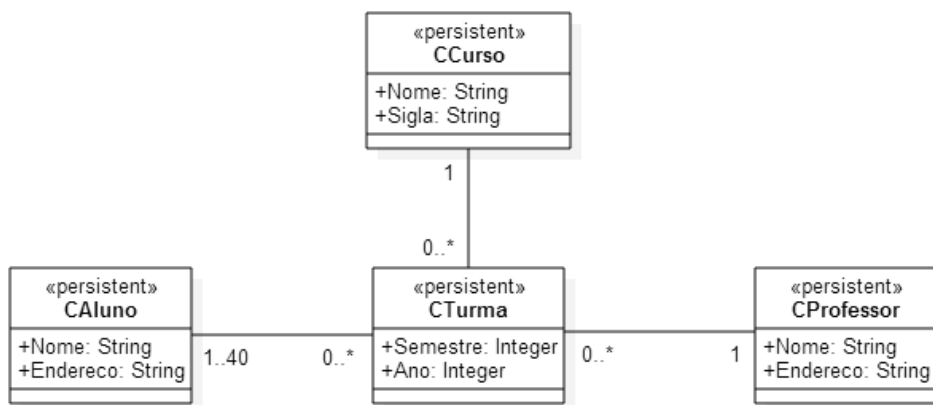
3.6 PRINCIPAIS DIAGRAMAS

Os diagramas UML (Unified Modeling Language) são cruciais para a modelagem e documentação de sistemas complexos, pois oferecem uma visão visual e estruturada que facilita a compreensão e o planejamento do projeto. Utilizando diagramas como o de classe e o de casos de uso, é possível representar de forma clara a estrutura do sistema e as interações entre seus componentes e usuários. Sem esses diagramas, o desenvolvimento pode enfrentar uma série de desafios, como falta de clareza na comunicação entre a equipe, dificuldades na identificação de requisitos e problemas de integração entre diferentes partes do sistema. A ausência de uma representação visual adequada pode levar a erros de design, retrabalho e atrasos no cronograma, prejudicando a eficiência e a qualidade do projeto. Portanto, a utilização de diagramas UML é fundamental para garantir uma abordagem organizada e eficaz no desenvolvimento de sistemas, minimizando riscos e promovendo um entendimento comum entre todos os envolvidos (FOWLER, 2003).

Os diagramas UML são ferramentas essenciais na modelagem de sistemas, proporcionando uma representação visual clara e estruturada dos diferentes aspectos de um projeto. Dentre os vários tipos de diagramas UML, o diagrama de classe se destaca por sua capacidade de detalhar a estrutura do sistema. Ele descreve as classes envolvidas, seus atributos, métodos e as relações entre elas, como associações, heranças e dependências. Esse diagrama é crucial para entender a arquitetura do sistema e facilitar o design e a implementação orientada a objetos, permitindo que desenvolvedores visualizem e planejem a estrutura do código de forma eficiente.

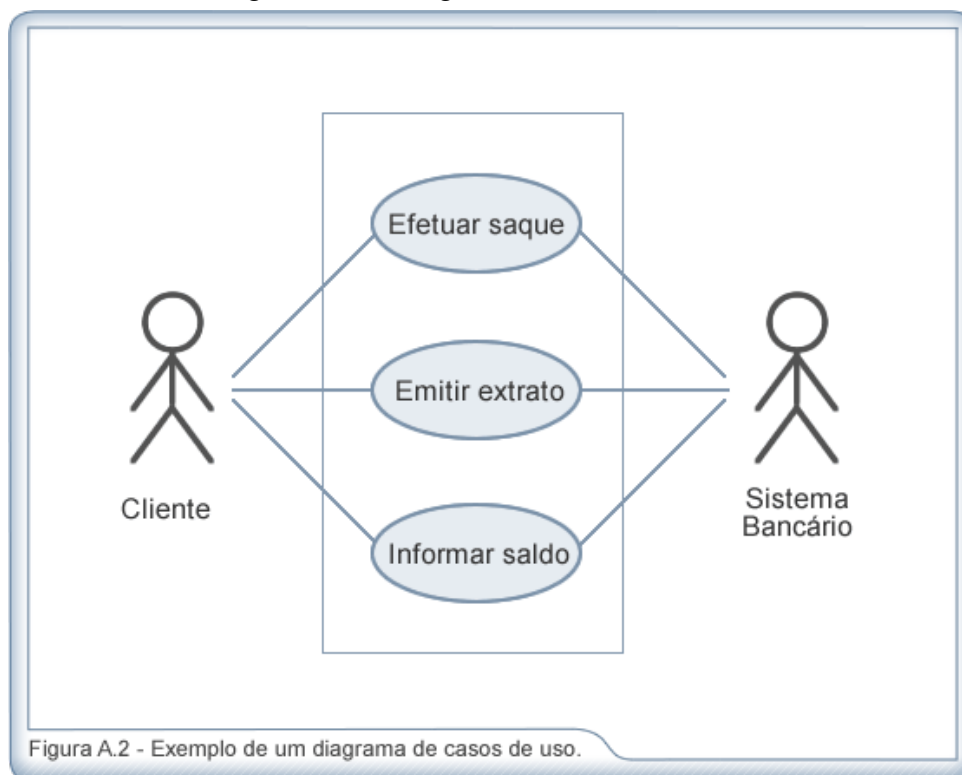
Já o diagrama de casos de uso foca na interação entre os usuários e o sistema. Ele ilustra as funcionalidades que o sistema deve oferecer e como diferentes atores (usuários ou outros sistemas) interagem com essas funcionalidades. Através desse diagrama, é possível identificar os requisitos do sistema e como estes serão atendidos, ajudando na definição dos casos de uso que guiarão o desenvolvimento. Enquanto o diagrama de classe oferece uma visão estática da estrutura, o diagrama de casos de uso proporciona uma visão dinâmica das interações e dos comportamentos esperados, permitindo que todas as partes envolvidas compreendam os objetivos e funcionalidades do sistema.

Figura 15 – Diagrama de Classe



Fonte: <<https://www.researchgate.net/publication/328781141/figure/fig1/AS:865127435882496@1583273862131/Figura-4-Diagrama-de-classes-do-sistema-escolar-exemplo.ppm>>

Figura 16 – Diagrama de Caso de Uso



Fonte: <<http://uab.ifsul.edu.br/tsiad/conteudo/modulo5/dao/ua/1/2.html>>

3.7 FORMA DE VALIDAÇÃO COM CLIENTES

Antes de desenvolver o protótipo do *Easy Sales*, foi fundamental entender as reais necessidades dos potenciais usuários, principalmente microempreendedores e pequenas empresas. Para isso, optou-se por realizar um processo preliminar de validação do conceito por meio de entrevistas. Essas entrevistas foram projetadas para identificar os principais desafios enfrentados na gestão de vendas, controle de estoque e acompanhamento financeiro, permitindo que o sistema fosse ajustado de acordo com as expectativas do mercado-alvo. A seguir, detalham-se as entrevistas realizadas com empreendedores de diferentes setores.

3.7.1 Entrevistas com Empreendedores

Para validar a proposta do *Easy Sales*, foram realizadas entrevistas com três empreendedores, com o objetivo de identificar as principais dificuldades enfrentadas na gestão de vendas e controle de estoque. A seguir, estão os detalhes das entrevistas:

- **Entrevistado 1 (Dona de loja de roupas)**

- **Setor:** Varejo de vestuário

- **Problemas mencionados:**

- * Dificuldade em acompanhar as vendas diárias, especialmente durante períodos de alta demanda.
 - * Falta de organização no controle de estoque.
 - * Necessidade de relatórios fáceis para entender o volume de vendas e preferências dos clientes.

- **Expectativas em relação ao *Easy Sales*:** Automatizar o controle de estoque e gerar relatórios de vendas diários de maneira simples.

- **Feedback sobre o protótipo:** Considerou o protótipo interessante, mas sugeriu personalizar os relatórios de acordo com períodos específicos.

- **Entrevistado 2 (Produtos diversos)**

- **Setor:** Geral

- **Problemas mencionados:**

- * Dificuldade em registrar vendas com diferentes formas de pagamento.
 - * Problemas no controle de gastos.
 - * Necessidade de previsões de gastos com base em compras e vendas anteriores.

- **Expectativas em relação ao *Easy Sales*:** Controlar a entrada e saída de produtos e registrar as diversas formas de pagamento.

- **Feedback sobre o protótipo:** Ficou satisfeita com o controle de estoque, mas sugeriu a inclusão de uma funcionalidade para organizar as formas de pagamento.

- **Entrevistado 3 (Microempreendedora de acessórios pessoais)**

- **Setor:** Serviços de Manufatura

- **Problemas mencionados:**

- * Dificuldade em manter um histórico de vendas de serviços.
 - * Falta de clareza sobre o lucro líquido mensal, considerando custos variáveis.
 - * Necessidade de relatórios de clientes e serviços prestados.

- **Expectativas em relação ao *Easy Sales*:** Gerar relatórios financeiros e proporcionar uma visão clara dos serviços prestados.

- **Feedback sobre o protótipo:** Gostou da interface, mas solicitou maior personalização nos relatórios financeiros.

Com base nas entrevistas, foi possível validar que o *Easy Sales* está no caminho certo ao focar na automação de processos e emissão de relatórios. No entanto, algumas melhorias foram sugeridas, como a personalização dos relatórios e a inclusão de funcionalidades específicas para certos setores, como a gestão de formas de pagamento. Essas interações contribuíram significativamente para a evolução do protótipo e a adaptação da solução às necessidades reais do mercado.

3.7.2 Validações Adicionais

Além das entrevistas iniciais realizadas com os empreendedores, foram conduzidas validações complementares para garantir que o *Easy Sales* esteja alinhado às necessidades dos usuários antes do seu lançamento oficial. Embora o sistema ainda não tenha sido disponibilizado para uso prático devido à ausência de *deploy*, foram realizadas análises internas para validar a viabilidade do projeto e antecipar possíveis ajustes. As principais validações incluíram:

- **Revisões Heurísticas:** Foram conduzidas análises baseadas em princípios de usabilidade para garantir que a interface esteja intuitiva e que os fluxos de navegação sejam coerentes com as expectativas do usuário.
- **Simulação de Cenários de Uso:** Com base nos casos de uso levantados, foram executados testes manuais para simular fluxos de cadastro, vendas e controle de estoque, garantindo a coerência dos processos.
- **Análise de Compatibilidade:** O sistema foi testado em diferentes navegadores e dispositivos para garantir a responsividade e compatibilidade com múltiplas plataformas.
- **Validação de Requisitos:** As funcionalidades desenvolvidas foram comparadas com os requisitos identificados nas entrevistas, assegurando que o sistema atenda às necessidades dos usuários conforme esperado.

Essas validações foram fundamentais para antecipar ajustes na interface e na lógica de negócio antes da disponibilização do sistema para testes com usuários reais. A etapa seguinte consistirá na implementação do *deploy* para permitir a coleta de feedbacks diretos dos empreendedores.

3.7.2.1 Revisão Heurística de Usabilidade

Tabela 3 – Revisão Heurística de Usabilidade

Heurística	Problema Identificado	Correção Implementada
Visibilidade do status do sistema	Não havia indicador de carregamento ao processar vendas	Adicionado indicador de progresso nas requisições
Consistência e padronização	Botões de ação estavam desalinhados em diferentes telas	Padronização dos componentes visuais
Controle e liberdade do usuário	Não havia botão para cancelar operações no cadastro de produtos	Adicionado botão de cancelamento para evitar ações indesejadas

Fonte: De Autoria Própria

3.7.2.2 Testes de Simulação de Cenários de Uso

Tabela 4 – Testes de Simulação de Cenários de Uso

Cenário	Ações do Usuário	Resultado Esperado	Resultado Obtido
Cadastro de Produto	Inserção de dados e submissão do formulário	Produto registrado corretamente no banco de dados	Sucesso
Venda de Produto	Seleção de produtos, inserção do cliente e finalização da venda	Redução do estoque e atualização do histórico de vendas	Sucesso
Relatório de Vendas	Acesso ao relatório de vendas do mês	Exibição correta dos dados filtrados	Sucesso

Fonte: De Autoria Própria

3.7.2.3 Testes de Compatibilidade do Sistema

Tabela 5 – Testes de Compatibilidade do Sistema

Dispositivo	Navegador	Resolução Testada	Resultado
Desktop Windows 10	Google Chrome	1920x1080	OK
Smartphone Android	Mozilla Firefox	1080x2400	Ajuste no layout aplicado
Tablet iOS	Safari	1536x2048	OK

Fonte: De Autoria Própria

3.7.2.4 Validação de Requisitos

Tabela 6 – Validação de Requisitos Identificados

Requisito	Origem (Entrevista)	Implementação no Sistema
Relatórios de vendas por período	Maria José Abreu da Silva	Implementado módulo de geração de relatórios com filtros por data
Controle de formas de pagamento	Nailma Ribeiro Gomes	Adicionada opção para registrar múltiplos métodos de pagamento
Personalização de relatórios financeiros	Marisa Amália Oliveira de Abreu	Permissão para gerar relatórios financeiros customizados

Fonte: De Autoria Própria

3.7.3 Justificativa para a Quantidade de Entrevistas

As entrevistas exploratórias realizadas foram planejadas para cobrir diferentes setores de pequenos negócios, garantindo um levantamento inicial das principais dificuldades enfrentadas por microempreendedores. A escolha dos participantes foi estratégica, contemplando:

- Varejo de vestuário, representado por uma empreendedora com desafios relacionados ao controle de estoque e análise de vendas.
- Comércio diversificado, com foco na gestão de pagamentos e previsão financeira.
- Prestação de serviços, com necessidades específicas na geração de relatórios financeiros e organização de clientes.

Durante o processo de desenvolvimento, verificou-se que os desafios apontados nas entrevistas se repetiam em diferentes cenários, tornando a amostra representativa para a fase inicial do projeto. Assim, optou-se por focar na implementação das funcionalidades já validadas ao invés de ampliar o número de entrevistas sem um sistema disponível para testes práticos.

Após a finalização do *deploy*, novos ciclos de validação serão conduzidos diretamente com os usuários para coletar feedbacks reais sobre o funcionamento do *Easy Sales*, garantindo um aprimoramento contínuo baseado na experiência de uso.

3.8 TECNOLOGIAS A SEREM UTILIZADAS

Nesta seção, serão explorados os aspectos tecnológicos que serão aplicados no desenvolvimento do sistema *Easy Sales*. A escolha das ferramentas e tecnologias para o desenvolvimento é crucial, pois afeta diretamente a eficácia do processo, a

escalabilidade do sistema e a facilidade de manutenção futura. O projeto abrangerá três componentes principais: a interface web (front-end), o aplicativo para dispositivos móveis (mobile) e a API de back-end. A seguir, apresentaremos uma visão detalhada das tecnologias necessárias para cada um desses componentes, considerando suas funções e requisitos técnicos específicos.

3.8.1 Front-end

O front-end da aplicação será desenvolvido utilizando TypeScript, Vue 3 e Ionic, uma combinação que oferece uma base sólida para a criação de interfaces de usuário eficientes, modernas e reativas. Essas tecnologias permitem que a aplicação seja responsiva e fácil de manter, enquanto garantem uma experiência de usuário fluida e intuitiva tanto para web quanto para dispositivos móveis.

O TypeScript é uma linguagem de programação que adiciona tipagem estática ao JavaScript, proporcionando maior segurança e robustez ao código. Sua capacidade de detectar erros em tempo de compilação melhora significativamente o processo de desenvolvimento, evitando problemas que poderiam surgir em runtime. Além disso, o uso de TypeScript facilita a manutenção de projetos maiores, já que o código torna-se mais organizado e previsível, beneficiando tanto equipes de desenvolvimento como a escalabilidade do projeto.

O Vue 3 é uma framework JavaScript progressiva que se destaca pela simplicidade e desempenho. Ele permite a criação de interfaces reativas e componentizadas, o que facilita a construção e a manutenção do front-end da aplicação. A modularidade do Vue, aliada ao seu sistema de reatividade, contribui para um desenvolvimento ágil, possibilitando a criação de interfaces dinâmicas com menos esforço e código mais limpo. Além disso, o Vue 3 traz melhorias significativas em termos de performance em comparação com sua versão anterior, sendo ideal para projetos que exigem alta interatividade.

Por fim, o Ionic será utilizado para desenvolver a interface mobile de forma híbrida, ou seja, a partir de um único código será possível criar aplicativos que rodem em diversas plataformas, como Android e iOS. O Ionic fornece uma ampla gama de componentes visuais prontos, que ajudam a acelerar o processo de desenvolvimento. Além disso, o framework oferece ferramentas para integrar recursos nativos dos dispositivos móveis, como GPS e câmeras, o que melhora a experiência do usuário ao interagir com o aplicativo em diferentes dispositivos.

Figura 17 – Vue e Ionic



Fonte: <<https://ionic.io/blog/announcing-the-ionic-vue-beta>>

3.8.2 Back-end

O back-end da aplicação foi construído utilizando Java, Spring Boot, *Supabase*, *Keycloak* e PostgreSQL. Essa combinação de tecnologias oferece uma infraestrutura robusta e escalável, garantindo a segurança, a eficiência e o processamento adequado das requisições feitas pela aplicação.

O Java é uma linguagem de programação amplamente utilizada em sistemas corporativos devido à sua robustez e versatilidade. A capacidade de Java de rodar em múltiplos sistemas operacionais sem necessidade de alterações no código (graças à JVM) a torna uma excelente escolha para o desenvolvimento de aplicações back-end. Além disso, o Java possui um ecossistema maduro e uma vasta quantidade de bibliotecas que facilitam a criação de soluções complexas.

O Spring Boot é um framework que simplifica o desenvolvimento de microserviços e APIs em Java. Sua configuração automatizada e sua modularidade permitem que os desenvolvedores foquem mais na lógica de negócios e menos em detalhes de configuração. A escalabilidade oferecida pelo Spring Boot é ideal para projetos que precisam crescer ao longo do tempo, além de garantir performance adequada mesmo em cenários de alta demanda.

Figura 18 – Java e Spring Boot



Fonte: <<https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQP9gmcZh0o2Cj8fPXOILSrpwqaBir27YXeRA&s>>

O *Supabase* é uma plataforma de backend como serviço (BaaS) que utiliza PostgreSQL como seu banco de dados subjacente. Além disso, essa ferramenta oferece

uma solução pronta para autenticação, gerenciamento de banco de dados e APIs em tempo real, o que acelera significativamente o desenvolvimento de funcionalidades de back-end. O uso do PostgreSQL, um banco de dados relacional robusto e com suporte a consultas complexas, garante a confiabilidade e a consistência dos dados armazenados e processados pela aplicação.

Figura 19 – Supabase e PostgreSQL



Fonte: https://supabase.com/_next/image?url=%2Fimages%2Fblog%2Fpostgres-bloat.png&w=3840&q=100&dpl=dpl_6kF4dSgrQozAFz56RPx9GCFXPDCg

O *Keycloak* foi empregado para gerenciamento de identidade e controle de acesso. Como uma solução open-source, o *Keycloak* facilita a implementação de autenticação e autorização, além de suportar protocolos como OAuth2 e SAML. Ele permite a centralização da gestão de usuários e funções, tornando o sistema mais seguro e eficiente em termos de controle de permissões e acessos.

Figura 20 – Keycloak



Fonte: <https://keycloak.org>

3.8.3 Trade-offs e desafios na escolha das tecnologias

A seleção das tecnologias para o desenvolvimento do *Easy Sales* envolveu diversas considerações técnicas e estratégicas, incluindo fatores como escalabilidade, facilidade de manutenção, curva de aprendizado e compatibilidade com outras ferramentas. No entanto, toda escolha tecnológica implica em trade-offs, ou seja, benefícios e desafios inerentes à decisão tomada.

No **front-end**, a escolha de Vue 3 e Ionic proporciona um desenvolvimento ágil e eficiente para múltiplas plataformas. No entanto, comparado a frameworks

mais estabelecidos como React e Angular, Vue 3 pode ter um ecossistema menor e menor disponibilidade de recursos avançados para grandes aplicações. Além disso, embora o Ionic facilite a criação de aplicações híbridas, ele pode apresentar desafios de desempenho em comparação com aplicações nativas desenvolvidas especificamente para iOS e Android.

No **back-end**, a adoção do *Spring Boot* com Java oferece robustez, segurança e alta escalabilidade. Contudo, essa escolha traz um custo adicional em termos de curva de aprendizado e maior consumo de recursos computacionais em comparação com alternativas mais leves, como Node.js com NestJS. Além disso, o *Spring Boot* exige um gerenciamento mais estruturado da infraestrutura para evitar gargalos de desempenho.

A utilização do **Supabase** como solução *backend-as-a-service* (BaaS) simplifica a implementação de banco de dados e autenticação. No entanto, essa abordagem pode gerar uma dependência em um provedor específico, limitando a flexibilidade caso seja necessário migrar para uma infraestrutura totalmente customizada no futuro. O *PostgreSQL*, por sua vez, é uma escolha confiável para bancos relacionais, mas pode ser mais complexo de gerenciar do que alternativas como MongoDB para casos onde o modelo de dados não é fortemente estruturado.

O **Keycloak** foi adotado para gerenciamento de identidade e controle de acesso, garantindo segurança e conformidade com protocolos de autenticação modernos. No entanto, sua configuração inicial pode ser complexa e demandar mais esforço de administração do que soluções mais leves, como Firebase Authentication.

Por fim, o desenvolvimento de um sistema robusto e escalável exige não apenas escolhas tecnológicas estratégicas, mas também um esforço significativo de aprendizado contínuo, manutenção e otimização. Dessa forma, os desafios de performance, compatibilidade e adoção do usuário precisam ser constantemente monitorados para garantir que o sistema evolua conforme as necessidades do mercado e dos clientes.

4 RESULTADOS

Neste capítulo, são apresentados os resultados obtidos com o desenvolvimento e a implementação das principais etapas do Easy Sales. Os resultados refletem a evolução do processo de desenvolvimento do software, destacando as funcionalidades implementadas e os testes realizados com os usuários finais. O objetivo é demonstrar como as funcionalidades do sistema foram desenvolvidas para atender às necessidades dos pequenos e médios empreendedores, proporcionando ganhos em eficiência, automatização de processos e suporte à tomada de decisões estratégicas. Além disso, são abordadas as questões relacionadas à usabilidade, desempenho e confiabilidade do software, que foram avaliadas ao longo do desenvolvimento, garantindo a robustez e a adequação da solução proposta.

4.1 APRESENTAÇÃO DO SOFTWARE

O Easy Sales é um sistema de gestão de vendas desenvolvido para otimizar as operações comerciais de pequenos e médios empreendedores. O software foi projetado com foco na simplicidade de uso e acessibilidade, permitindo que os usuários possam gerenciar de forma eficiente suas vendas, estoques, clientes e pedidos. A plataforma oferece uma interface amigável e intuitiva, garantindo que mesmo usuários com pouca experiência técnica possam utilizá-la sem dificuldades. O principal objetivo do Easy Sales é facilitar o controle de vendas e automatizar processos, reduzindo a necessidade de métodos manuais e promovendo maior eficiência operacional.

O sistema tem funcionalidades integradas para a gestão de estoque, emissão de relatórios de vendas, controle de inadimplência e simulação de negociações de débitos. O Easy Sales permite o cadastro de produtos e fornecedores, bem como o acompanhamento detalhado das transações realizadas, desde a entrada de mercadorias até a venda final. Além disso, o software oferece relatórios analíticos que auxiliarão na tomada de decisões estratégicas, proporcionando aos usuários uma visão clara e precisa sobre o desempenho de seus negócios. O uso de gráficos e dashboards facilitará o monitoramento das métricas-chave, como volume de vendas, produtos mais vendidos e análise de clientes.

A arquitetura do Easy Sales foi construída de maneira modular e escalável, utilizando tecnologias robustas como Java com Spring Boot no back-end e Vue 3 com Ionic no front-end. A integração com o banco de dados PostgreSQL garante segurança e confiabilidade na gestão dos dados, enquanto o Keycloak oferecerá um sistema de autenticação seguro, baseado em OAuth2. O design do sistema foi pensado para permitir fácil adaptação e expansão, tornando-o uma solução viável tanto

para pequenos negócios em fase inicial quanto para empresas em crescimento que necessitam de maior controle sobre suas operações.

4.2 MODELAGEM

A modelagem desempenha um papel fundamental no desenvolvimento de sistemas, proporcionando uma representação estruturada das principais funcionalidades e interações entre os atores e o sistema. Com a modelagem concluída, foram gerados diagramas detalhados que permitem visualizar claramente o comportamento esperado e os requisitos funcionais da aplicação. Diagramas de casos de uso e de classes foram utilizados para documentar como o sistema interage com os usuários e como as entidades estão inter-relacionadas, garantindo uma base sólida para o desenvolvimento.

A partir dessa modelagem, os atores envolvidos e os principais cenários de interação com o sistema foram devidamente mapeados. Essa abordagem permitiu a identificação antecipada de possíveis desafios, reduzindo riscos e assegurando que o sistema estivesse alinhado com os requisitos estabelecidos pelos stakeholders. A modelagem realizada forneceu uma diretriz clara para a implementação, contribuindo para um desenvolvimento mais eficiente e estruturado.

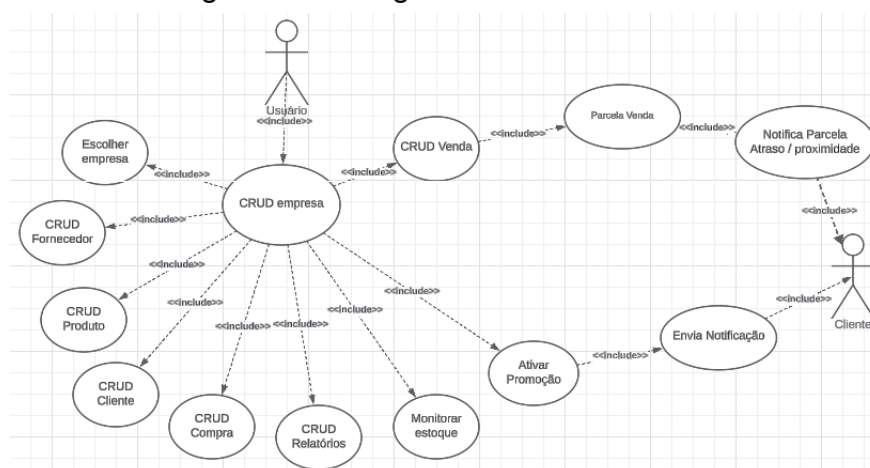
4.2.1 Diagrama de casos de uso

O diagrama de caso de uso ilustra, de forma visual, as interações entre os usuários ou atores do sistema e as funcionalidades oferecidas. Cada caso de uso representa um objetivo que o ator deseja atingir por meio do sistema, sendo uma maneira eficiente de capturar os requisitos funcionais. No diagrama de caso de uso apresentado 21, temos diversos atores, como usuários do sistema, interagindo com funções como "CRUD Empresa" e "Criar Notificação", que descrevem operações de cadastro, consulta, modificação e exclusão de empresas e notificações.

Por exemplo, o ator "Usuário Administrador" pode executar operações de gerenciamento de empresas, enquanto o "Usuário Comum" pode interagir com as funcionalidades de consulta de notificações. Esses casos de uso demonstram as principais responsabilidades do sistema e garantem que todos os requisitos de negócio sejam devidamente cobertos durante o desenvolvimento.

Através desse diagrama, é possível visualizar de forma clara as responsabilidades do sistema, permitindo que os desenvolvedores e partes interessadas validem se as interações descritas estão em conformidade com o que foi proposto inicialmente. Assim, o diagrama de caso de uso contribui para uma melhor compreensão do sistema como um todo, garantindo um desenvolvimento mais alinhado com as expectativas dos usuários finais.

Figura 21 – Diagrama de Caso de Uso



Fonte: De Autoria Própria

4.2.2 Diagrama de Classes

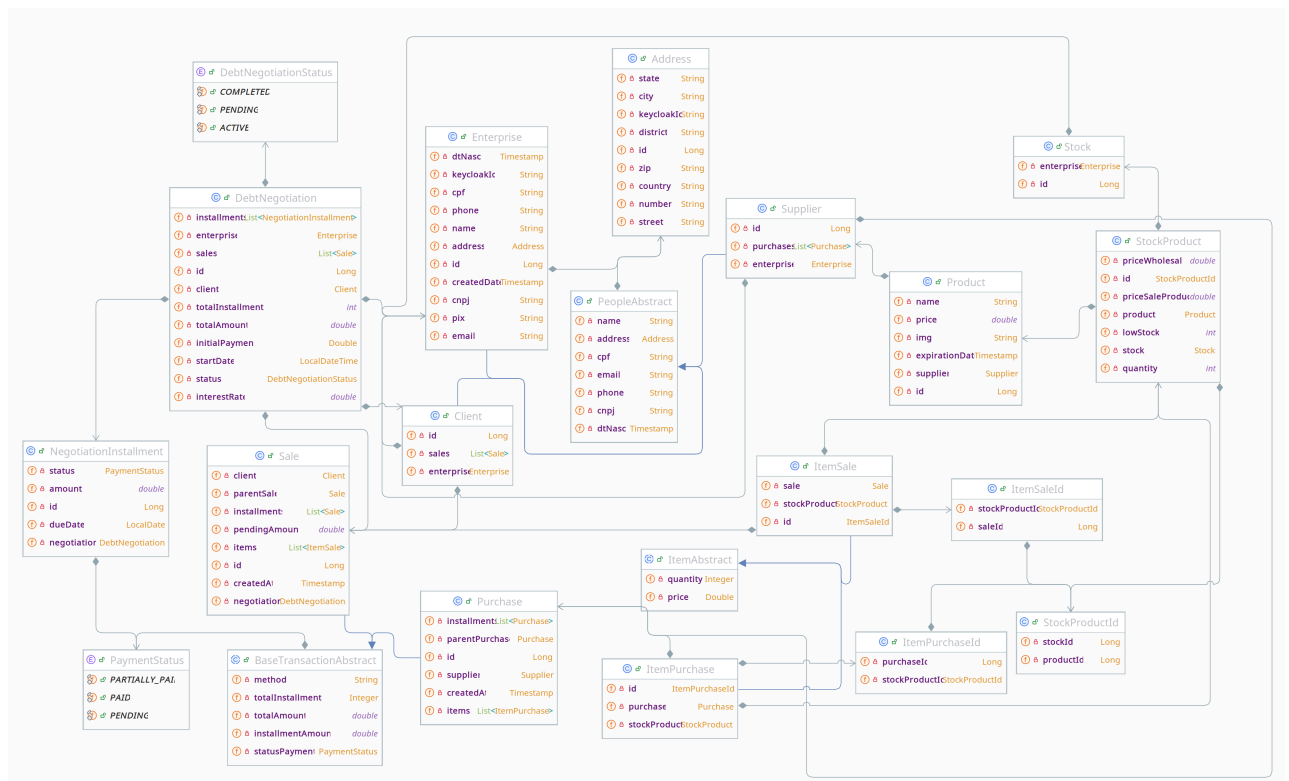
O diagrama de classes apresentado na imagem 22 descreve a estrutura do sistema em termos de suas entidades e os relacionamentos entre elas. Cada classe representa um componente do sistema, com seus atributos e associações, detalhando como esses componentes interagem para suportar os processos de negócio.

- **DebtNegotiation:** Responsável por armazenar as informações de negociações de dívidas.
- **NegotiationInstallment:** Representa as parcelas de uma negociação de dívida, associadas ao status de pagamento.
- **DebtNegotiationStatus:** Define os possíveis estados da negociação de dívida (COMPLETO, PENDENTE e ATIVO).
- **PaymentStatus:** Enumeração que indica o status do pagamento (PARCIALMENTE PAGO, PAGO e PENDENTE).
- **Enterprise:** Representa a empresa cadastrada no sistema, contendo informações como nome, CNPJ, endereço e contato.
- **Supplier e Client:** Ambas as classes contêm atributos comuns, como nome, endereço e contato. A classe `Client` está associada a uma lista de vendas (`List<Sale>`), representando as transações realizadas por esse cliente. Já a classe `Supplier` está associada a uma lista de compras (`List<Purchase>`), indicando que o fornecedor é responsável por fornecer os itens comprados pela empresa.
- **Stock:** Gerencia os estoques associados às empresas cadastradas.

- **StockProduct:** Relaciona os produtos ao estoque, incluindo informações de preço e quantidade.
- **Sale:** Classe que modela uma venda, contendo o cliente, produtos vendidos e o status da negociação.
- **Purchase:** Representa uma compra realizada de fornecedores para reposição de estoque. Similar à classe `Sale`, a classe `Purchase` descreve as transações de aquisição de produtos. A compra está relacionada com o fornecedor e contém uma lista de itens adquiridos (`List<ItemSale>`), possibilitando o controle detalhado de cada item de uma compra.
- **ItemSale e ItemPurchase:** Essas classes relacionam diretamente os itens vendidos ou comprados em suas respectivas transações. A `ItemSale` tem uma associação com `Stock`, que permite deduzir o estoque ao vender um item, enquanto `ItemPurchase` atualiza o estoque ao adquirir novos produtos.

A modelagem representada no diagrama de classes permite uma visualização clara das entidades envolvidas no sistema e suas interações, facilitando o entendimento e o desenvolvimento das funcionalidades propostas.

Figura 22 – Diagrama de Classes



Fonte: De Autoria Própria

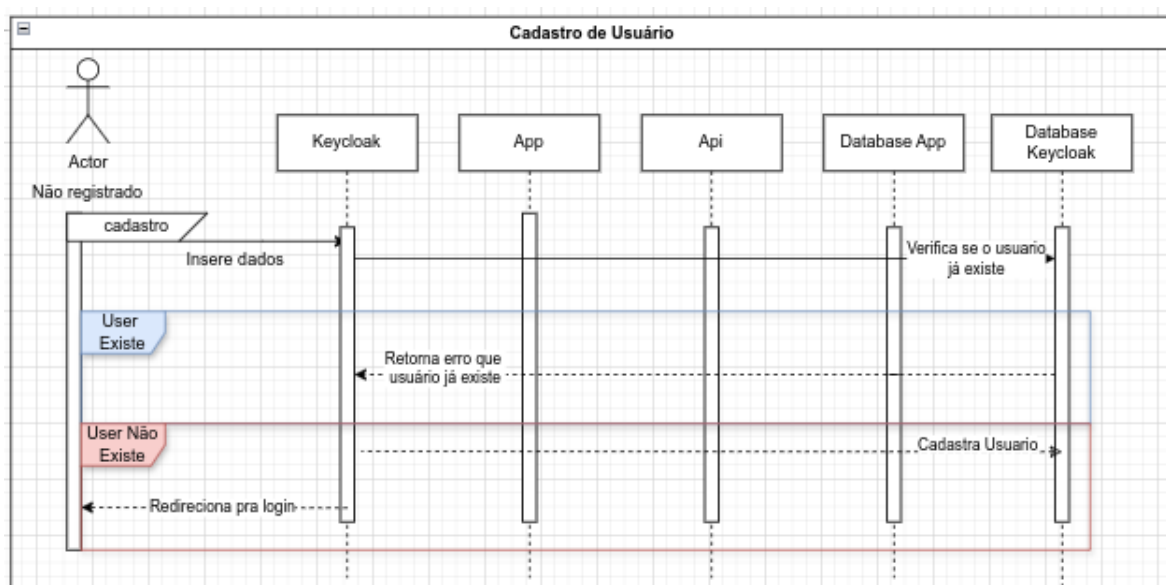
4.3 DIAGRAMAS DE SEQUÊNCIA

Nesta subseção, são apresentados os diagramas de sequência do sistema. Os diagramas de sequência descrevem as interações entre os objetos do sistema ao longo do tempo, modelando os fluxos de execução das principais funcionalidades. Eles são essenciais para entender a troca de mensagens entre os componentes do sistema e validar o comportamento esperado nas operações principais. Cada diagrama a seguir representa um processo fundamental dentro do sistema.

4.3.1 Registro de Usuário

O processo de registro de um novo usuário no sistema envolve a troca de informações entre o usuário, a interface de cadastro, a API e o banco de dados. Esse fluxo garante que os dados inseridos sejam validados e armazenados corretamente.

Figura 23 – Diagrama de Sequência do Registro de Usuário



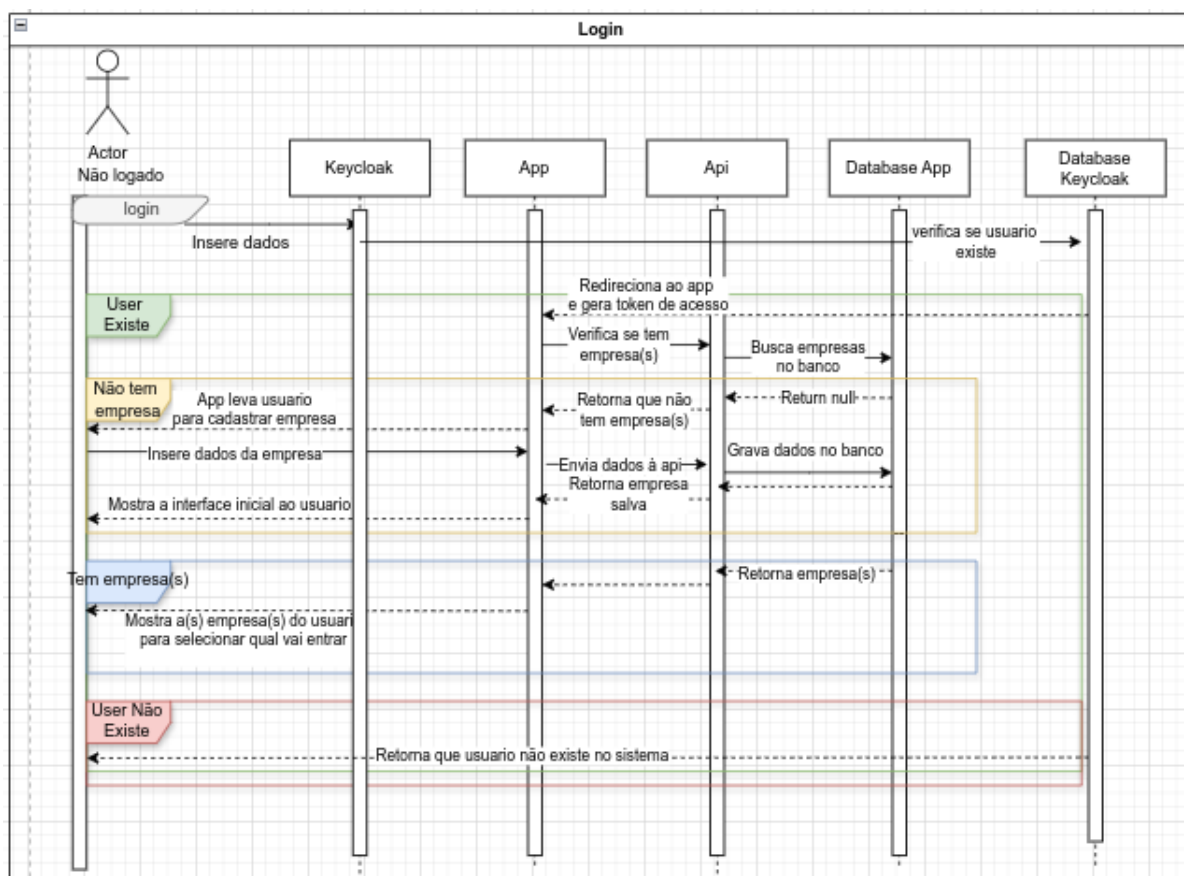
Fonte: De Autoria Própria

No diagrama apresentado na Figura 23, o usuário insere seus dados no sistema, que realiza validações e, caso todas as informações estejam corretas, armazena-as no banco de dados. Caso contrário, mensagens de erro são retornadas para correção.

4.3.2 Login no Sistema

O processo de login envolve a autenticação do usuário para garantir acesso seguro às funcionalidades do sistema. Esse fluxo inclui a verificação das credenciais inseridas e a geração de um token de autenticação.

Figura 24 – Diagrama de Sequência de Login



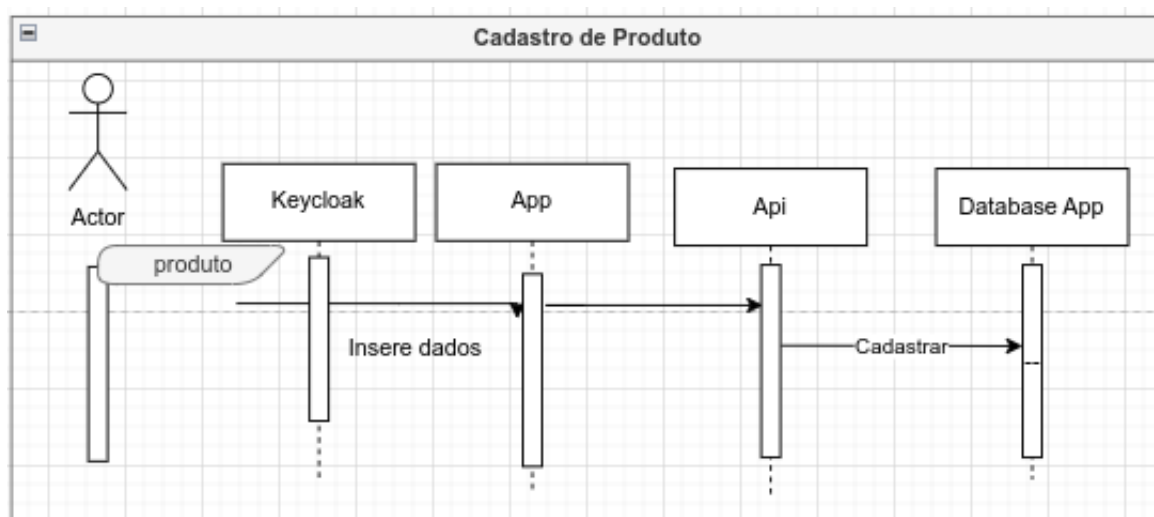
Fonte: De Autoria Própria

A Figura 24 apresenta a interação entre o usuário e os componentes de autenticação. O sistema verifica as credenciais inseridas e, se válidas, retorna um token de autenticação para permitir o acesso às funcionalidades protegidas.

4.3.3 Cadastro de Produto

O cadastro de produtos no sistema permite que o usuário insira informações detalhadas sobre um novo item, garantindo que ele fique disponível no estoque para futuras vendas.

Figura 25 – Diagrama de Sequência de Cadastro de Produto



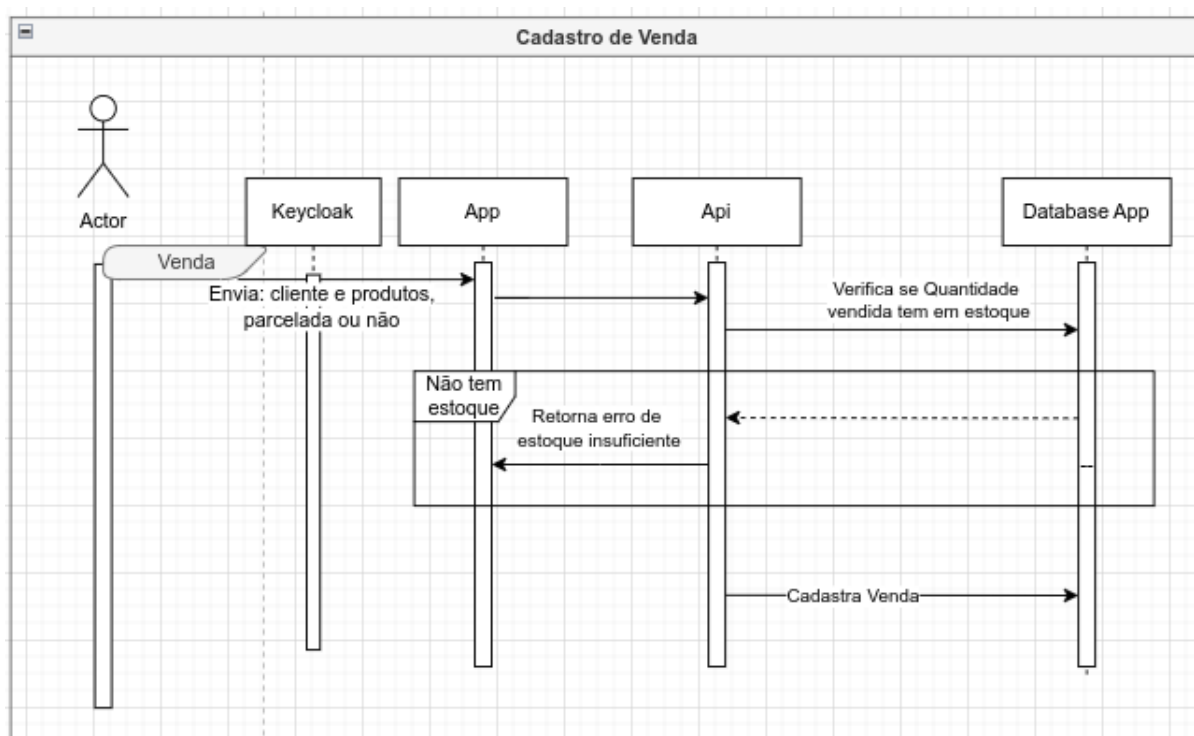
Fonte: De Autoria Própria

A Figura 25 demonstra o fluxo de cadastro de um produto, onde o usuário fornece detalhes como nome, quantidade e valor. O sistema processa a requisição e, após validação, persiste os dados no banco de dados.

4.3.4 Cadastro de Venda

O processo de cadastramento de uma venda envolve a seleção de produtos, a definição do cliente e a finalização da transação, garantindo a atualização do estoque e do histórico de vendas.

Figura 26 – Diagrama de Sequência de Cadastro de Venda



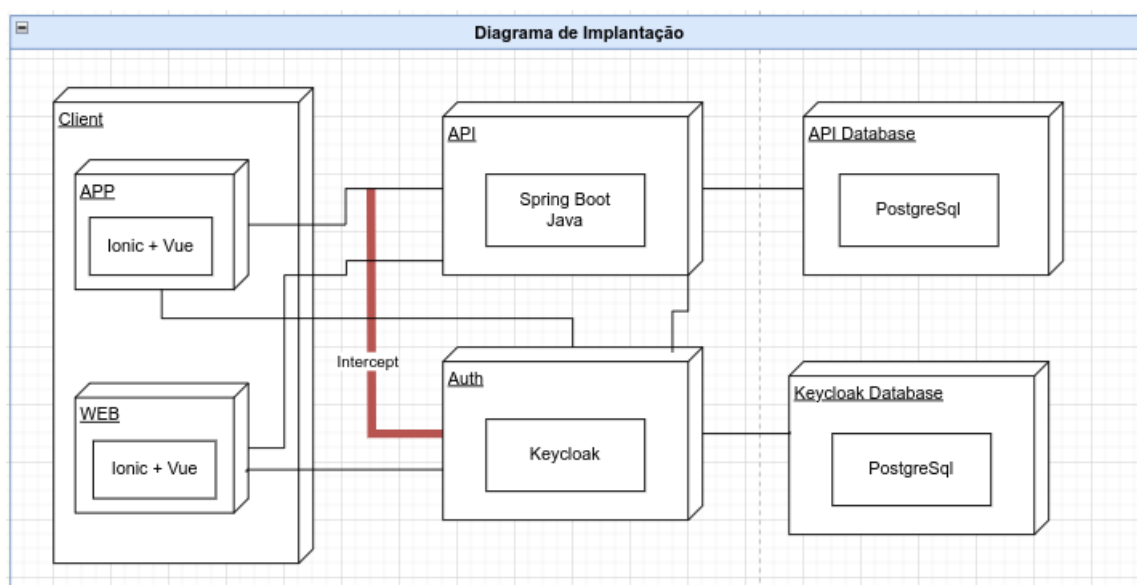
Fonte: De Autoria Própria

A Figura 26 ilustra as interações no processo de venda, desde a seleção dos produtos até a atualização do estoque e do histórico financeiro. O sistema verifica a disponibilidade dos itens e registra a venda com os detalhes necessários.

4.4 DIAGRAMA DE IMPLANTAÇÃO

O diagrama de implantação apresenta a arquitetura física do sistema, incluindo os servidores e dispositivos necessários para sua execução. Ele detalha a distribuição dos componentes do software e como eles se comunicam na infraestrutura de TI.

Figura 27 – Diagrama de Implantação do Sistema



Fonte: De Autoria Própria

A Figura 27 representa a infraestrutura necessária para o funcionamento do sistema, destacando a comunicação entre as aplicações cliente (*front-end*), a API e o servidor de autenticação. O *Keycloak* é responsável pela interceptação das requisições, garantindo que todas as operações sejam validadas antes de acessar os dados do banco. Esse modelo assegura a proteção das informações e um fluxo seguro para os usuários do sistema.

4.4.1 Arquitetura da aplicação

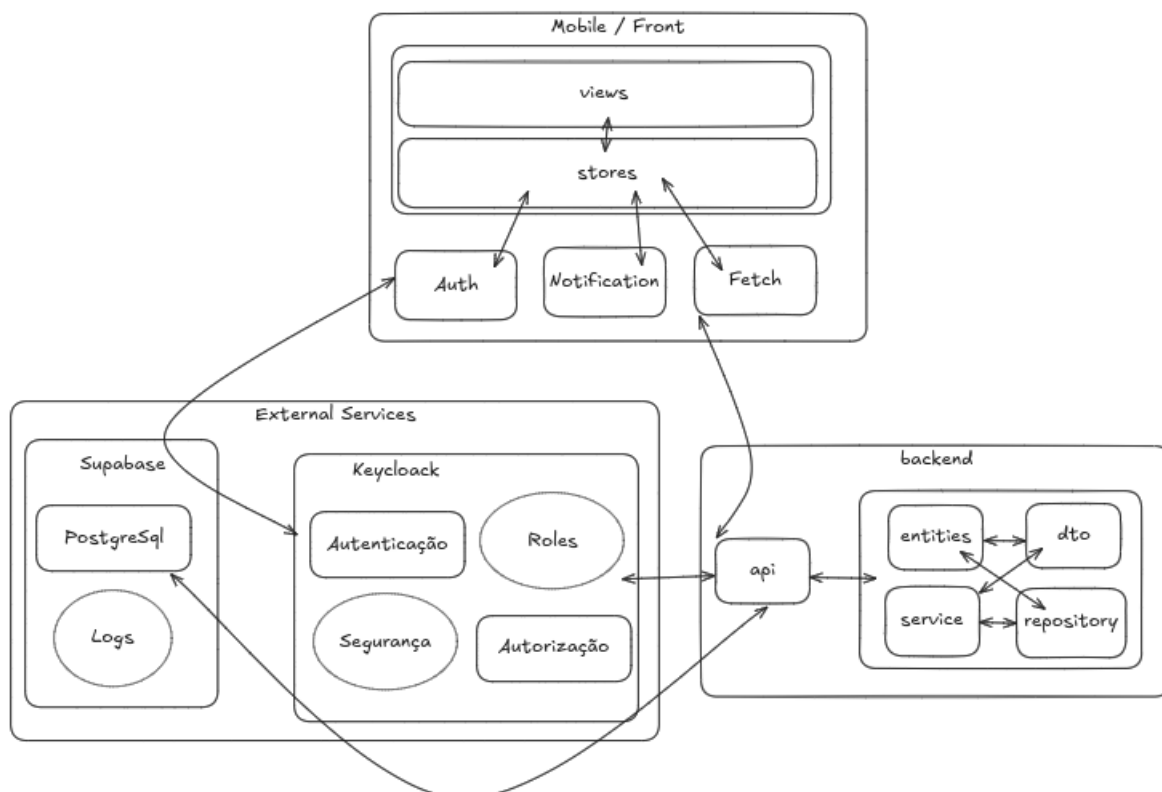
A arquitetura da aplicação mostrada na imagem 28 é composta por três principais camadas: o Mobile/Front, os External Services e o Backend. Cada uma dessas camadas desempenha um papel fundamental no funcionamento do sistema, proporcionando uma separação clara de responsabilidades e facilitando a escalabilidade e a manutenção da aplicação.

- **Mobile / Front:** A camada de front-end é responsável pela interação direta com o usuário. Ela é composta por views, que representam a interface gráfica, e stores, que gerenciam o estado da aplicação. As operações de autenticação (Auth), notificações (Notification) e requisições (Fetch) são feitas a partir dessa camada, que se comunica com os serviços externos e o backend por meio de APIs.

- **External Services:** Esta camada consiste em serviços externos que suportam a operação do sistema. O Supabase é utilizado para armazenamento de dados e logs, usando o PostgreSQL como banco de dados principal. O Keycloak é responsável pela autenticação e autorização, gerenciando papéis (Roles) e segurança. Esses serviços se integram diretamente com o backend e o front-end para fornecer funcionalidades de segurança e persistência de dados.
- **Backend:** A camada de backend é composta por várias partes, incluindo api, entities, dto, services e repository. O api é a interface de comunicação entre o front-end e o backend, enquanto as entities representam os objetos de domínio da aplicação. As dto (Data Transfer Objects) são utilizadas para transferir dados entre as camadas, e os services contêm a lógica de negócio. A comunicação com o banco de dados é feita pelos repositories, que garantem o acesso e a persistência dos dados.

A arquitetura modular da aplicação facilita a manutenção e a evolução do sistema. O uso de serviços externos especializados, como Supabase e Keycloak, permite delegar funcionalidades importantes, como autenticação e armazenamento, a ferramentas robustas, enquanto o *backend* se concentra na lógica de negócio e na comunicação com o banco de dados.

Figura 28 – Arquitetura da Aplicação



Fonte: De Autoria Própria

4.5 API

Este capítulo apresenta a documentação da API para o sistema Easy Sales, uma aplicação voltada para pequenos e médios empreendedores que precisam gerenciar vendas, estoque e clientes de maneira eficiente. A API está organizada de forma RESTful, facilitando a comunicação entre o backend, o frontend web e mobile (em Vue + Ionic). Além disso, será utilizada autenticação e autorização via Keycloak para garantir a segurança dos dados.

A seguir, detalharemos a estrutura básica da API, incluindo a configuração da autenticação, os endpoints de cada entidade, o tratamento de erros e a parte de relatórios, que será fundamental para fornecer informações detalhadas aos usuários sobre as operações de vendas e compras.

4.5.1 Autenticação e Autorização

A autenticação e autorização na API do Easy Sales são gerenciadas pelo Keycloak, uma solução robusta e open-source para controle de acesso e autenticação de usuários. Toda comunicação entre o cliente (frontend) e a API deve ser acompanhada por um token JWT, garantindo que somente usuários autenticados tenham acesso a operações sensíveis. Nesta seção, explicaremos como obter e utilizar tokens JWT. Além disso, a autenticação via Keycloak permite controlar o acesso de diferentes níveis de usuários (administradores, clientes, vendedores, etc.), além de permitir uma integração segura e centralizada com o sistema.

Para que um usuário possa interagir com a API protegida, ele precisará obter um token JWT como é mostrado na tabela 7. Esse token será utilizado nas requisições subsequentes para garantir que o usuário esteja autenticado e autorizado a realizar as operações. Abaixo, exemplificamos como realizar a requisição para obter o token. Dessa forma, Quando um usuário tenta se autenticar com credenciais inválidas, o Keycloak retorna um erro indicando que a autenticação falhou conforme a tabela 8. Esse erro é importante para que o cliente saiba o motivo da falha e possa tomar as medidas necessárias, como corrigir a senha ou verificar suas permissões.

4.5.2 Entidades e Controladores

A API Easy Sales é composta por vários controladores, cada um responsável por uma entidade do sistema, como clientes, fornecedores, empresas, entre outros. Cada controlador possui um conjunto de endpoints que permitem a manipulação dos dados dessas entidades. Nesta seção, descreveremos os principais controladores e suas funções. Além disso, esses controladores são projetados para serem facilmente acessados pelo frontend e mobile, proporcionando uma interface simples e eficaz para interagir com o backend, mantendo a consistência entre as operações.

Tabela 7 – Requisição para Obter Token de Acesso

Descrição	Detalhes
Método HTTP	POST
URL	/auth/realms/easysales/protocol/openid-connect/token
Host	keycloak.example.com
Content-Type	application/x-www-form-urlencoded
Parâmetros	client_id=<CLIENT_ID> client_secret=<CLIENT_SECRET> grant_type=password username=<USERNAME> password=<PASSWORD>

Fonte: De Autoria Própria

Tabela 8 – Formato do Erro de Autenticação

Campo	Valor
error	invalid_grant
error_description	Invalid user credentials

Fonte: De Autoria Própria

4.5.3 ClienteController

O `ClienteController` gerencia todas as operações relacionadas aos clientes cadastrados no sistema Easy Sales. Esse controlador permite que o sistema crie, edite, remova e liste clientes. Além disso, ele oferece mecanismos para validar dados de clientes e garantir que as informações estejam corretas antes de serem processadas.

Tabela 9 – Tabela de Endpoints: ClienteController

Método HTTP	Endpoint	Descrição
GET	/api/clientes	Lista todos os clientes
POST	/api/clientes	Cria um novo cliente
GET	/api/clientes/{id}	Retorna detalhes de um cliente
PUT	/api/clientes/{id}	Atualiza os dados de um cliente
DELETE	/api/clientes/{id}	Remove um cliente do sistema

Fonte: De Autoria Própria

4.5.4 FornecedorController

O `FornecedorController` é responsável pelo gerenciamento dos fornecedores no sistema. Ele lida com a criação de novos fornecedores, a atualização de dados, bem como a exclusão de fornecedores que não são mais necessários. Este controlador também inclui funcionalidades para obter uma lista completa de fornece-

dores registrados. Portanto, o uso de fornecedores é essencial para controlar o fluxo de compras e abastecimento do estoque, garantindo que o sistema tenha informações precisas sobre os produtos adquiridos.

Tabela 10 – Tabela de Endpoints: FornecedorController

Método HTTP	Endpoint	Descrição
GET	/api/fornecedores	Lista todos os fornecedores
POST	/api/fornecedores	Adiciona um novo fornecedor
GET	/api/fornecedores/{id}	Detalhes de um fornecedor
PUT	/api/fornecedores/{id}	Atualiza dados de um fornecedor
DELETE	/api/fornecedores/{id}	Exclui um fornecedor

Fonte: De Autoria Própria

4.5.5 EmpresaController

O `EmpresaController` permite o gerenciamento das empresas que utilizam o sistema. Cada empresa pode ser gerenciada por múltiplos usuários, e este controlador lida com a criação, atualização e exclusão de dados da empresa, bem como a consulta de informações detalhadas sobre cada uma. Dessa forma, a capacidade de gerenciar múltiplas empresas é útil para usuários que atuam em diferentes negócios ou unidades.

Tabela 11 – Tabela de Endpoints: EmpresaController

Método HTTP	Endpoint	Descrição
GET	/api/empresas	Lista todas as empresas
POST	/api/empresas	Registra uma nova empresa
GET	/api/empresas/{id}	Retorna detalhes de uma empresa
PUT	/api/empresas/{id}	Atualiza os dados de uma empresa
DELETE	/api/empresas/{id}	Exclui uma empresa

Fonte: De Autoria Própria

4.5.6 CompraController

O `CompraController` gerencia as compras feitas pelas empresas no sistema. Esse controlador é responsável por criar novas entradas de compra, atualizar registros existentes e excluir compras quando necessário. Ele também oferece uma visão geral de todas as compras registradas. Portanto, controlar as compras de forma eficiente é crucial para a gestão de estoque e controle financeiro da empresa.

Tabela 12 – Tabela de Endpoints: CompraController

Método HTTP	Endpoint	Descrição
GET	/api/compras	Lista todas as compras
POST	/api/compras	Registra uma nova compra
GET	/api/compras/{id}	Detalha uma compra específica
PUT	/api/compras/{id}	Atualiza uma compra
DELETE	/api/compras/{id}	Exclui uma compra

Fonte: De Autoria Própria

4.5.7 VendaController

O `VendaController` gerencia as vendas realizadas pela empresa. Este controlador é essencial para registrar e atualizar vendas, além de fornecer uma visão geral de todas as transações comerciais da empresa. Logo, a eficiência no controle de vendas permite que a empresa tenha uma visão clara de suas operações, facilitando a tomada de decisões estratégicas.

Tabela 13 – Tabela de Endpoints: VendaController

Método HTTP	Endpoint	Descrição
GET	/api/vendas	Lista todas as vendas
POST	/api/vendas	Registra uma nova venda
GET	/api/vendas/{id}	Detalha uma venda específica
PUT	/api/vendas/{id}	Atualiza uma venda
DELETE	/api/vendas/{id}	Exclui uma venda

Fonte: De Autoria Própria

O `ItemVendaController` gerencia os itens que compõem cada venda registrada no sistema. Assim como no caso das compras, o gerenciamento detalhado dos itens de venda permite um controle mais granular das operações comerciais. Dessa forma, cada venda pode incluir múltiplos produtos, e esse controlador permite o gerenciamento eficiente desses itens.

Tabela 14 – Tabela de Endpoints: ItemVendaController

Método HTTP	Endpoint	Descrição
GET	/api/itens-vendas	Lista todos os itens das vendas
POST	/api/itens-vendas	Adiciona um item a uma venda
GET	/api/itens-vendas/{id}	Detalha um item específico
PUT	/api/itens-vendas/{id}	Atualiza um item de venda
DELETE	/api/itens-vendas/{id}	Exclui um item de venda

Fonte: De Autoria Própria

4.5.8 SaleController

O `SaleController` gerencia as vendas realizadas pela empresa. Este controlador é essencial para registrar e atualizar vendas, além de fornecer uma visão geral de todas as transações comerciais da empresa. Logo, a eficiência no controle de vendas permite que a empresa tenha uma visão clara de suas operações, facilitando a tomada de decisões estratégicas.

Tabela 15 – Tabela de Endpoints: SaleController

Método HTTP	Endpoint	Descrição
GET	/api/sales/id	Retorna detalhes de uma venda
PUT	/api/sales/id	Atualiza uma venda
DELETE	/api/sales/id	Remove uma venda
GET	/api/sales	Lista todas as vendas
POST	/api/sales	Registra uma nova venda
GET	/api/sales/client/id/status	Obtém o status da venda para um cliente

Fonte: De Autoria Própria

4.5.9 EnterpriseController

O `EnterpriseController` permite o gerenciamento das empresas que utilizam o sistema. Cada empresa pode ser gerenciada por um usuário, e este controlador lida com a criação, atualização e exclusão de dados da empresa, bem como a consulta de informações detalhadas sobre cada uma. Dessa forma, a capacidade de gerenciar múltiplas empresas é útil para usuários que atuam em diferentes negócios ou unidades.

Tabela 16 – Tabela de Endpoints: EnterpriseController

Método HTTP	Endpoint	Descrição
POST	/enterprise	Registra uma nova empresa
GET	/enterprise/list-all	Lista todas as empresas
GET	/enterprise/by-user	Retorna empresas associadas ao usuário
PUT	/enterprise/id	Atualiza os dados da empresa

Fonte: De Autoria Própria

4.5.10 SupplierController

O `SupplierController` é responsável pelo gerenciamento dos fornecedores no sistema. Ele lida com a criação de novos fornecedores, a atualização de dados,

bem como a exclusão de fornecedores que não são mais necessários. Este controlador também inclui funcionalidades para obter uma lista completa de fornecedores registrados. Portanto, o uso de fornecedores é essencial para controlar o fluxo de compras e abastecimento do estoque, garantindo que o sistema tenha informações precisas sobre os produtos adquiridos.

Tabela 17 – Tabela de Endpoints: SupplierController

Método HTTP	Endpoint	Descrição
POST	/api/suppliers	Registra um novo fornecedor
GET	/api/suppliers/enterprise/enterpriseld	Lista fornecedores de uma empresa
PUT	/api/suppliers/enterprise/enterpriseld	Atualiza fornecedor de uma empresa
DELETE	/api/suppliers/enterprise/enterpriseld	Deleta fornecedor de uma empresa

Fonte: De Autoria Própria

4.5.11 PurchaseController

O `PurchaseController` O `CompraController` gerencia as compras feitas pelas empresas no sistema. Esse controlador é responsável por criar novas entradas de compra, atualizar registros existentes e excluir compras quando necessário. Ele também oferece uma visão geral de todas as compras registradas. Portanto, Controlar as compras de forma eficiente é crucial para a gestão de estoque e controle financeiro da empresa.

Tabela 18 – Tabela de Endpoints: PurchaseController

Método HTTP	Endpoint	Descrição
POST	/api/purchases	Registra uma nova compra
POST	/api/purchases/purchaseId/pay	Processa o pagamento de uma compra
GET	/api/purchases/id	Retorna detalhes de uma compra
GET	/api/purchases/enterprise/enterpriseld	Lista compras de uma empresa
DELETE	/api/purchases/enterprise/enterpriseld	Delete compra de uma empresa
PUT	/api/purchases/enterprise/enterpriseld	Atualiza uma compra

Fonte: De Autoria Própria

4.5.12 DebtNegotiationController

O `DebtNegotiationController` gerencia as negociações de dívida no sistema, permitindo sua criação, atualização e geração de relatórios. Esse controlador é responsável por criar novas negociações com os clientes, atualizar registros existentes e excluir quando necessário. Ademais, ele também fornece uma visão geral de todas as negociações que foram salvas.

Tabela 19 – Tabela de Endpoints: DebtNegotiationController

Método HTTP	Endpoint	Descrição
POST	/api/debts-negotiation	Inicia uma negociação de dívida
PATCH	/api/debts-negotiation/id	Atualiza uma negociação de dívida
PATCH	/api/debts-negotiation/id/confirm	Confirma uma negociação
PATCH	/api/debts-negotiation/id/close	Finaliza uma negociação
GET	/api/debts-negotiation/id/report	Gera relatório de uma negociação
GET	/api/debts-negotiation/status	Obtém os status disponíveis

Fonte: De Autoria Própria

4.5.13 ClientController

O `ClientController` gerencia todas as operações relacionadas aos clientes cadastrados no sistema. Ele permite que os clientes sejam criados, atualizados, listados e removidos do sistema.

Tabela 20 – Tabela de Endpoints: ClientController

Método HTTP	Endpoint	Descrição
GET	/api/client	Lista todos os clientes
POST	/api/client	Registra um novo cliente
GET	/api/client/id	Retorna detalhes de um cliente
DELETE	/api/client/id	Remove um cliente
PUT	/api/client/id	Atualiza informações de um cliente

Fonte: De Autoria Própria

4.5.14 StockController

O `StockController` gerencia o estoque de produtos das empresas cadastradas no sistema Easy Sales. Ele permite a consulta e gerenciamento do estoque associado a cada empresa.

Tabela 21 – Tabela de Endpoints: StockController

Método HTTP	Endpoint	Descrição
GET	/api/stock/enterprise/enterpriseld	Retorna o estoque de uma empresa

Fonte: De Autoria Própria

4.5.15 Relatórios

Os relatórios são uma parte vital do sistema Easy Sales, permitindo que os administradores e gestores extraiam informações detalhadas sobre as operações da empresa. Com os relatórios, os usuários podem tomar decisões baseadas em dados sobre vendas, compras, estoque e clientes. Nesta seção, detalhamos os principais endpoints para a geração de relatórios de vendas, compras e movimentações de estoque, financeiros, de estoque e operacionais.

Dessa forma, Os relatórios de vendas permitem que os administradores acompanhem as vendas por período, cliente ou tipo de pagamento, oferecendo dados essenciais para analisar o desempenho comercial da empresa e identificar oportunidades de crescimento. Esses relatórios ajudam a entender as tendências de vendas e a ajustar estratégias de marketing e vendas conforme necessário.

Ademais, os relatórios de compras ajudam os gestores a entender os gastos com fornecedores e controlar o fluxo de entrada de produtos, com a possibilidade de filtragem por data, fornecedor e valor total das compras. Esses relatórios são fundamentais para o controle financeiro e a otimização de processos de compra. Além disso, os relatórios de estoque permitem a monitoração da quantidade de produtos disponíveis e das movimentações recentes, sendo essenciais para evitar perdas e garantir que os pedidos dos clientes possam ser atendidos de forma eficaz.

Tabela 22 – Endpoints para geração de relatórios

Método	Endpoint	Descrição
GET	/api/report/enterpriseld	Relatório geral da empresa.
GET	/api/report/sales	Gera um relatório de todas as vendas.
GET	/api/report/sales?dataInicio=dataInicio&dataFim=dataFim	Relatório de vendas filtrado por intervalo de datas.
GET	/api/report/sales/clientId	Relatório de vendas filtrado por cliente.
GET	/api/report/purchases	Relatório de todas as compras.
GET	/api/report/purchases?dataInicio=dataInicio&dataFim=dataFim	Relatório de compras filtrado por data.
GET	/api/report/purchases/supplierId	Relatório de compras filtrado por fornecedor.

Fonte: De Autoria Própria

4.5.16 Tratamento de Erros

Os erros na API podem ser categorizados em diversos tipos como mostrado na tabela 23, sendo que cada um deles reflete situações comuns que o cliente pode enfrentar ao interagir com o sistema. Erros de autenticação surgem quando as credenciais fornecidas são inválidas ou o token de autorização é ausente ou inválido, bloqueando o acesso a recursos protegidos. Já os erros de validação ocorrem quando os dados fornecidos pelo cliente não estão de acordo com as regras estabelecidas, como a ausência de campos obrigatórios ou formatações incorretas, o que resulta em uma resposta clara para que o usuário possa corrigir a requisição.

Além disso, erros como o 404 (Recurso Não Encontrado) indicam que o cliente tentou acessar um recurso inexistente, enquanto o 405 (Método Não Permitido) alerta que o método HTTP utilizado não é aceito para o endpoint em questão. Erros como o 409 (Conflito) apontam inconsistências no estado do recurso, como a tentativa de duplicar dados. Por fim, erros relacionados ao tamanho da solicitação (413 Payload Muito Grande) e o tempo de processamento (408 Request Timeout) alertam o cliente para os limites de upload e de resposta do servidor, respectivamente, sugerindo ajustes para o bom funcionamento da requisição.

Tabela 23 – Exemplos de mensagens de erro retornadas pela API

Código de Status	Mensagem de Erro	Descrição
401	Acesso negado	Credenciais inválidas ou ausência de autorização.
400	Dados inválidos	Problemas de validação nos dados da requisição.
404	Recurso não encontrado	O recurso solicitado não existe ou não foi encontrado.
405	Método não permitido	O método HTTP utilizado não é permitido para este recurso.
409	Conflito no estado do recurso	Conflito com o estado atual do recurso (por exemplo, duplicidade).
413	Payload muito grande	O corpo da requisição excede o limite permitido pela API.
408	Tempo limite excedido	O servidor demorou muito para processar a requisição.

Fonte: De Autoria Própria

4.6 TESTES E VALIDAÇÕES

Neste segmento, os resultados dos testes realizadas no Easy Sales serão elucidados, abrangendo avaliações funcionais, de integração e de desempenho. Essas avaliações foram executadas para verificar se o sistema estava em conformidade com os requisitos estipulados e funcionava com precisão em todas as suas capacidades. Além disso, a eficácia do sistema em vários cenários de carga e a interatividade entre os módulos foram examinadas, com o objetivo de identificar possíveis aprimoramentos que facilitariam uma experiência perfeita e eficaz para os usuários. Abaixo, será apresentado um relato detalhado da metodologia empregada para cada categoria de avaliação e dos principais resultados alcançados.

4.6.1 Testes De Segurança

A análise de código-fonte foi realizada utilizando a ferramenta **SonarQube**, que permite identificar vulnerabilidades, problemas de confiabilidade, duplicação de código e manutenibilidade do projeto. Os critérios avaliados incluem:

- **Segurança:** Verificação de vulnerabilidades no código, como falhas de autenticação, exposição de dados sensíveis e injeção de código.

- **Confiabilidade:** Identificação de possíveis bugs que podem afetar a estabilidade da aplicação.
- **Manutenibilidade:** Medida da complexidade do código, analisando a presença de código duplicado e a facilidade de manutenção.

A Tabela 24 apresenta um resumo dos resultados obtidos.

Tabela 24 – Métricas de Qualidade no SonarQube

Métrica	Resultado	Descrição
Segurança	A	Sem vulnerabilidades identificadas
Confiabilidade	A	Nenhum bug crítico encontrado
Manutenibilidade	A (112)	Baixo nível de complexidade
Duplicação de Código	1.1%	Baixo nível de código duplicado

Fonte: De Autoria Própria

Os testes de segurança foram realizados para identificar vulnerabilidades no código-fonte e garantir a proteção contra possíveis ataques. O SonarQube utilizou análise estática para verificar falhas como:

- Injeção de SQL e XSS;
- Falhas de autenticação e autorização;
- Exposição de informações sensíveis;
- Uso inadequado de bibliotecas externas.

Conforme ilustrado na Figura 29, o projeto obteve **nota A** em segurança, o que indica que nenhuma vulnerabilidade foi encontrada durante a análise.

Figura 29 – Relatório de Segurança no SonarQube



Fonte: De Autoria Própria

4.6.2 Testes De Performance

Os testes de performance são fundamentais para avaliar o comportamento de um sistema sob carga. O **Apache JMeter** é uma ferramenta popular para testes de carga, permitindo simular múltiplos usuários acessando simultaneamente uma aplicação web. Neste experimento, utilizamos o JMeter para avaliar o desempenho da API do projeto **Easy Sales**, que utiliza **Keycloak** para autenticação.

Para o teste, foram configurados:

- **100 usuários simultâneos;**
- **3 requisições por usuário;**
- **3 end points para cada usuário;**
- **Período de 3 segundos.**

O objetivo foi medir o tempo de resposta da API, a taxa de erros e a capacidade de processamento do servidor.

4.6.2.1 Resultados Obtidos

Os resultados dos testes foram registrados na Tabela 25.

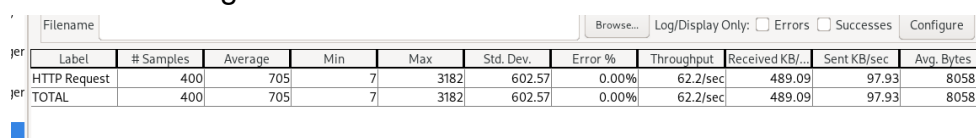
Tabela 25 – Resultados do Teste de Performance no JMeter

Amostras	Média(ms)	Max(ms)	Desv. Padrão (ms)	Erro %	Throughput(req/s)
400	7	3182	602.57	0.00%	62.2

Fonte: De Autoria Própria

A Figura 30 apresenta uma captura da tela do JMeter com os resultados.

Figura 30 – Resultados do teste no JMeter



Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	400	705	7	3182	602.57	0.00%	62.2/sec	489.09	97.93	8058.
TOTAL	400	705	7	3182	602.57	0.00%	62.2/sec	489.09	97.93	8058.

Fonte: De Autoria Própria

4.6.2.2 Interpretação dos Resultados

Os resultados mostram que a API conseguiu processar **400 requisições** sem erros, mantendo uma taxa de **62,2 requisições por segundo**. O tempo médio de resposta foi de **705 ms**, enquanto o tempo máximo chegou a **3182 ms**, sugerindo que algumas requisições demoraram significativamente mais que outras.

A ****variação no tempo de resposta**** (desvio padrão de 602.57 ms) indica possíveis gargalos, como processamento no banco de dados ou limitações na infraestrutura.

4.6.2.3 Conclusão e Limites do Teste

Os testes foram executados em um ambiente local, utilizando um **computador com processador Intel Core i3 de 5ª geração e 12GB de RAM**. Esse hardware pode ter influenciado nos tempos de resposta, uma vez que ****não representa um ambiente de produção otimizado****. Em um servidor mais robusto, o desempenho pode ser significativamente melhor.

Apesar das limitações do hardware, os resultados sugerem que a API está respondendo dentro de um tempo aceitável para carga moderada. Entretanto, para cenários com maior volume de requisições simultâneas, pode ser necessário otimizar o banco de dados, implementar cache ou escalonar a aplicação.

4.7 BACKLOG E SPRINTS

Na Sprint 1, o foco foi principalmente no desenvolvimento de operações CRUD para diferentes entidades do sistema, tanto no front-end quanto no back-end. Entre as tarefas realizadas, destaca-se o CRUD de vendedor, cliente, vendas, fornecedor, produto e empresa, abrangendo todas as camadas principais do sistema. Esse estágio inicial foi fundamental para a construção das funcionalidades básicas do sistema, permitindo a manipulação eficiente dos dados necessários para o funcionamento da aplicação. Com isso, estabeleceu-se uma base sólida para os próximos incrementos funcionais, garantindo que as operações principais estivessem integradas entre as camadas de interface e back-end.

Figura 31 – Sprint 1

✓	EAS-104	[front-end] CRUD empresa	KW Kairon Willyan	None +1	Medium	CONCLUÍDO ▾
✓	EAS-103	[back-end] CRUD empresa	KW Kairon Willyan	None +1	Medium	CONCLUÍDO ▾
✓	EAS-24	[front-end] CRUD produto	KW Kairon Willyan	None +1	Medium	CONCLUÍDO ▾
✓	EAS-10	[back-end] CRUD cliente	KW Kairon Willyan	None +1	Medium	CONCLUÍDO ▾
✓	EAS-9	[front-end] CRUD cliente	KW Kairon Willyan	None +1	Medium	CONCLUÍDO ▾
✓	EAS-8	[back-end] CRUD produto	KW Kairon Willyan	None +1	Medium	CONCLUÍDO ▾
✓	EAS-7	[front-end] CRUD fornecedor	KW Kairon Willyan	None +1	Medium	CONCLUÍDO ▾
✓	EAS-6	[back-end] CRUD fornecedor	KW Kairon Willyan	None +1	Medium	CONCLUÍDO ▾

Fonte: De Autoria Própria

A Sprint 2 foi marcada pela implementação de funcionalidades mais avançadas e pela integração de ferramentas externas essenciais para a segurança e infraestrutura do sistema. As tarefas principais incluíram o desenvolvimento do CRUD das funcionalidades principais do sistema, que são as de venda e compra, e a implementação do Keycloak para a gestão de identidade e controle de acesso. Além disso, foi configurado o serviço de banco de dados, o que foi crucial para garantir que os dados fossem armazenados e acessados de maneira eficiente e segura. Essa sprint serviu para aprimorar a estrutura do projeto, garantindo maior escalabilidade e segurança, o que será essencial para o crescimento contínuo do sistema.

Figura 32 – Sprint 2

Type	Chave	Resumo	Responsável	Sprint	Prioridade	Estado
	EAS-108	[front-end] CRUD de compras	 Kairon Willyan	None +1	Medium	CONCLUÍDO
	EAS-107	[back-end] CRUD de compras	 Kairon Willyan	None +1	Medium	CONCLUÍDO
	EAS-100	[front-end] Implantar Keycloak	 Kairon Willyan	None +1	Medium	CONCLUÍDO
	EAS-35	Configuração de serviço de banco de dados	 Kairon Willyan	None +1	Medium	CONCLUÍDO
	EAS-32	[back-end] Implantar Keycloak	 Kairon Willyan	None +1	Medium	CONCLUÍDO
	EAS-5	[front-end] CRUD de vendas	 Kairon Willyan	None +1	Medium	CONCLUÍDO
	EAS-4	[back-end] CRUD de vendas	 Kairon Willyan	None +1	Medium	CONCLUÍDO

Fonte: De Autoria Própria

Na Sprint 3, o foco principal foi a geração de relatórios e a finalização da infraestrutura do projeto. Entre as principais tarefas, foram implementados relatórios de vendas e produtos, funcionalidades importantes para o acompanhamento e análise do desempenho da aplicação. Além disso, foram realizados os ajustes no controle financeiro, que permitem uma melhor gestão dos recursos financeiros do sistema.

Figura 33 – Sprint 3

▼ quadro Sprint 3 1 jan – 31 jan (12 problemas)			7	0	5	Concluir sprint	...
■	EAS-13	[front-end] Gerar relatório das vendas	👤	RELATÓRIO	CONCLUÍDO ▼	👤	
■	EAS-105	[front-end] relatório dos produtos			CONCLUÍDO ▼	👤	KW
■	EAS-41	[back-end] relatório dos produtos	👤	RELATÓRIO	CONCLUÍDO ▼	👤	
■	EAS-42	[back-end] Gerar relatório das vendas	👤	RELATÓRIO	CONCLUÍDO ▼	👤	
■	EAS-25	[back-end] Controle financeiro ✎	👤	RELATÓRIO	CONCLUÍDO ▼	👤	...
■	EAS-109	[front-end] grafico de vendas			A FAZER ▼	👤	KW
■	EAS-110	[front-end] grafico de compras			A FAZER ▼	👤	KW
■	EAS-111	[front-end] grafico de produtos			A FAZER ▼	👤	KW
■	EAS-112	[front-end] grafico de empresas			A FAZER ▼	👤	KW
■	EAS-114	[back-end] grafico de compras			A FAZER ▼	👤	KW
■	EAS-115	[back-end] grafico de produtos			A FAZER ▼	👤	KW
■	EAS-116	[back-end] grafico de empresas			A FAZER ▼	👤	KW

Fonte: De Autoria Própria

Na Sprint 4, o foco principal é a geração da funcionalidade de negociação de dívidas e a finalização da infraestrutura do projeto. Entre as principais tarefas, foram implementadas a negociação de dívidas com o cliente. Por fim, o deploy da aplicação tanto no front-end quanto no back-end será realizado, concluindo assim uma etapa importante no ciclo de desenvolvimento, com a entrega de um produto pronto para ser testado em um ambiente de produção.

Figura 34 – Sprint 4



Item	Summary	Status	Column
EAS-102	[front-end] deploy	EM PROGRESSO	INFRAESTRUTURA DA P...
EAS-101	[back-end] deploy	CONCLUÍDO	INFRAESTRUTURA DA P...
EAS-106	Keycloak deploy	A FAZER	A FAZER
EAS-117	[back-end] Negociação	A FAZER	A FAZER
EAS-118	[front-end] Negociação	A FAZER	A FAZER

Fonte: De Autoria Própria

4.8 CRONOGRAMA DE EXECUÇÃO

O cronograma do projeto foi planejado em três sprints conforme a Tabela 26 abaixo, cada uma com um conjunto específico de atividades voltadas à construção progressiva do sistema. A Sprint 1 focou na implementação inicial das funcionalidades CRUD, com o objetivo de permitir o gerenciamento de entidades essenciais como clientes, fornecedores, produtos e vendas, estabelecendo uma base funcional preliminar. Na Sprint 2, o foco está voltado para a infraestrutura do sistema, com a adição planejada de componentes críticos como a documentação da API e a integração do Keycloak para gestão de segurança. Ademais, a Sprint 3 abordará a entrega de relatórios detalhados. Por fim, a Sprint 4 será responsável pelo deploy da aplicação e negociação de dívidas, finalizando o ciclo de desenvolvimento, deixando a aplicação pronta para ser testada e avaliada em um ambiente de produção. Esse planejamento metódico foi pensado para garantir uma implementação incremental e estruturada, com o intuito de aumentar as chances de sucesso do projeto, conforme as etapas forem sendo concluídas.

Tabela 26 – Cronograma de execução das sprints

Sprint	Tarefas	Início	Fim
Sprint 1	- [front-end] CRUD do fornecedor - [front-end] CRUD cliente - [front-end] CRUD produto - [front-end] CRUD empresa - [back-end] CRUD fornecedor - [back-end] CRUD cliente - [back-end] CRUD produto - [back-end] CRUD empresa	30 set 2024	15 out 2024
Sprint 2	- [front-end] CRUD de vendas - [front-end] CRUD de compras - [back-end] CRUD de vendas - [back-end] CRUD de compras - [back-end] Implantar Keycloak - [front-end] Implantar Keycloak - Configuração do serviço de banco de dados	30 out 2024	30 nov 2024
Sprint 3	- [front-end] Gerar relatório de vendas - [front-end] Relatório dos produtos - [front-end] Relatório dos compras - [front-end] Relatório dos empresas - [back-end] Gerar relatório de vendas - [back-end] Relatório dos produtos - [back-end] Relatório dos compras - [back-end] Relatório dos empresas	01 jan 2025	31 jan 2025
Sprint 4	- [back-end] negociação - [back-end] negociação - [back-end] Deploy - [front-end] Deploy	01 fev 2025	10 fev 2025

Fonte: De Autoria Própria

4.9 INTERFACES DESENVOLVIDAS

Nesta seção, serão apresentadas as interfaces reais desenvolvidas para a aplicação. Essas interfaces representam com precisão o layout e as interações do usuário, incorporando todos os elementos visuais e funcionais definidos no design. O design final inclui cores, tipografia, ícones e demais detalhes gráficos, garantindo uma experiência visual e interativa completa. As interfaces refletem fielmente a versão que será entregue aos usuários, permitindo a validação final antes da implantação. Dessa forma, asseguram que o produto atenda às expectativas dos usuários e aos requisitos de usabilidade, proporcionando uma experiência consistente e intuitiva.

- **Cadastro de Empresas:** O processo de cadastro permite ao usuário registrar uma nova empresa ou selecionar uma já existente. Durante essa etapa, são inseridas informações essenciais, como nome, CPF ou CNPJ e dados de contato. Ao final, o usuário pode revisar os dados antes de concluir o cadastro, garantindo que todas as informações estejam corretas.

Figura 35 – Fluxo de Cadastro de Empresas

The figure displays three sequential mobile app screens for company registration, each titled 'Cadastrar Empresa'.

Screen 1: Dados da empresa

- Field: Nome da empresa
- Section: Você possui:
- Options: ☐ CPF, ☐ CNPJ
- Field: CNPJ
- Button: PRÓXIMA ETAPA

Screen 2: Endereço da Empresa

- Field: Endereço: Deanna Green
- Field: Número: Lane Slater
- Field: Cidade: Madison Gardner
- Field: Bairro: Iona Hale
- Field: CEP: Ignatius Frost
- Field: País: Calista Herrera
- Buttons: VOLTAR, PRÓXIMA ETAPA

Screen 3: Confirme seus Dados

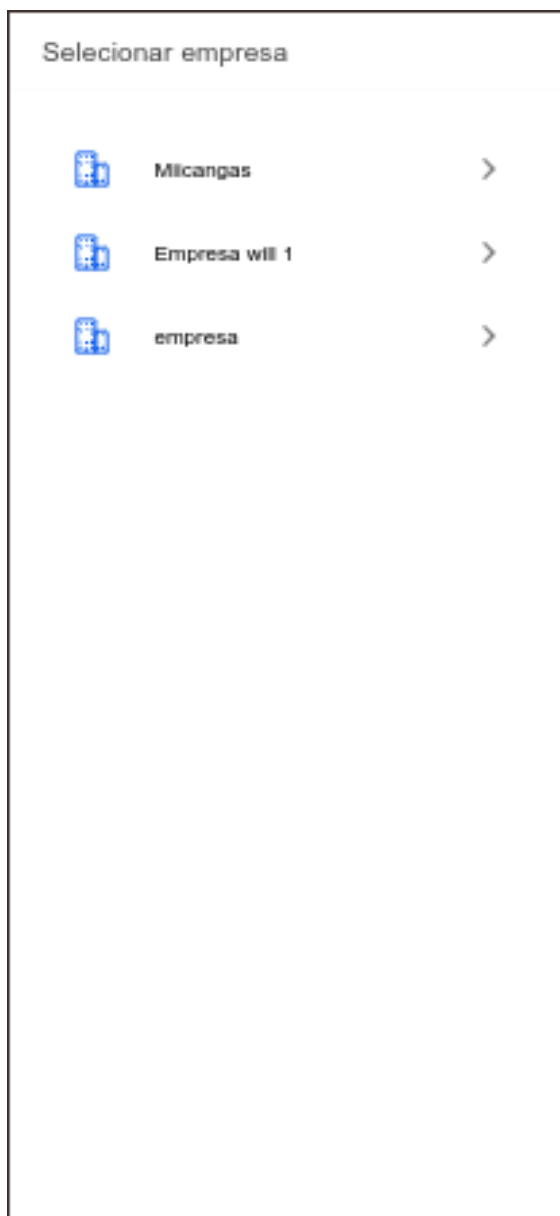
- Field: Nome da loja: empresa
- Field: Documento: CPF - 999999999
- Field: Endereço: Deanna Green, Lane Slater
- Field: Cidade/Bairro: Madison Gardner / Iona Hale
- Field: CEP/País: Ignatius Frost / Calista Herrera
- Buttons: VOLTAR, CONFIRMAR E CADASTRAR

Fonte: De Autoria Própria

A interface do cadastro foi projetada para ser intuitiva, permitindo um preenchimento rápido e eficiente. O usuário pode visualizar os dados inseridos e realizar ajustes antes de finalizar o processo.

- **Listagem de Empresas:** Para facilitar a gestão de múltiplas empresas, o sistema disponibiliza uma tela de listagem onde todas as organizações registradas podem ser visualizadas. O usuário pode alternar entre elas e acessar rapidamente os dados de cada empresa.

Figura 36 – Listagem de Empresas



Fonte: De Autoria Própria

Essa funcionalidade permite um gerenciamento eficiente, evitando a necessidade de múltiplos acessos ou cadastros repetitivos.

- **Alternância entre Empresas:** Para usuários que administram mais de uma empresa, o sistema oferece a opção de alternância entre organizações cadastradas. A troca pode ser feita rapidamente por meio de um menu suspenso (*dropdown*), proporcionando uma navegação ágil.

Figura 37 – Alternância entre Empresas



Fonte: De Autoria Própria

Esse recurso é essencial para administradores que precisam monitorar diferentes negócios dentro da mesma plataforma.

- **Tela Inicial (Dashboard):** Após o login e a seleção da empresa, o usuário é direcionado para o painel inicial da aplicação. O dashboard apresenta um resumo das principais informações do negócio, incluindo métricas de vendas, produtos em estoque e gráficos analíticos.

Figura 38 – Tela Inicial do Sistema



Fonte: De Autoria Própria

Além disso, o usuário pode acessar rapidamente outras seções do sistema por meio do menu lateral, garantindo uma experiência fluida e eficiente.

- **Menu Lateral:** O menu lateral facilita a navegação entre as diferentes funcionalidades do sistema, oferecendo acesso rápido a seções como clientes, produtos, vendas, relatórios e configurações.

Figura 39 – Menu de Navegação Lateral



Fonte: De Autoria Própria

O menu foi desenvolvido para permanecer fixo na interface, permitindo uma navegação intuitiva e contínua sem a necessidade de recarregar a página.

- **Cadastro de Fornecedores:** Nesta tela, o usuário pode registrar novos fornecedores, informando detalhes como nome, contato e endereço. Além disso, a interface possibilita a visualização de todos os fornecedores cadastrados e o acesso ao histórico de compras realizadas.

Figura 40 – Cadastro de Fornecedores

Dados Pessoais

Nome
Xantha Petersen

CPF/CNPJ
Sophia Neal

Telefone
Kim Green

Empresa/Produto
Constance Bruce

Próxima etapa

Endereço

Logradouro
Halla Leach

Bairro
Paki Solis

Número
Oren Mayer

CEP
Dante Hester

Cidade
Tucker Duffy

Estado
PI

Finalizar cadastro

Voltar

Fonte: De Autoria Própria

A funcionalidade de avaliação de fornecedores, representada por um ícone visual, permite um melhor gerenciamento das relações comerciais.

- **Cadastro de Produtos:** O sistema possibilita o cadastro de novos produtos, permitindo a inclusão de informações como nome, código de identificação, quantidade em estoque, valor unitário e uma imagem ilustrativa.

Figura 41 – Cadastro de Produtos

9:41

← Cadastro de produtos

Adicionar produto

Nome do produto

ID

ID do produto

Quantidade

Unidade

Un

Valor do produto

Valor

Imagem do produto

↑

Faça upload de uma foto do seu produto

Procurar na galeria

Capturar agora

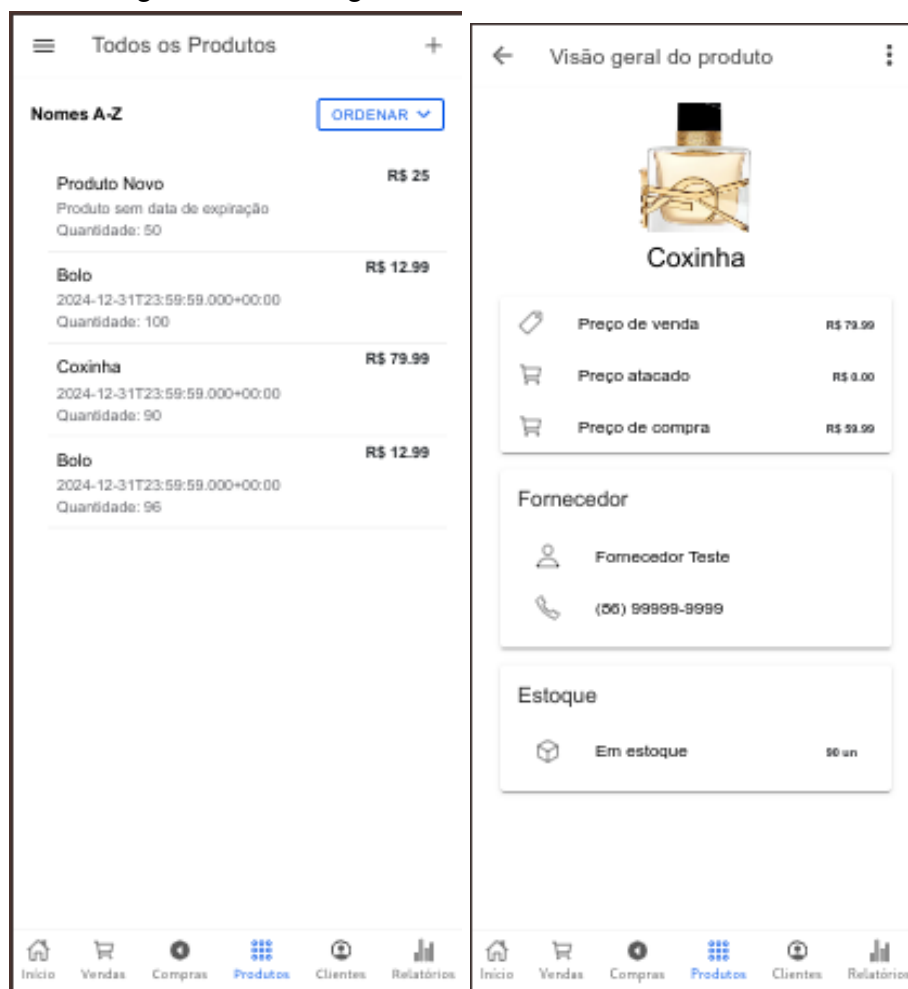
Cadastrar produto

Fonte: De Autoria Própria

Além disso, é possível associar cada produto a um fornecedor e definir preços diferenciados para revenda e atacado.

- **Listagem e Detalhes de Produtos:** Essa tela apresenta uma visão organizada de todos os produtos cadastrados. O usuário pode visualizar informações detalhadas, como nome, preço e quantidade disponível em estoque.

Figura 42 – Listagem e Detalhamento de Produtos

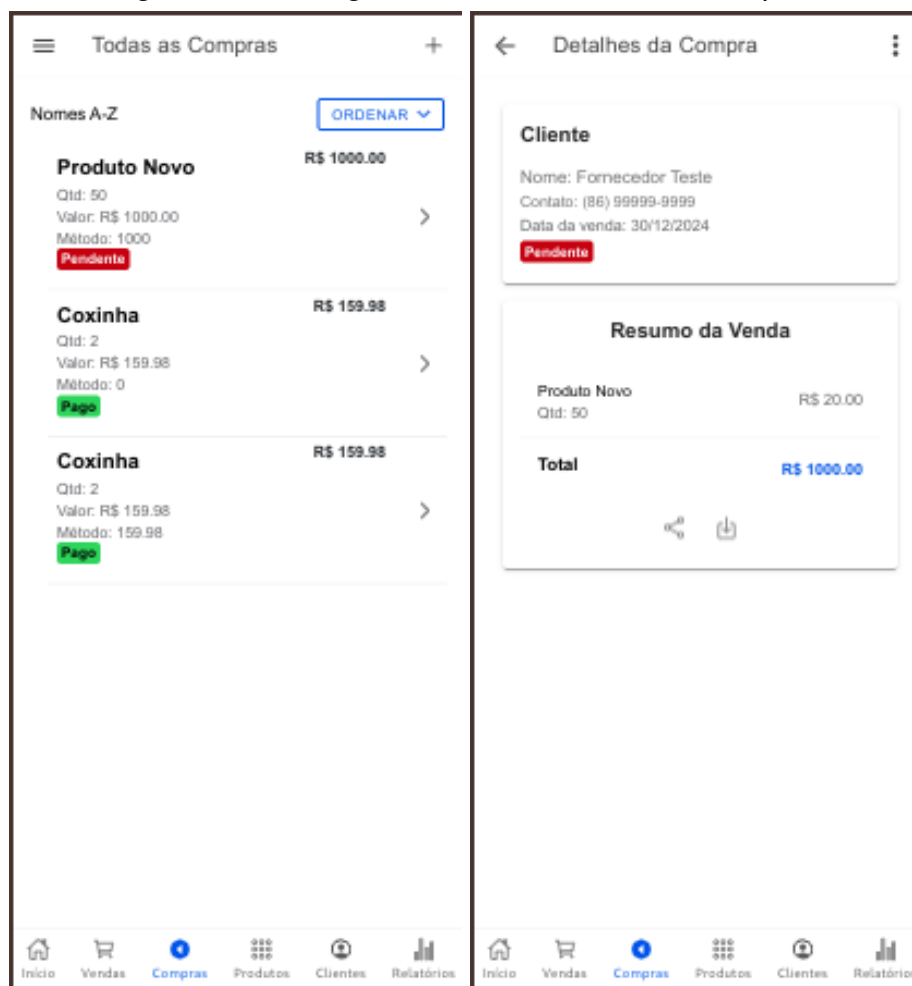


Fonte: De Autoria Própria

A funcionalidade de filtros permite ordenar os produtos conforme critérios específicos, facilitando a gestão do catálogo.

- **Detalhes e Listagem de Compras:** O sistema permite ao usuário acessar informações detalhadas sobre cada compra registrada, incluindo os itens adquiridos e seus respectivos valores.

Figura 43 – Listagem e Detalhamento de Compras

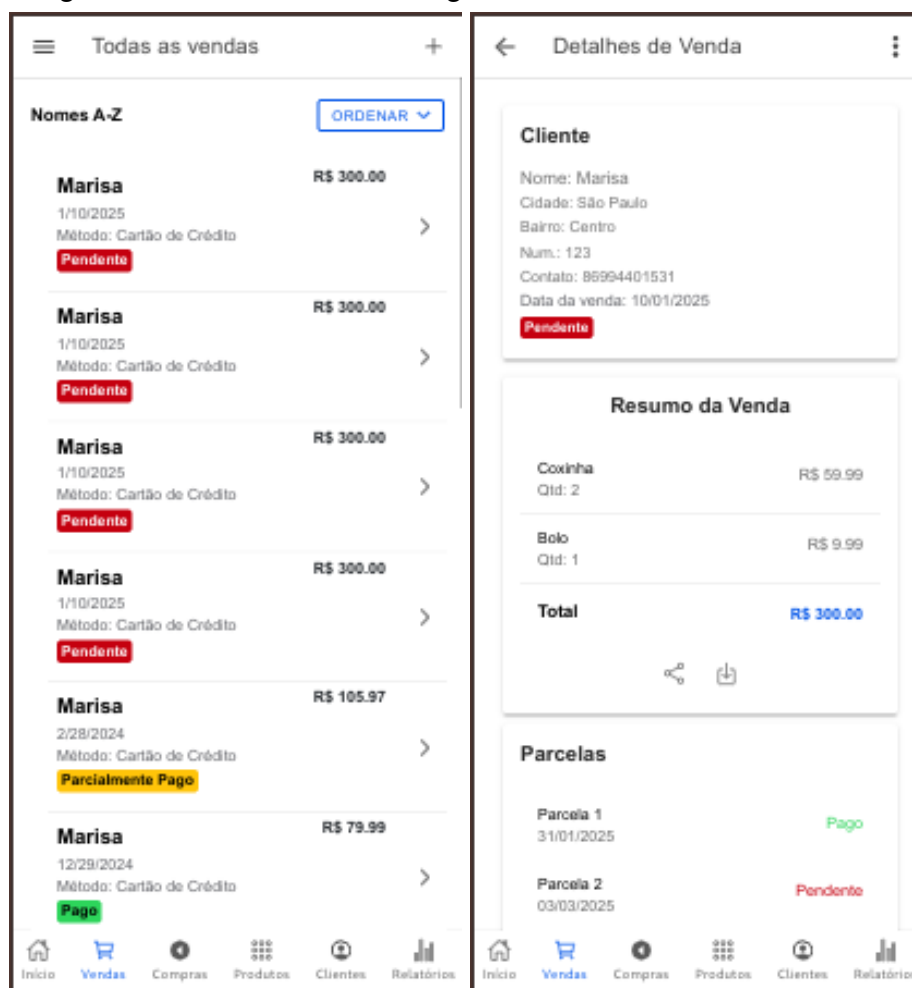


Fonte: De Autoria Própria

Também é possível aplicar filtros para buscar compras por fornecedor, data ou valor total, otimizando a gestão financeira e logística da empresa.

- **Listagem de Vendas:** Essa tela permite ao usuário visualizar e gerenciar as vendas registradas no sistema. O histórico de vendas é organizado de forma clara, exibindo o nome do cliente, a data da venda e o valor total da transação. Além disso, a interface utiliza uma codificação por cores para indicar o status da venda (à vista, parcelada ou pendente), facilitando a identificação rápida da situação de cada transação.

Figura 44 – Interface de Listagem e Detalhamento de Vendas



Fonte: De Autoria Própria

O detalhamento da venda permite ao usuário acessar informações mais específicas sobre cada transação, como a lista de produtos vendidos, quantidades e valores individuais. Esse recurso auxilia na conferência e no acompanhamento financeiro das operações comerciais.

- **Telas de Dashboards:** Após o login, o usuário tem acesso ao painel de controle (*dashboard*) do sistema, que fornece uma visão consolidada das principais métricas do negócio. Gráficos e indicadores visuais exibem informações essenciais, como total de vendas realizadas, estoque disponível e alertas para produtos com baixa quantidade.

Figura 45 – Dashboard com Indicadores de Desempenho



Fonte: De Autoria Própria

A interface do dashboard também permite a troca entre empresas cadastradas, facilitando a gestão de múltiplos negócios em um único ambiente. Além disso, o usuário pode acessar rapidamente relatórios detalhados e outras funcionalidades estratégicas para o monitoramento da empresa.

- **Relatórios de Produtos:** Nessa tela, o usuário pode gerar relatórios que mostram o desempenho dos produtos. Informações como produtos mais vendidos e alerta de itens com baixo estoque são apresentadas de forma visual e interativa. O sistema também permite a exportação dos relatórios para facilitar a análise externa.

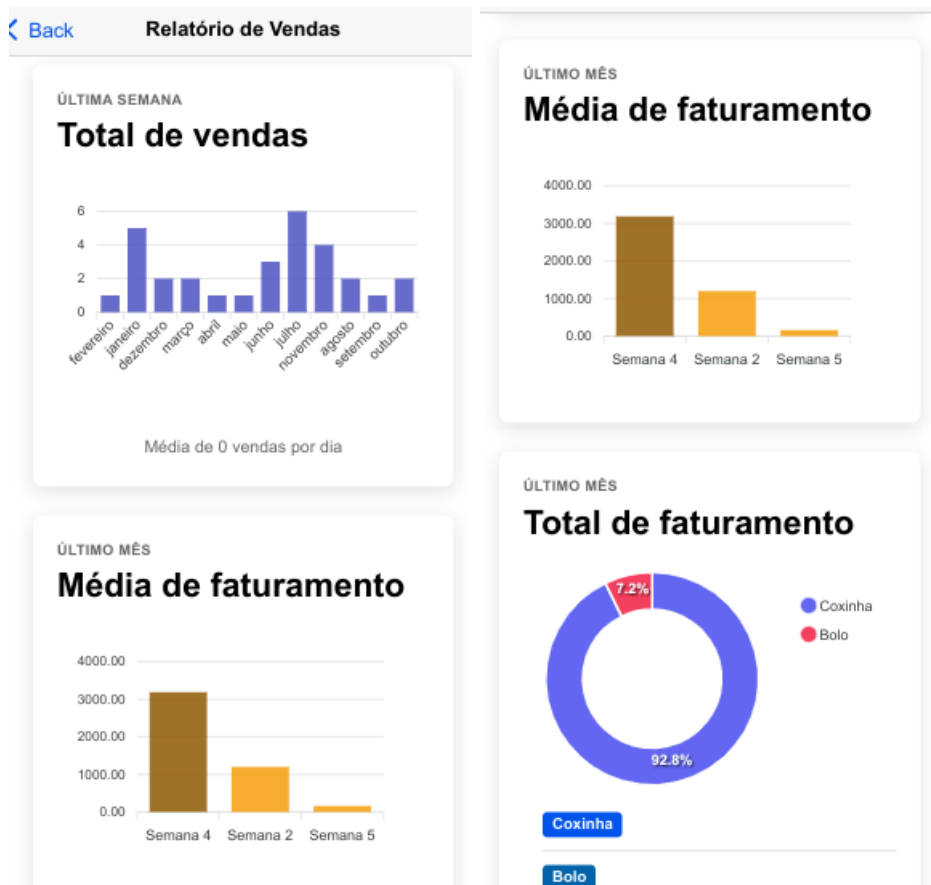
Figura 46 – Dashboard de Produtos



Fonte: De Autoria Própria

- **Relatórios de Vendas:** A tela de relatórios de vendas oferece uma visão completa sobre as transações realizadas. Nela, o usuário pode visualizar detalhes sobre as vendas em diferentes períodos, comparar o desempenho por cliente ou por produto, e gerar gráficos de evolução das vendas.

Figura 47 – Dashboard de Vendas



Fonte: De Autoria Própria

- **Relatórios de Empresas:**

A interface abaixo apresenta um relatório consolidado das empresas, destacando as movimentações financeiras e o desempenho de vendas no período selecionado.

Figura 48 – Interface do Relatório de Empresas

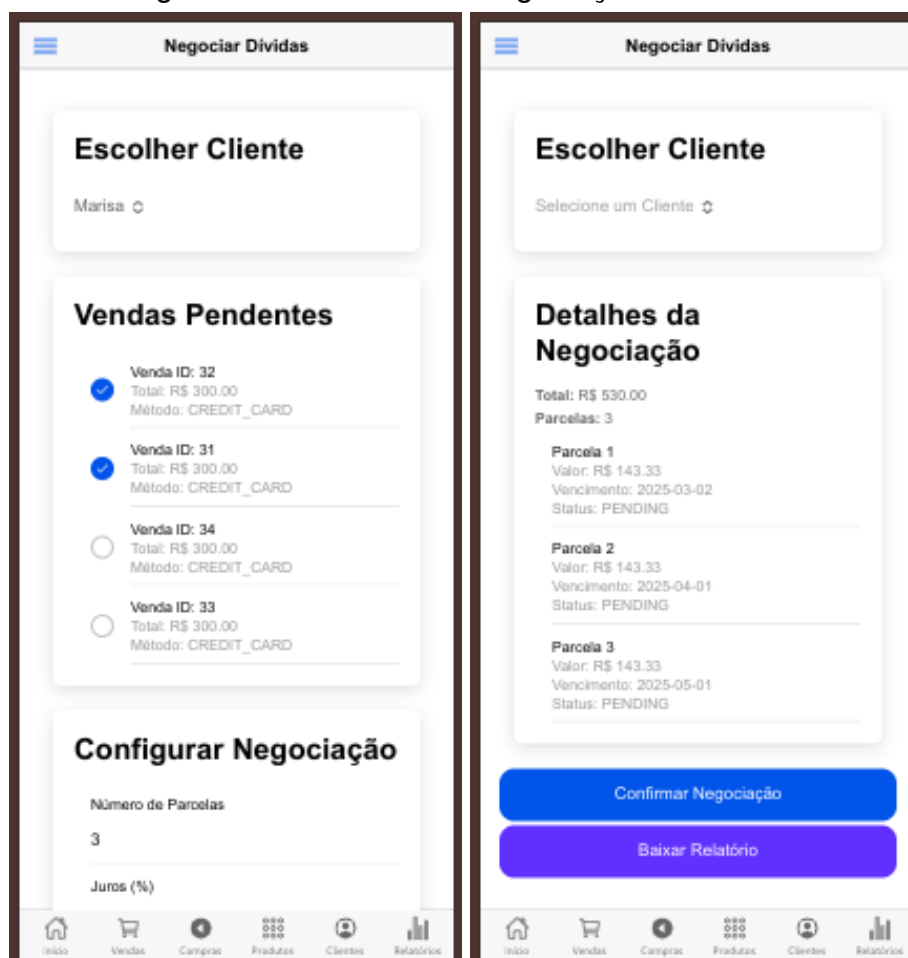


Fonte: De Autoria Própria

Esse relatório permite visualizar rapidamente as entradas e saídas financeiras das empresas cadastradas, auxiliando na análise de desempenho e tomada de decisões estratégicas.

- **Negociação de Dívidas:** A funcionalidade de negociação de dívidas permite que o usuário gerencie débitos em atraso de forma prática e transparente. Nesta tela, é possível selecionar um cliente inadimplente, visualizar seus débitos pendentes e configurar os parâmetros da negociação, como a aplicação de juros, quantidade de parcelas e descontos sobre o valor total.

Figura 49 – Interface de Negociação de Dívidas



Fonte: De Autoria Própria

O sistema apresenta um resumo da simulação da negociação, permitindo que o usuário revise e confirme os termos antes da formalização do acordo. Essa funcionalidade visa facilitar a recuperação de valores pendentes e oferecer flexibilidade tanto para o administrador quanto para os clientes.

5 CONCLUSÃO E TRABALHOS FUTUROS

Em síntese, o desenvolvimento do Easy Sales avançou para oferecer uma solução eficiente voltada ao gerenciamento de vendas, estoque, clientes e fornecedores, com ênfase em pequenas e médias empresas. O sistema destaca-se por sua usabilidade e integração com tecnologias modernas, como Ionic e Spring Boot, proporcionando uma experiência de usuário fluida tanto em dispositivos móveis quanto na web. Apesar de já estar implementando um conjunto robusto de funcionalidades que visam agilizar os processos de vendas e controle de estoque, o desenvolvimento continua, e há espaço para inovações e aprimoramentos. O foco em uma interface intuitiva, segura e eficiente é um dos principais objetivos do projeto, buscando aumentar a produtividade e a organização dos pequenos empreendedores.

O Easy Sales propõe-se como uma ferramenta essencial para a gestão empresarial, oferecendo relatórios detalhados e ferramentas de análise que visam auxiliar na tomada de decisões estratégicas. Sua modularidade e escalabilidade foram projetadas para permitir que o sistema seja facilmente expandido, de forma a garantir que ele possa evoluir conforme as necessidades dos usuários. Para os trabalhos futuros, há propostas para o aprimoramento de várias áreas, como a expansão dos relatórios analíticos, com o objetivo de incluir funcionalidades de relatórios preditivos de vendas e estoques, utilizando inteligência artificial. Isso permitirá que os empreendedores adaptem suas operações de maneira mais eficiente, respondendo rapidamente às demandas do mercado.

Dentre as propostas para o futuro, destacam-se a criação de um módulo de integração financeira, que permitirá conectar o Easy Sales a plataformas de contabilidade e sistemas bancários, facilitando a gestão financeira e a conciliação de pagamentos. Além disso, a integração com plataformas de e-commerce, como *Shopify* ou *WooCommerce*, abrirá novas oportunidades para os usuários expandirem seus negócios online. Outro aprimoramento relevante será a implementação de uma versão offline, permitindo que os usuários operem mesmo sem acesso à internet, com sincronização dos dados posteriormente. Por fim, a adição de opções de personalização avançada para relatórios e interfaces trará maior flexibilidade ao sistema, tornando-o ainda mais adaptável às necessidades individuais de cada usuário. Essas inovações visam fortalecer continuamente o Easy Sales, assegurando sua relevância em um mercado cada vez mais competitivo.

REFERÊNCIAS

- AGÊNCIA Brasileira de Desenvolvimento Industrial. "Maturidade digital das MPes ganha impulso em 2023, aponta pesquisa da ABDI.". [S.l.]: ABDI, 2023. Citado na página 15.
- BUXTON, B. **Sketching User Experiences: Getting the Design Right and the Right Design**. [S.l.]: Morgan Kaufmann, 2007. Citado 3 vezes nas páginas 25, 31 e 32.
- COHN, M. **Agile Estimating and Planning**. [S.l.]: Prentice Hall, 2005. Citado 2 vezes nas páginas 28 e 30.
- Confederação Nacional da Indústria (CNI). A digitalização como fator-chave para a competitividade das pmes. **Relatório CNI**, 2022. Disponível em: <<https://www.cni.org.br>>. Citado na página 16.
- CONTÁBEIS, P. **PMEs: 76% não possuem nenhum sistema para a gestão dos negócios**. 2024. Disponível em: <<https://www.contabeis.com.br/noticias/65816/pmes-76-nao-possuem-nenhum-sistema-para-a-gestao-dos-negocios/>>. Citado 3 vezes nas páginas 18, 19 e 20.
- FOWLER, M. **UML Distilled: A Brief Guide to the Standard Object Modeling Language**. [S.l.]: Addison-Wesley, 2003. Citado na página 32.
- GOODWIN, K. **Designing for the Digital Age: How to Create Human-Centered Products and Services**. [S.l.]: Wiley, 2009. Citado na página 31.
- MINISTÉRIO do Desenvolvimento, Indústria, Comércio e Serviços. "E-commerce no Brasil cresce 4% e alcança R\$ 196 bi em 2023.". [S.l.]: gov, 2024. Citado na página 15.
- SCHWABER, K. **Scrum: The Art of Doing Twice the Work in Half the Time**. [S.l.]: Crown Business, 2014. Citado 3 vezes nas páginas 28, 29 e 30.
- SCRUM.ORG. **Scrum Guide**. 2022. Disponível em: <<https://www.scrum.org/resources/scrum-guide>>. Citado na página 28.
- VALENTE, N. T. Z. Implementação de erp em pequenas e médias empresas: estudo de caso em empresa do setor da construção civil. 2004. 2004. Citado 2 vezes nas páginas 19 e 20.
- VERHEYEN, G. **Scrum – A Pocket Guide**. [S.l.]: Van Haren Publishing, 2020. Citado 3 vezes nas páginas 28, 29 e 30.