



UNIVERSIDADE PAULISTA

ICET - INSTITUTO DE CIÊNCIAS EXATAS E TECNOLOGIA

**CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E
DESENVOLVIMENTO DE SISTEMAS**

PROJETO INTEGRADO MULTIDISCIPLINAR

PIM III

Levantamento e análise de requisitos de um sistema de controle de fornecedores, produção e venda dos produtos (ou prestação de serviços), para uma fazenda urbana de uma startup focada em garantir inovação para área de segurança alimentar.

Nome

R.A

Matheus Rafael da Silva Jesus

N3573A1

Victor Hugo Rodrigues Barros Antunes

G71GEG3

**SÃO JOSÉ DOS CAMPOS – SP
JUNHO/2024**

	RA
Matheus Rafael da Silva Jesus	N3573A1
Victor Hugo Rodrigues Barros Antunes	G71GEG3

Levantamento e análise de requisitos de um sistema de controle de fornecedores, produção e venda dos produtos (ou prestação de serviços), para uma fazenda urbana de uma startup focada em garantir inovação para área de segurança alimentar.

Projeto Integrado Multidisciplinar (PIM) desenvolvido como exigência parcial dos requisitos obrigatórios à aprovação semestral no Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da UNIP (Universidade Paulista), orientado pelo corpo docente do curso.

RESUMO

Este estudo se concentrou no desenvolvimento de um sistema destinado a fazendas urbanas, com um enfoque na sua integração com os Objetivos de Desenvolvimento Sustentável (ODS) e os critérios ESG (Ambiental, Social e de Governança). Utilizou-se a metodologia ágil Scrum, juntamente com o desenvolvimento de diagramas de UML e entidade-relacionamento para a modelagem de dados. Esse processo foi fundamental para conectar ideias e apresentar as etapas de evolução do software. Uma pesquisa abrangente foi conduzida para examinar os processos das fazendas urbanas, seus impactos no Brasil e no mundo, bem como sua viabilidade após a pandemia de Covid-19. Concluiu-se que as fazendas urbanas representam uma abordagem tecnológica e sustentável para a agricultura, especialmente ao aproveitar a economia de espaço proporcionada por estufas verticais. Além disso, o estudo também contemplou o COP30 (30ª Conferência sobre Mudança do Clima), programado para ocorrer no Brasil em 2025, entre 10 e 21 de setembro, em Belém, no Pará. Os resultados evidenciaram que a metodologia ágil Scrum facilitou a colaboração entre a equipe de desenvolvimento e o cliente, garantindo uma resposta eficaz às demandas. Em conclusão, o sistema desenvolvido atendeu aos requisitos estabelecidos, promovendo práticas agrícolas sustentáveis alinhadas com os ODS e critérios ESG, o que pode atrair investidores interessados em uma solução inovadora e sustentável.

Palavras-Chave: Metodologia Ágil; ODS; ESG; Scrum; Fazenda Urbana; COP30.

SUMÁRIO

	Pág.
1. INTRODUÇÃO	5
2. PESQUISA	8
3. CARACTERIZAÇÃO DA EMPRESA (STARTUP)	13
4. DESENVOLVIMENTO DO PROJETO	15
5. CONSIDERAÇÕES FINAIS	
6. REFERÊNCIAS	

1. INTRODUÇÃO

No contexto atual de crescimento urbano acelerado e preocupações cada vez mais prementes sobre segurança alimentar e sustentabilidade ambiental, as fazendas urbanas emergem como uma resposta inovadora e promissora para os desafios enfrentados pelas cidades modernas. Como afirmou Michael Pollan (2012), autor e ativista alimentar, "Não acho que a agricultura urbana seja a única resposta, mas é um componente importante. Existem alternativas para os problemas alimentares nas cidades que não apenas dependem da industrialização da agricultura."

Nesse sentido, é crucial compreender o papel das fazendas urbanas como elementos integradores nos ambientes urbanos contemporâneos. Ao abordar a interseção entre produção de alimentos, desenvolvimento urbano e questões ambientais, as fazendas urbanas desempenham um papel multifacetado que vai além da simples agricultura. Elas se tornam espaços de educação, engajamento comunitário e regeneração ambiental, como apontado por autores como Karen Washington, líder comunitária e ativista alimentar.

Diante desse contexto, este trabalho visa explorar de forma abrangente e contextualizada o fenômeno das fazendas urbanas, destacando sua importância e potencialidades na construção de cidades mais sustentáveis e resilientes. As fazendas urbanas surgem como soluções inovadoras para uma série de problemas enfrentados pelas cidades modernas, tais como segurança alimentar, desperdício de alimentos, impacto ambiental, engajamento comunitário, resiliência urbana.

Portanto, este estudo propõe-se a investigar as fazendas urbanas como um fenômeno contemporâneo de grande relevância, fornecendo insights valiosos para políticas públicas, práticas agrícolas e iniciativas comunitárias voltadas para a construção de cidades mais sustentáveis e inclusivas.

1.1 Objetivo geral

O objetivo geral deste estudo é desenvolver um sistema que possa controlar processos de uma fazenda urbana, podendo assim trazer uma maneira mais tecnológica e inovadora para automatizar os processos internos.

1.2 Objetivos Específicos

- Utilizar metodologia ágil para que os processos tenham um bom entendimento e comunicação entre a equipe de desenvolvimento e o cliente (Incluí o stakeholders);
- Realizar testes de sistema e de usabilidade, avaliando a partir delas o resultado obtido;
- Realizar a partir de pesquisas o que é uma fazenda urbana, como funciona seus processos, suas características, qual o seu diferencial, a viabilidade e qual a dificuldade encontrada para que ela seja implementada;
- Coletar dados importantes para o desenvolvimento do banco de dados.

2. PESQUISA

As fazendas urbanas representam um avanço significativo na agricultura contemporânea, consistindo em plantações de vegetais, hortaliças ou frutas em ambientes urbanos, especialmente em grandes cidades e seus arredores. Este modelo de agricultura adota frequentemente estruturas verticalizadas, otimizando o espaço disponível.

O conceito de fazendas urbanas visa garantir a produção de alimentos de alta qualidade, em ambientes controlados e esterilizados, livres de pragas e com reduzida emissão de CO₂. Este enfoque tecnológico e controlado promete revolucionar a prática agrícola, oferecendo benefícios tanto para a saúde pública quanto para o meio ambiente.

Além disso, as fazendas urbanas têm o potencial de revitalizar áreas urbanas subutilizadas ou abandonadas, conferindo-lhes uma finalidade produtiva. Esta abordagem não apenas contribui para a melhoria da paisagem arquitetônica, mas também gera oportunidades de emprego local.

Um aspecto crucial das fazendas urbanas é sua capacidade de integrar práticas sustentáveis, incluindo a reciclagem de resíduos humanos e agrícolas para a produção de energia, como o metano. Este ciclo fechado de produção e reciclagem minimiza o impacto ambiental e reduz a dependência de recursos externos.

Segundo Barthel, Parker e Ernstson (2015), a Organização das Nações Unidas para Alimentação e Agricultura (FAO) estabeleceu categorias para a agricultura urbana, incluindo fazendas comunitárias, comerciais, institucionais e jardins comunitários. Esta classificação proporciona uma base estruturada para a compreensão e desenvolvimento contínuo deste setor emergente.

As fazendas urbanas têm emergido como uma solução inovadora para a produção de alimentos sustentáveis no Brasil, ganhando destaque em meio à crescente preocupação com a segurança alimentar e a necessidade de práticas agrícolas mais responsáveis.

A expansão das fazendas urbanas tem modificado significativamente a paisagem das cidades brasileiras, com um aumento notável nas operações de pequena escala, incluindo cultivos no solo, em telhados e terraços. Esta abordagem tem impacto global, beneficiando aproximadamente 810 milhões de pessoas em

todo o mundo e proporcionando acesso a alimentos para comunidades de baixa renda.

Além de sua função primária de produção de alimentos, as fazendas urbanas também desempenham papéis multifacetados, servindo como espaços de lazer, relaxamento, educação ambiental e terapia. No Brasil, o BeGreen é um exemplo proeminente, pioneiro no estabelecimento de fazendas urbanas desde 2014 e atualmente reconhecido como referência em várias cidades do país.

A origem da ideia de fazendas urbanas remonta aos anos 1960, ligada ao movimento ambientalista e à busca por um estilo de vida mais natural e comunitário. Grupos como os Green Guerrillas dos EUA foram pioneiros nessa jornada rumo a um sistema de produção alimentar autossustentável, visando democratizar o acesso aos alimentos para toda a comunidade.

O Sistema Nacional de Segurança Alimentar e Nutricional (SISAN) foi estabelecido com base na Lei nº 11.346, de 17 de setembro de 2006, consolidando-se como um marco importante para garantir a segurança alimentar no Brasil. Este sistema visa assegurar que todas as pessoas tenham acesso adequado a alimentos nutritivos em quantidade suficiente para suprir suas necessidades básicas.

De acordo com a definição da Organização das Nações Unidas para Alimentação e Agricultura (FAO), a segurança alimentar é alcançada quando todos têm acesso físico, social, econômico a alimentos seguros e nutritivos. No contexto brasileiro, o Conselho Nacional de Segurança Alimentar e Nutricional (CONSEA) define segurança alimentar como o direito de todos ao acesso regular e permanente a alimentos de qualidade, em quantidade suficiente, sem comprometer outras necessidades essenciais.

Os bancos de alimentos desempenham um papel crucial no combate à fome e à insegurança alimentar, recolhendo doações de alimentos que de outra forma seriam desperdiçados ao longo da cadeia produtiva. Essas organizações sem fins lucrativos trabalham para redistribuir alimentos entre aqueles que mais necessitam, evitando o desperdício e promovendo o uso responsável dos recursos alimentares.

No Brasil, a Rede Brasileira de Bancos de Alimentos (RBBA), estabelecida pelo decreto nº 10.490 de 17 de setembro de 2020, unifica bancos de alimentos públicos e privados com o objetivo de fortalecer sua atuação conjunta. Os bancos de alimentos aderentes têm acesso a benefícios, como a isenção de ICMS em doações

realizadas por doadores privados, incentivando a colaboração e ampliando o alcance dessas iniciativas.

No século XX, os países membros das Nações Unidas estabeleceram os 8 Objetivos de Desenvolvimento do Milênio (ODM), delineando metas a serem alcançadas entre 2000 e 2015. Este período testemunhou progressos significativos na redução da pobreza global, no acesso à educação e à água potável, indicando avanços positivos.

Os Objetivos de Desenvolvimento Sustentável (ODS) surgiram como sucessores dos ODM durante a Conferência RIO+20, realizada em junho de 2012. Neste evento, os 193 países membros da ONU discutiram o conceito de desenvolvimento sustentável, visando atender às necessidades da geração atual sem comprometer as gerações futuras.

Figura 1 – 17 objetivos da ODS



Fonte: Nações Unidas Brasil (2024)

Os ODS constituem um plano abrangente composto por 17 objetivos globais a serem alcançados até 2030. Essa agenda visa promover o crescimento e a cooperação entre todos os países em prol da sustentabilidade, abordando questões fundamentais como erradicação da pobreza, fome zero, saúde e bemestar, educação de qualidade, igualdade de gênero, entre outros aspectos essenciais para o desenvolvimento humano e ambiental.

O conceito de ESG, oriundo do inglês Environmental, Social and Governance (Ambiental, Social e Governança), é uma ferramenta essencial para avaliar o desempenho de uma organização em termos de responsabilidade socioambiental e ética.

O pilar ambiental concentra-se nas práticas e políticas adotadas para a preservação do meio ambiente, incluindo a redução de emissões de gases de efeito estufa, a gestão eficiente dos recursos naturais e o combate à poluição. A adoção de práticas sustentáveis é crucial para minimizar os impactos negativos das empresas no meio ambiente e garantir a segurança do planeta.

O pilar social aborda as ações voltadas para o bem-estar das pessoas, como a promoção da diversidade e inclusão, o respeito aos direitos humanos, a segurança dos trabalhadores e a contribuição para as comunidades onde a empresa atua. Garantir um ambiente saudável, seguro e justo para os colaboradores é uma prioridade.

O pilar de governança envolve práticas de gestão, transparência e ética corporativa, assegurando uma estrutura de liderança responsável e uma cultura empresarial sólida. Isso inclui a implementação de políticas anticorrupção, a prestação de contas e a transparência nas operações. Uma boa governança promove a confiança dos investidores, a reputação da empresa e o desenvolvimento sustentável do negócio.

Empresas têm cada vez mais reconhecido a importância do ESG, pois além de contribuir para o meio ambiente, elas ganham a confiança dos investidores, dos clientes e criam valores compartilhados. O ESG é benéfico para a empresa e para a sociedade como um todo, contribuindo para um mundo mais sustentável, ético e justo. As empresas que adotam o ESG são vistas de forma positiva no mercado, garantindo um futuro melhor e agregando valor aos seus stakeholders.

Os critérios de ESG (Ambiental, Social e Governança) e os Objetivos de Desenvolvimento Sustentável (ODS) têm sido cada vez mais demandados por investidores e stakeholders como resposta aos desafios globais contemporâneos. Empresas que adotam práticas de ESG em suas operações têm maior probabilidade de contribuir para a realização dos princípios dos ODS.

A adesão a práticas responsáveis, como a redução de emissões de carbono e a conservação dos recursos naturais, está alinhada com ODS específicos, como o

ODS 13 (Ação Contra a Mudança Global do Clima) e o ODS 15 (Vida Terrestre). Da mesma forma, ações focadas na promoção da igualdade de gênero e no fornecimento de condições de trabalho dignas, correspondentes ao ODS 5 (Igualdade de gênero) e ao ODS 8 (Trabalho Decente e Crescimento Econômico), contribuem para a dimensão social dos pilares ESG.

A governança corporativa sólida e transparente desempenha um papel fundamental no avanço dos ODS, pois empresas bem geridas têm maior capacidade de investir em iniciativas sustentáveis e socialmente responsáveis. Além disso, são percebidas como mais resilientes e preparadas para enfrentar desafios futuros.

A integração de práticas de ESG e a busca pela realização dos ODS não apenas fortalecem a reputação das empresas no mercado, mas também contribuem para um desenvolvimento sustentável e inclusivo, alinhado com as necessidades atuais e futuras da sociedade e do planeta.

A 30ª Conferência da ONU sobre Mudanças Climáticas (COP30), agendada para novembro de 2025 em Belém (PA), promete ser um evento de destaque, com a expectativa de atrair mais de 40 mil participantes, incluindo aproximadamente 7 mil membros da "família COP", representando equipes da ONU e delegações de países membros.

Para o presidente da república, Luiz Inácio Lula da Silva, a COP30 terá um significado singular, concentrando-se na importância da Amazônia no contexto brasileiro. Ele enfatiza a necessidade de discutir questões indígenas e compreender as realidades dos povos ribeirinhos.

O anúncio da COP30 intensifica os debates sobre mudanças climáticas em nível global, tradicionalmente centrados em estratégias de redução de emissões de gases de efeito estufa provenientes de indústrias, sob a perspectiva do Norte Global. No entanto, com a realização da conferência em Belém, a Amazônia e as políticas de conservação e redução de emissões relacionadas à região ocuparão um papel central pela primeira vez. Além disso, será destacado o papel do Sul Global nas discussões climáticas, ampliando a perspectiva global sobre a questão.

3. CARACTERIZAÇÃO DA EMPRESA (STARTUP)

As startups representam um fenômeno empresarial inovador que ganhou destaque nas últimas décadas. Segundo Eric Ries, autor de "The Lean Startup (2011)", as startups são definidas como "uma instituição humana projetada para criar novos produtos e serviços sob condições de extrema incerteza".

Nesse contexto, Steve Blank, autor de "The Startup Owner's Manual (2012)", enfatiza que as startups diferem das empresas tradicionais em sua abordagem ao mercado. Ele descreve uma startup como "uma organização temporária em busca de um modelo de negócios escalável e repetível".

Essas definições apontam para características-chave das startups, como a busca pela inovação, a disposição para assumir riscos e a necessidade de adaptação rápida em um ambiente de incerteza. As startups frequentemente operam em setores de tecnologia e tendem a priorizar o crescimento rápido e a escalabilidade.

No entanto, é importante observar que nem todas as empresas novas se enquadram na definição de startup. Como observado por Peter Thiel, autor de "Zero to One (2014)", "começar um novo negócio não é suficiente para ser uma startup; você precisa estar operando em um mercado de crescimento acelerado". Assim, as startups são mais do que simplesmente empresas recém-criadas; são entidades enxutas, ágeis e orientadas para a inovação, com o objetivo de criar e escalar novos produtos ou serviços em mercados dinâmicos e competitivos. Essas organizações estão constantemente buscando soluções criativas para problemas existentes ou identificando novas oportunidades de negócios, com um foco claro na disrupção e no crescimento acelerado.

A empresa contratante quer um sistema multifacetado, abrangendo três áreas-chave: controle de fornecedores, controle de produção e controle de vendas e prestação de serviços.

O sistema será desenvolvido com a capacidade de monitoramento de um sistema gerenciador de uma fazenda urbana pequena, com poucos funcionários(a) e poucas estufas de criação. Isso garanti a eficiência da fazenda com logística e cultivo.

Na esfera dos requisitos funcionais, é imprescindível a implementação de funcionalidades como cadastro de usuários, fornecedores e pedidos.

Adicionalmente, é crucial permitir a visualização, modificação, exclusão e desativação/ativação de informações relacionadas a esses elementos, de acordo com sua permissão no sistema, seguindo a hierarquia proposta pela contratante. O monitoramento da produção também é um requisito essencial.

Em relação aos requisitos não funcionais, destaca-se a necessidade de estabelecer uma hierarquia de acesso baseada no modelo CRUD (Create, Read, Update, Delete), garantindo que cada usuário tenha permissões apropriadas para suas funções. Além disso, é fundamental assegurar a conformidade com a Lei Geral de Proteção de Dados (LGPD), garantindo a privacidade e segurança das informações manipuladas pelo sistema.

Quanto à plataforma, o sistema deverá ser desenvolvido para acesso via mobile, desktop e WEB, com restrições específicas para acessos móveis, visando mitigar possíveis vulnerabilidades. Finalmente, o software será projetado para operar nos sistemas operacionais Android e Windows, garantindo compatibilidade e acessibilidade aos usuários.

Essas informações são cruciais para a definição dos requisitos do sistema, permitindo a identificação precisa das necessidades da empresa e orientando o desenvolvimento de um software que atenda de maneira eficiente e eficaz às suas demandas específicas.

4. DESENVOLVIMENTO DO PROJETO

Nos últimos anos, o surgimento e a expansão das fazendas urbanas têm gerado um impacto significativo na economia local e global. Essas iniciativas agrícolas dentro de áreas urbanas estão redefinindo a maneira como produzimos e consumimos alimentos, e estão contribuindo para uma série de mudanças econômicas positivas.

Estimulando as economias locais, as fazendas frequentemente utilizam terrenos abandonados ou edifícios industriais reutilizados, o que contribui para a revitalização de áreas urbanas degradadas. Além disso, ao produzir alimentos localmente, as fazendas reduzem a dependência de importações e contribuem para o fortalecimento da economia local, ao mesmo tempo em que reduzem os custos de transporte e armazenamento.

O surgimento das fazendas urbanas tem promovido o empreendedorismo, incentivando a criação de novas startups e empresas inovadoras no setor agrícola. Muitas dessas empresas desenvolvem tecnologias e soluções voltadas para a agricultura urbana, gerando oportunidades de negócios e investimentos em pesquisa e desenvolvimento.

As áreas urbanas podem reduzir os custos de produção de alimentos, principalmente em áreas onde o preço do terreno é elevado, e onde o espaço é consideravelmente pequeno para reproduzir alimentos com qualidade. Além disso, ao produzir alimentos localmente e de forma sazonal, as fazendas urbanas podem reduzir o desperdício de alimentos ao eliminar a necessidade de transporte de longa distância e o armazenamento prolongado.

Ao produzir alimentos localmente, as fazendas urbanas aumentam a segurança alimentar das comunidades urbanas, garantindo o acesso a alimentos frescos e saudáveis. Isso é especialmente importante em áreas onde o acesso a alimentos de qualidade é limitado devido a barreiras econômicas ou geográficas.

O surgimento e crescimento das startups têm desempenhado um papel fundamental na economia global, trazendo consigo uma série de impactos positivos que impulsionam o desenvolvimento econômico e a inovação em diversos setores.

Abaixo estão alguns dos principais impactos econômicos das startups:

- Startups são conhecidas por serem grandes geradoras de empregos, especialmente em setores de alta tecnologia e inovação. Elas oferecem oportunidades de trabalho para uma ampla gama de profissionais, desde engenheiros e desenvolvedores de software até profissionais de marketing e vendas. Além disso, as startups muitas vezes promovem uma cultura de inovação e empreendedorismo, incentivando a criação de novos negócios e empregos adicionais na comunidade.
- As startups frequentemente atraem investimentos, tanto de capital de risco quanto de investidores anjos, que injetam capital na economia local e global. Esses investimentos não apenas impulsionam o crescimento das próprias startups, mas também contribuem para o crescimento de toda a cadeia de valor, incluindo fornecedores, parceiros e clientes.
- O surgimento de startups cria um ambiente favorável ao empreendedorismo e à criação de novos negócios. Elas estimulam a competição e a diversificação econômica, incentivando outros empreendedores a buscarem oportunidades de negócios inovadores e disruptivos. Além disso, as startups muitas vezes compartilham recursos, conhecimentos e experiências com outros empreendedores e empresas, enriquecendo o ecossistema empreendedor como um todo.

A gestão eficiente é fundamental para o sucesso de uma fazenda urbana, pois enfrenta desafios únicos decorrentes do ambiente urbano. Nesse contexto, um sistema de gerenciamento pode desempenhar um papel crucial ao otimizar processos, monitorar dados e facilitar a tomada de decisões. O sistema proposto visa atender a essa necessidade, fornecendo uma solução integrada para gerenciar todos os aspectos da fazenda urbana, desde o plantio e cultivo até a distribuição e comercialização dos produtos.

As fazendas urbanas enfrentam uma série de desafios, incluindo gerenciamento de estoque, monitoramento de plantio e gerenciamento de pedidos. O sistema proposto visa abordar essas necessidades, oferecendo ferramentas e

funcionalidades específicas para superar os desafios enfrentados pelas fazendas urbanas.

O sistema de gerenciamento para fazenda urbana é uma plataforma abrangente projetada para facilitar todas as etapas do processo agrícola urbano. Ele inclui módulos para planejamento de cultivos, gestão de estoque e recursos, monitoramento de condições ambientais, rastreamento de produção e vendas, entre outros. O sistema é altamente personalizável e adaptável às necessidades específicas de cada fazenda urbana, oferecendo uma interface intuitiva e fácil de usar.

A implementação do sistema de gerenciamento trará uma série de benefícios para as fazendas urbanas. Isso inclui aumento da eficiência operacional, redução de custos de produção, melhoria na qualidade dos produtos, maior rastreabilidade e transparência na cadeia de suprimentos, e capacidade de tomar decisões mais informadas com base em dados em tempo real. Esses benefícios contribuirão para a sustentabilidade e o sucesso a longo prazo das fazendas urbanas.

4.1 Método Ágil

O projeto do sistema foi desenvolvido com base na metodologia ágil Scrum. Contando com dois participantes, decidimos que cada um assumiria um papel específico no desenvolvimento do projeto. Victor Hugo foi designado como Product Owner (PO), enquanto Matheus Rafael assumiu o papel de Scrum Master. Devido à ausência de outros participantes, o papel de Developer Team não foi preenchido. A divisão de papéis no Scrum foi, portanto, a seguinte:

Tabela 1 – Tabela de Separação SCRUM

SCRUM	
Link do Projeto: https://github.com/math20122004/pim_work_2024	
Matheus Rafael da Silva Jesus	Scrum Master
Victor Hugo Rodrigues Barros Antunes	PO

Fonte: O Autor (2024)

Após a definição dos papéis e a realização de pesquisas iniciais, foram estabelecidos os requisitos do cliente (o que o cliente necessita) e os requisitos técnicos (o que é necessário desenvolver para atender às necessidades do cliente). Os requisitos técnicos foram divididos em duas categorias: requisitos funcionais e requisitos não funcionais. Como mostra a tabela abaixo:

Tabela 2 – Tabela Requisitos

REQUISITOS DO CLIENTE
R1. Sistema para controle de fornecedores;
R2. Sistema para controle de produção;
R3. Sistema para controle de vendas e prestação de serviços.
REQUISITOS FUNCIONAIS
R1. Cadastro de usuários no sistema. O sistema terá uma interface para cadastros de novos usuários. Apenas usuários com permissão.
R2. Cadastro de fornecedores. O sistema terá que cadastrar novos fornecedores no sistema para controle. Apenas com permissão.
R3. Visualização de fornecedor. Para a visualização de fornecedores, iremos usar um filtro direto, para que o usuário que procurar coloque o CNPJ ou nome da empresa, e assim possa verificar as informações que deseja.
R4. Modificar dados dos fornecedores cadastrados. O sistema irá permitir que usuários com as devidas autorizações possam modificar dados do fornecedor que deseja. Apenas com permissão.
R5. Desativação e ativação de fornecedores cadastrados. O sistema não irá permitir a exclusão de fornecedor, e sim a desativação e ativação no sistema. Apenas com permissão.
R6. Monitoramento de produção. O sistema terá um painel contendo informações da produção.
R7. Inclusão de pedidos. Usuários com as devidas permissões poderão incluir pedidos no sistema. Apenas com permissão.
R8. Visualização de pedidos. Os usuários poderão visualizar os pedidos que foram incluídos e poderá usar o filtro para ir direto para o pedido que deseja.
R9. Modificar dados dos pedidos incluídos. Usuários com as permissões necessárias poderão modificar dados dos pedidos. Apenas com permissão.
REQUISITOS NÃO FUNCIONAIS
R1. Hierarquia. O sistema terá níveis de acessos. O acesso atribuído a um certo usuário dependerá do que for incluído pelo usuário master, que terá todas as permissões do sistema. A hierarquia será separa por CRUD (Create, Read, Update, Delete) para cada opção.
R2. LGPD. Os dados coletados e tratados pelo sistema, seguiram as leis de LGPD.
R3. Plataforma. O sistema será desenvolvido pelas plataformas mobile, desktop e WEB, mas acessos via mobile serão restritos a fim de evitar pessoas má intencionadas e vazamentos de dados.
R4. Sistema operacional. O sistema será desenvolvido para os sistemas operacionais Android e Windows.

Fonte: O Autor (2024)

Com os requisitos definidos, iniciamos a elaboração do Backlog, uma parte essencial do método Scrum. O Backlog envolve a organização dos requisitos em Sprints, cada um com datas previstas para conclusão. A priorização dos processos pode variar de acordo com sua importância.

Após a organização das Sprints, criamos a Sprint Backlog, um modelo detalhado de cada Sprint. Este modelo inclui uma divisão passo a passo de como realizar cada requisito estabelecido. Foi feito um planejamento do tempo estimado para a execução de cada passo.

Essa abordagem metódica e organizada garantiu que o desenvolvimento do sistema seguisse um cronograma claro e eficiente, focando sempre na qualidade e no cumprimento dos requisitos do cliente. Como mostra os dados de Backlog e do desenvolvimento do estudo:

Tabela 3 – Tabela BackLog

BACKLOG		
SPRINT 1 (24/03 - 30/03)		
Pesquisa do tema	OK	Matheus Rafael
SPRINT 2 (31/03 - 06/04)		
Requisitos do Cliente	OK	Matheus Rafael
Viabilidade	OK	Victor Hugo
Requisitos do Sistema	OK	Matheus Rafael
Criação Backlog	OK	Matheus Rafael
SPRINT 3 (07/04 - 13/04)		
Começo Protótipo de Tela	OK	Matheus Rafael
Início Diagrama de Caso de Uso	OK	Matheus Rafael
Início Recursos Humanos Utilizados	OK	Victor Hugo
SPRINT 4 (14/04 - 20/04)		
Fim Diagrama de Caso de Uso	OK	Matheus Rafael
Início Diagrama de Classe	OK	Matheus Rafael
SPRINT 5 (21/04 - 27/04)		
Fim Diagrama de Classe	OK	Matheus Rafael
Início Diagrama de Sequência	OK	Matheus Rafael
Início Diagrama de Implantação	OK	Victor Hugo
SPRINT 6 (28/04 - 04/05)		
Fim Diagrama de Sequência	OK	Matheus Rafael
Fim Diagrama de Implantação	OK	Victor Hugo
Início Diagrama Entidade-Relacionamento	OK	Matheus Rafael
Dicionário de Dados	OK	Matheus Rafael
SPRINT 7 (05/05 - 11/05)		
Fim Diagrama Entidade-Relacionamento (E	OK	Matheus Rafael
Script Banco de Dados	OK	Matheus Rafael
Roteiro de Testes Banco de Dados	OK	Matheus Rafael
Finalização Protótipo de Telas	OK	Matheus Rafael
Início PoC em Modo Console	OK	Matheus Rafael
SPRINT 8 (12/05 - 18/05)		
Manual do Sistema	OK	Victor Hugo
Planilha de Testes	OK	Victor Hugo
Homologação do Sistema	OK	Victor Hugo
Apresentação Power Point	OK	Victor Hugo / Matheus Rafael

Fonte: O Autor (2024)

A Sprint Backlog foi gerenciada através do aplicativo GitHub, utilizando a aba de projetos. Essa abordagem facilitou o desenvolvimento e automatizou diversas etapas, permitindo um controle mais eficiente do progresso. A utilização do GitHub não apenas organizou as tarefas, mas também gerou dados visuais através de gráficos, proporcionando uma visão clara do andamento e facilitando a tomada de decisões durante o desenvolvimento. [Sprint Backlog](#).

A ferramenta do GitHub desempenhou um papel crucial no versionamento e na direção do projeto, permitindo que as etapas concluídas fossem postadas e rastreadas no site. Essa funcionalidade facilitou significativamente o desenvolvimento do projeto, promovendo uma comunicação clara e um entendimento compartilhado entre os membros da equipe. O uso do GitHub não só organizou as tarefas, mas também assegurou que todas as modificações e atualizações fossem registradas de maneira estruturada, proporcionando um histórico detalhado do progresso do projeto e melhorando a coordenação e a colaboração entre os participantes.

Nosso grupo realizou reuniões diárias (Dailys) para discutir assuntos relacionados à sprint da semana. Essas reuniões são fundamentais para entender melhor os próximos passos, estabelecer critérios e definir objetivos claros. Durante as Dailys, frequentemente surgiram mudanças de postura, ajustes nos requisitos e atualizações no sprint backlog, garantindo que todos estivessem alinhados e prontos para as próximas tarefas.

No final de cada Sprint fizemos uma Sprint Review, revisando o que foi feito, incrementando mais coisas no backlog, ou adaptando os processos do produto.

Realizamos Sprint Retrospective para discutir a sprint passada. Essas reuniões são essenciais para aprimorar as sprints futuras, permitindo identificar áreas de melhoria e debater pontos que precisam ser ajustados. Além de ajudar a melhorar a Sprint passada.

4.2 Diagramas UML

A modelagem do sistema orientado a objeto foi realizada utilizando a UML (Unified Modeling Language), uma linguagem padrão que facilita a visualização, especificação, construção e documentação de sistemas de software. A UML é composta por 14 tipos de diagramas que se complementam na modelagem, divididos em dois grupos principais: diagramas estruturais e diagramas comportamentais.

Foram utilizados 4 diagramas: Caso de uso, Classe, Sequência e Implementação.

O primeiro é o diagrama de casos de uso, que pertence ao grupo comportamental. Esse diagrama foi essencial para mapear as interações entre os usuários e o sistema, definindo claramente as funcionalidades e os requisitos do sistema sob a perspectiva dos usuários.

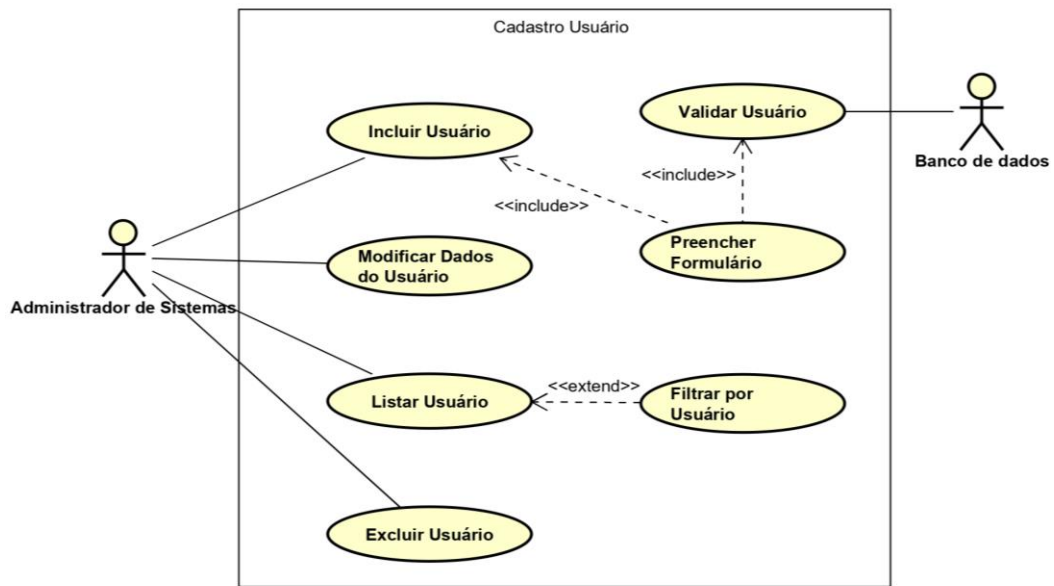
O segundo diagrama utilizado foi o diagrama de classes, pertencente ao grupo estrutural. Este diagrama foi crucial para modelar a estrutura estática do sistema, detalhando as classes, seus atributos, métodos e os relacionamentos entre elas.

Terceiro diagrama é o de sequência é fundamental para capturar a dinâmica das interações entre os componentes de um sistema ao longo do tempo. Eles ilustram como os objetos colaboram em um cenário específico para realizar uma tarefa, detalhando a ordem das mensagens trocadas entre eles. Isso permite que desenvolvedores e arquitetos entendam a lógica de execução dos processos, identifiquem possíveis gargalos ou falhas de comunicação, e assegurem uma visão unificada do fluxo de interação no sistema.

Por outro lado, diagramas de implementação focam na organização física do sistema. Eles mostram como os componentes do software são distribuídos em termos de arquivos, bibliotecas e módulos, bem como a infraestrutura de hardware em que o software será executado.

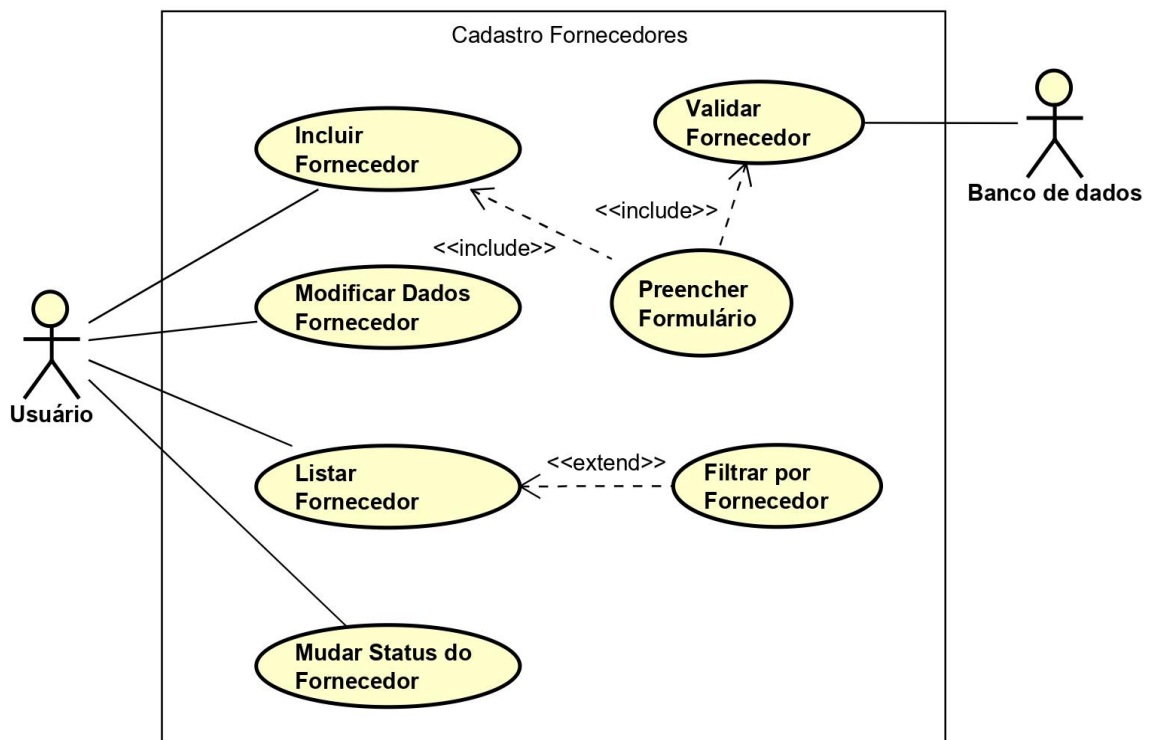
O diagrama de casos de uso foi dividido em partes menores e um inteiro com o intuito de melhorar a visualização. O processo foi realizado a partir de um software de modelagem UML chamado Atah.

Figura 2 – Diagrama de Caso de Uso Cadastro de Usuário



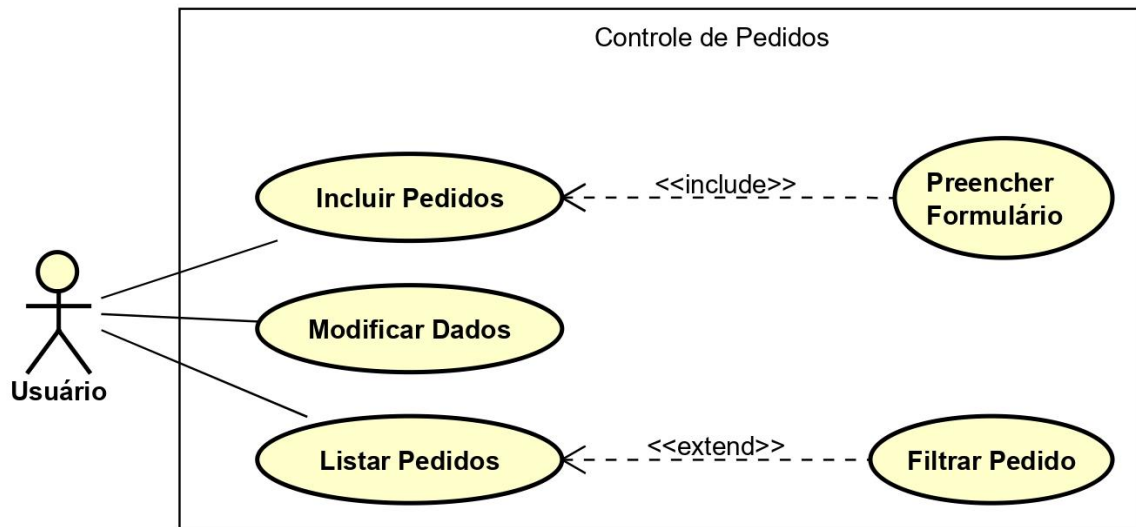
Fonte: O Autor (2024)

Figura 3 – Diagrama de Caso de Uso Cadastro de Fornecedores



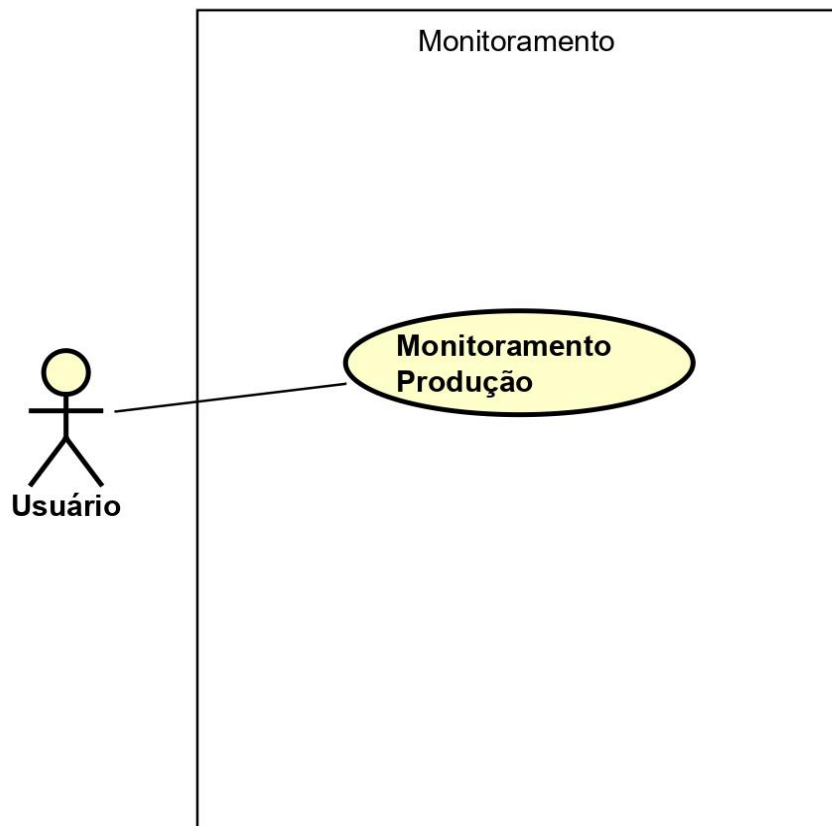
Fonte: O Autor (2024)

Figura 4 – Diagrama de Caso de Uso Controle de Pedidos



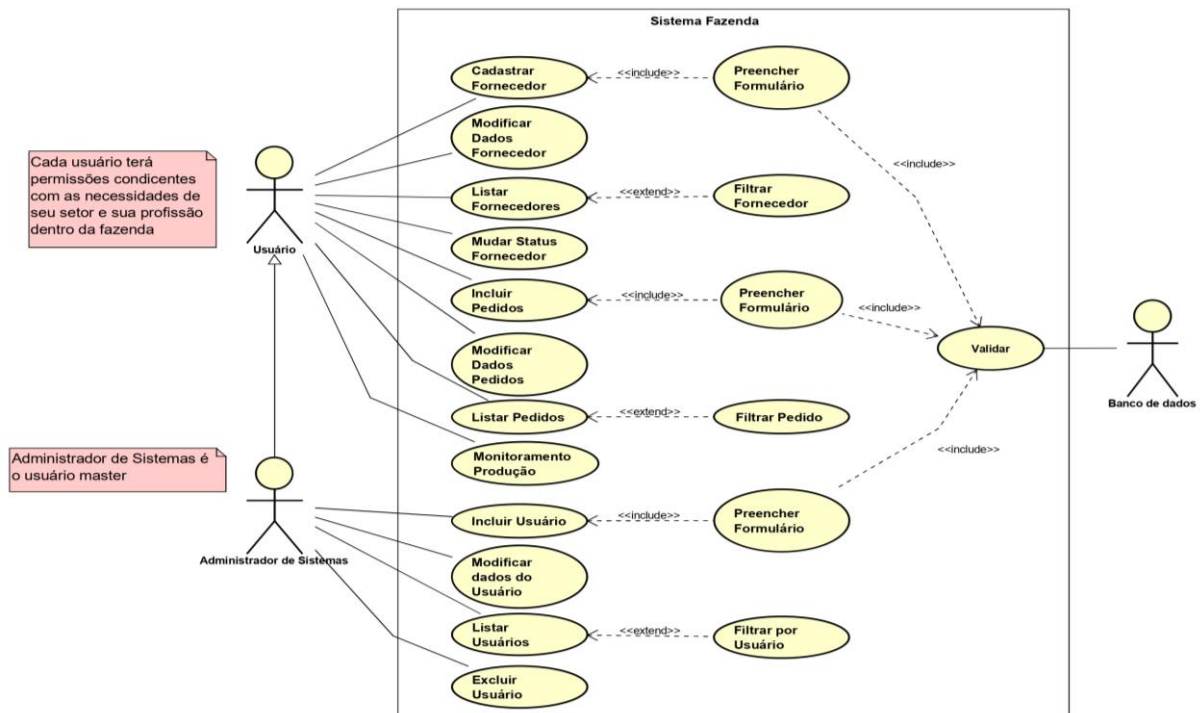
Fonte: O Autor (2024)

Figura 5 – Diagrama de Caso de Uso Monitoramento



Fonte: O Autor (2024)

Figura 6 – Diagrama de Caso de Uso Geral

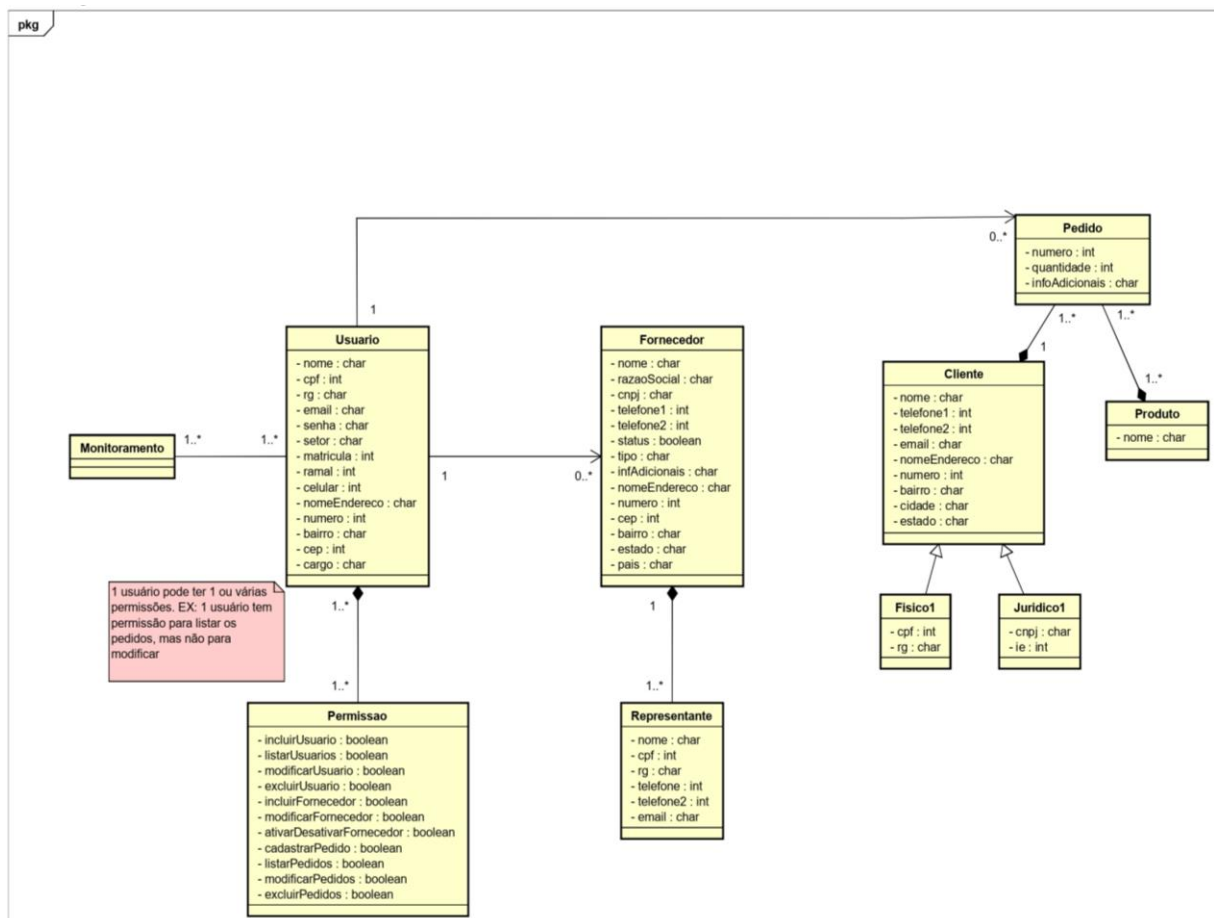


Fonte: O Autor (2024)

O diagrama de casos de uso é essencial na modelagem de sistemas orientados a objetos, especialmente na definição inicial dos requisitos. Ele captura as funcionalidades do sistema a partir da perspectiva dos usuários finais, conhecidos como atores, e descreve as interações entre esses atores e o sistema. Segundo Ivar Jacobson, um dos criadores da UML, "um caso de uso é uma sequência de ações, incluindo variantes, que um sistema pode realizar, interagindo com atores externos do sistema" (Jacobson, I., Booch, G., & Rumbaugh, J., 1999). Esse diagrama proporciona uma visão clara das funcionalidades necessárias, facilitando a comunicação entre desenvolvedores e stakeholders e assegurando que todas as interações significativas sejam consideradas. Os diagramas de casos de uso ajudam a identificar e organizar requisitos funcionais, promovendo uma compreensão compartilhada do sistema e alinhando expectativas. Além disso, eles permitem a identificação precoce de possíveis problemas e ambiguidades nos requisitos. Em resumo, o diagrama de casos de uso é uma ferramenta crucial na engenharia de software, garantindo que o sistema atenda às necessidades dos usuários e aumentando a eficiência do processo de desenvolvimento.

Por outro lado, o diagrama de classe é utilizado para representar a estrutura estática do sistema. Ele descreve as classes do sistema, seus atributos, métodos e os relacionamentos entre essas classes. Segundo Grady Booch, um dos criadores da UML, "um diagrama de classe é uma representação estática e estrutural de classes e seus relacionamentos" (Booch, G., Rumbaugh, J., & Jacobson, I., 2005). Esse diagrama proporciona uma visão clara da arquitetura do sistema, ajudando os desenvolvedores a entenderem a estrutura do código e a relação entre os diferentes componentes. Os diagramas de classe são fundamentais para o design e a implementação de sistemas complexos, pois permitem uma organização eficiente do código e facilitam a comunicação entre os membros da equipe de desenvolvimento. Eles ajudam a identificar as classes principais do sistema, seus atributos e métodos, e os relacionamentos entre essas classes, proporcionando uma base sólida para o desenvolvimento do software. Em resumo, o diagrama de classe é uma ferramenta poderosa na engenharia de software, fornecendo uma representação visual da estrutura estática do sistema e auxiliando no processo de design e implementação.

Figura 7 – Diagrama de Classe



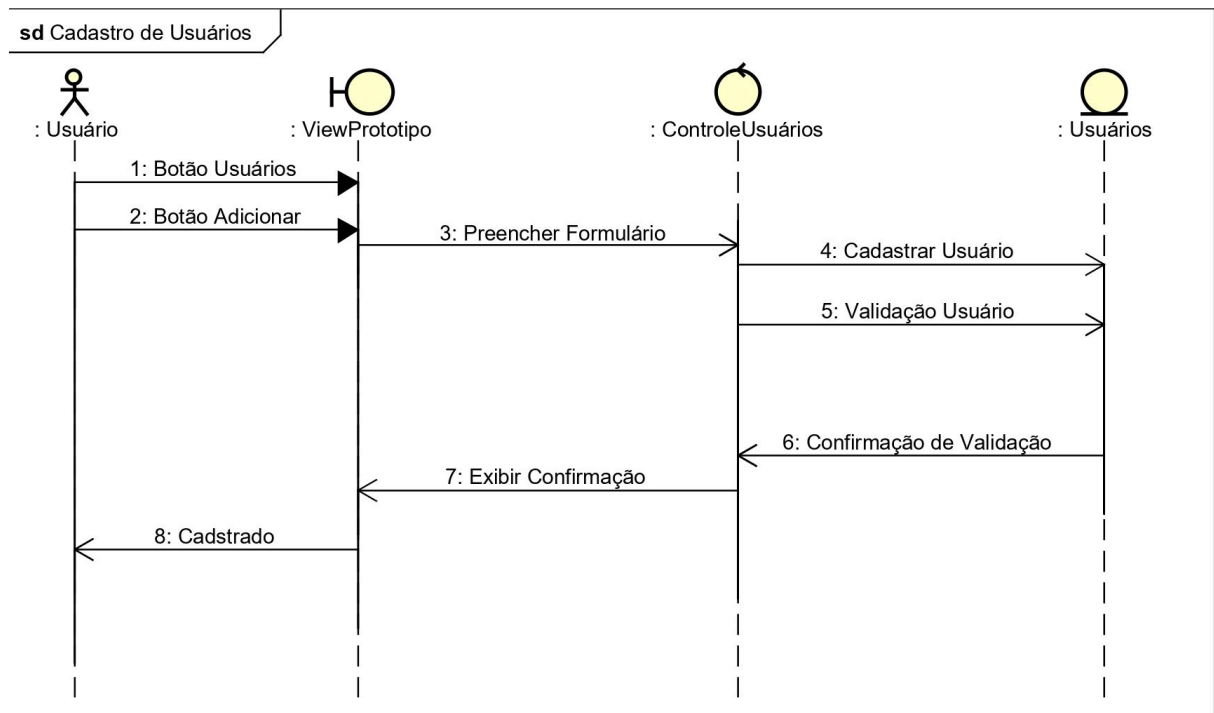
Fonte: O Autor (2024)

Um diagrama de sequência é uma ferramenta visual usada para descrever como diferentes componentes de um sistema interagem entre si ao longo do tempo. Ele é especialmente útil no desenvolvimento de software, pois mostra claramente a ordem das mensagens trocadas entre objetos ou componentes para realizar uma função específica.

No diagrama de sequência, você começaria identificando os atores (quem ou o que está interagindo) e os objetos envolvidos, como o "Usuário", a "Interface" (ou View), o "Controlador" (Controller), e o "Banco de Dados" (ou Model).

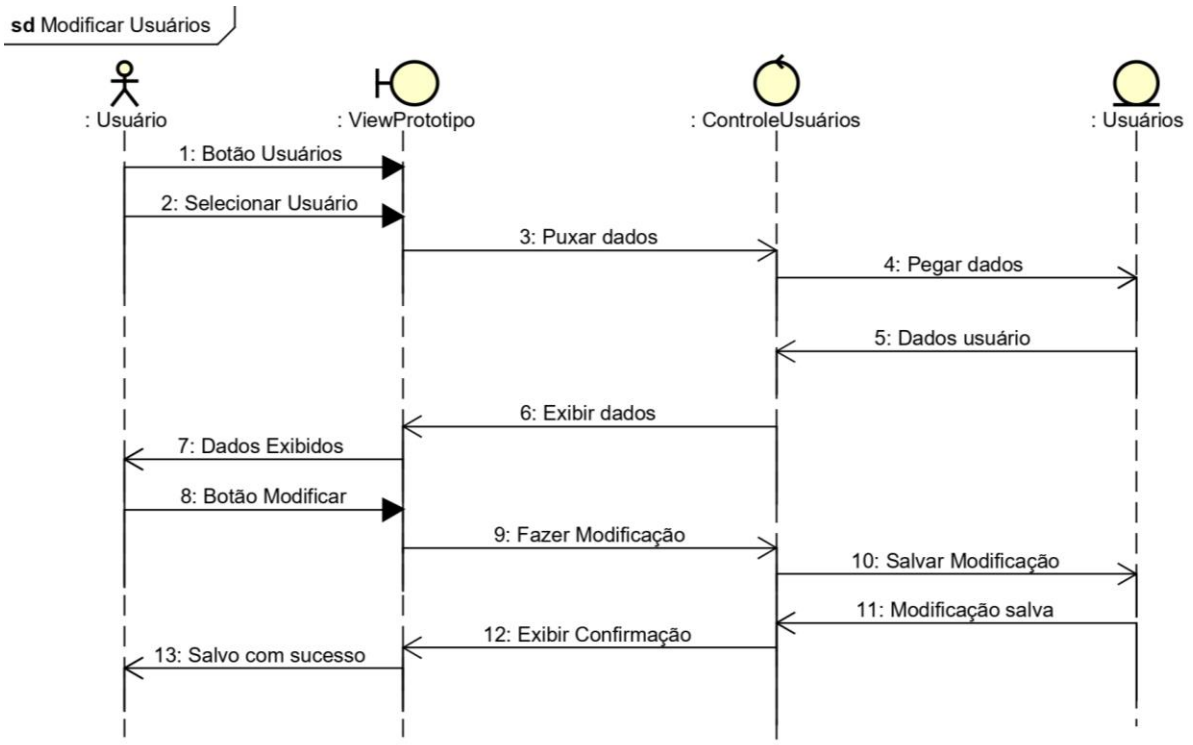
O diagrama é composto por linhas verticais chamadas de "lifelines" que representam a vida dos objetos durante a interação. As mensagens trocadas entre esses objetos são representadas por setas horizontais. Existem dois tipos principais de mensagens: síncronas e assíncronas. Mensagens síncronas são aquelas em que o remetente espera uma resposta antes de continuar, enquanto mensagens assíncronas permitem que o remetente continue sem esperar.

Figura 8 – Diagrama de Sequência Incluir Usuário



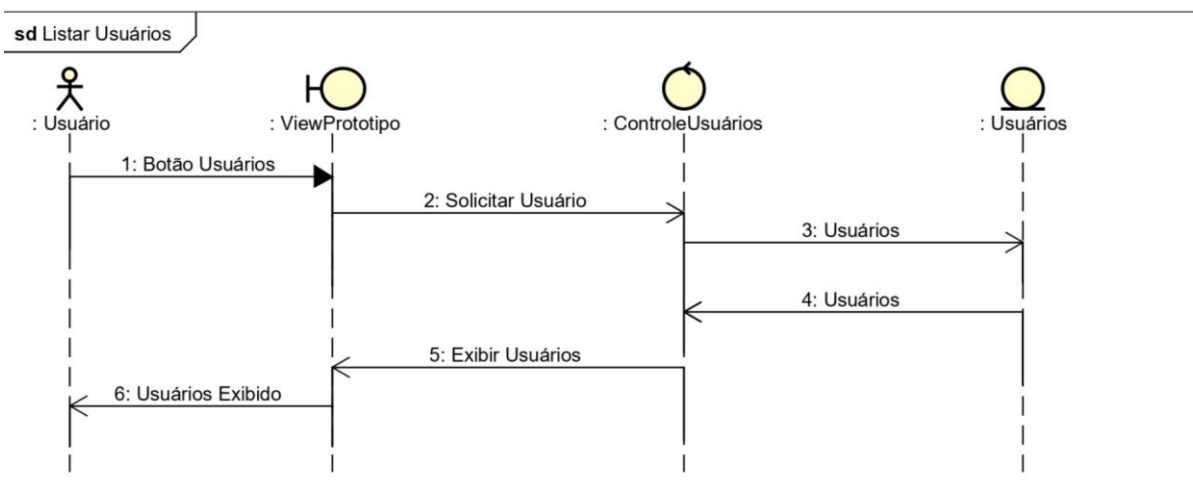
Fonte: O Autor (2024)

Figura 9 – Diagrama de Sequência Modificar Usuário



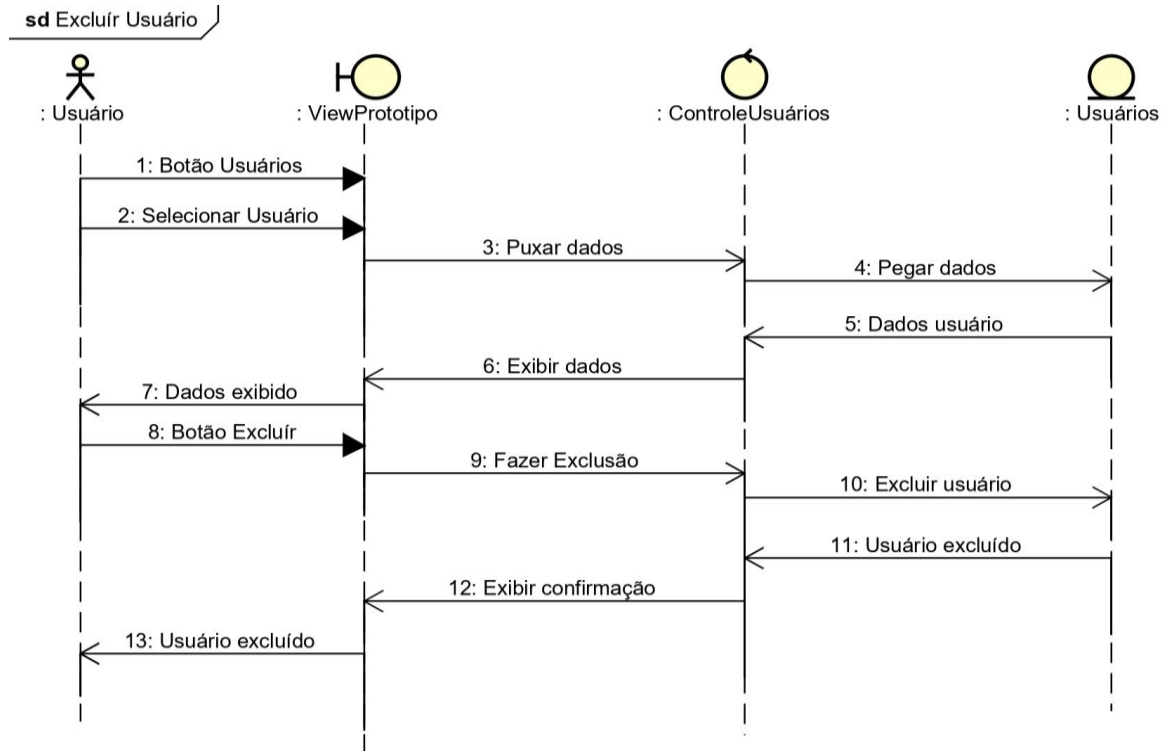
Fonte: O Autor (2024)

Figura 10 – Diagrama de Sequência Listar Usuários



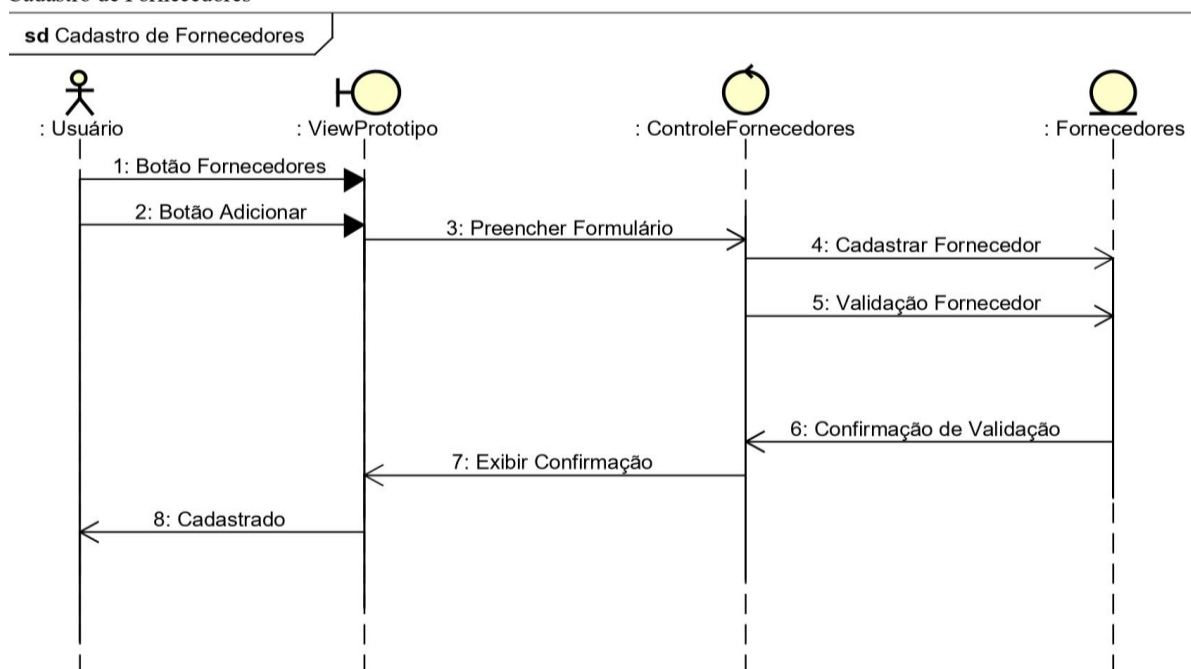
Fonte: O Autor (2024)

Figura 11 – Diagrama de Sequência Excluir Usuário



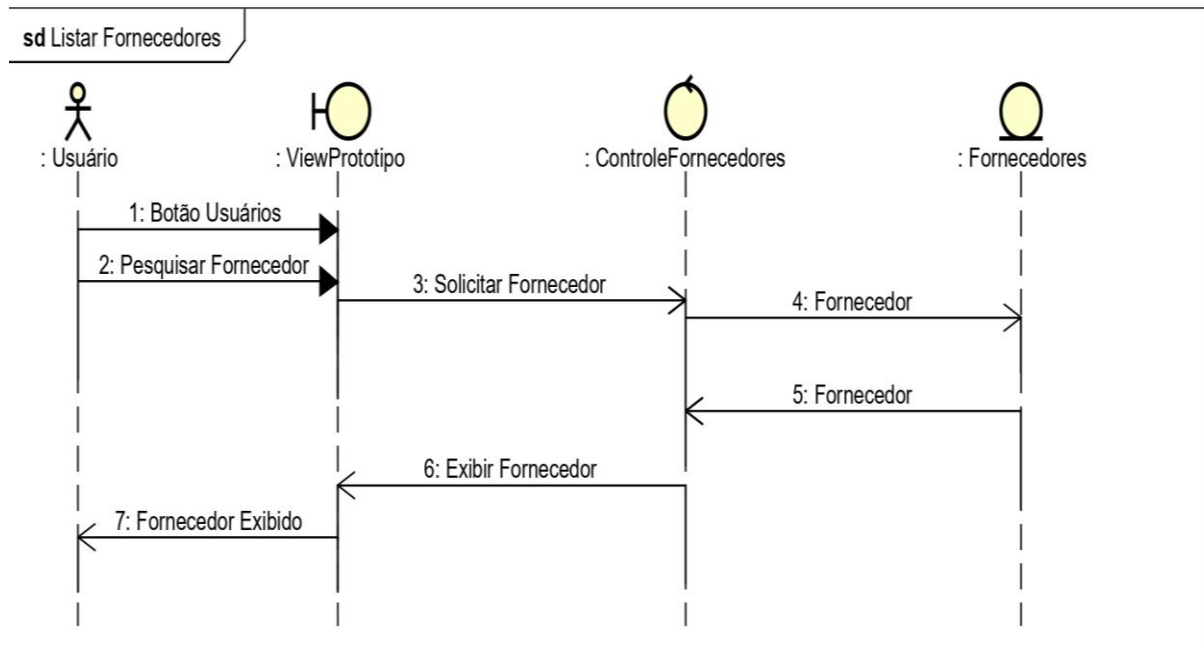
Fonte: O Autor (2024)

Figura 12 – Diagrama de Sequência Incluir Fornecedor



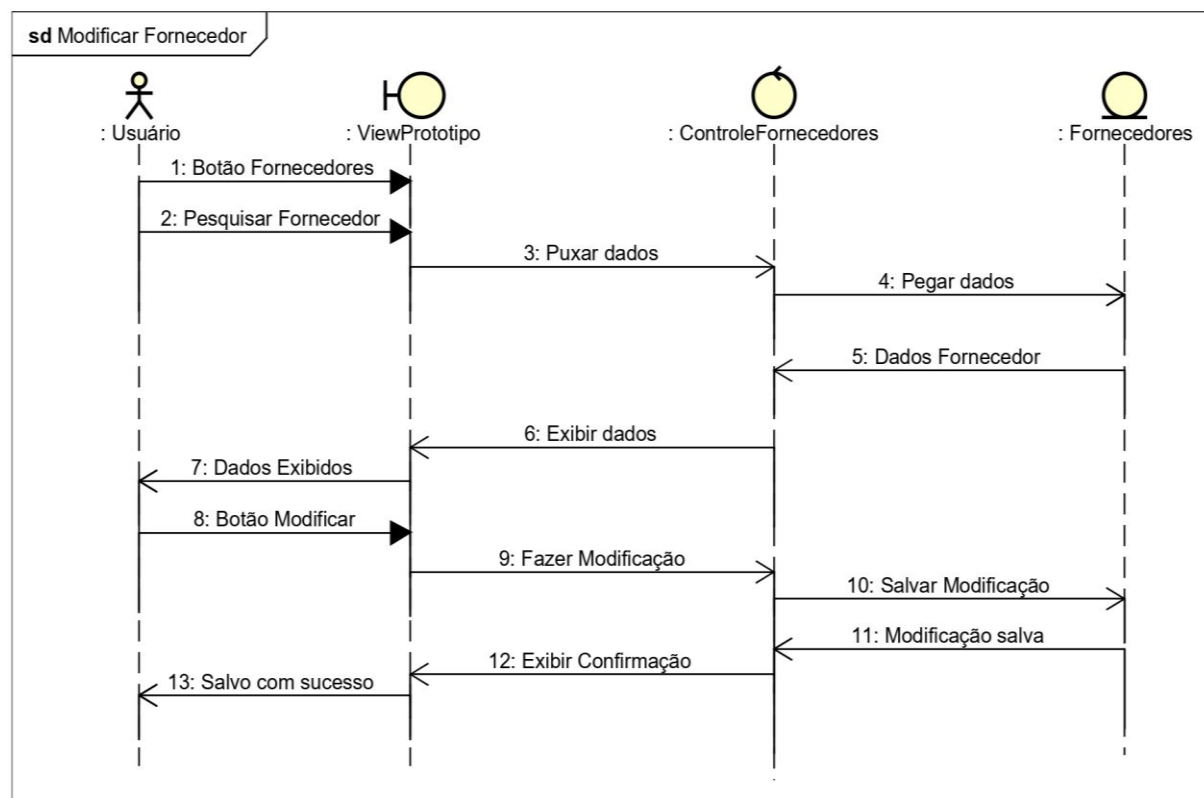
Fonte: O Autor (2024)

Figura 13 – Diagrama de Sequência Listar Fornecedores



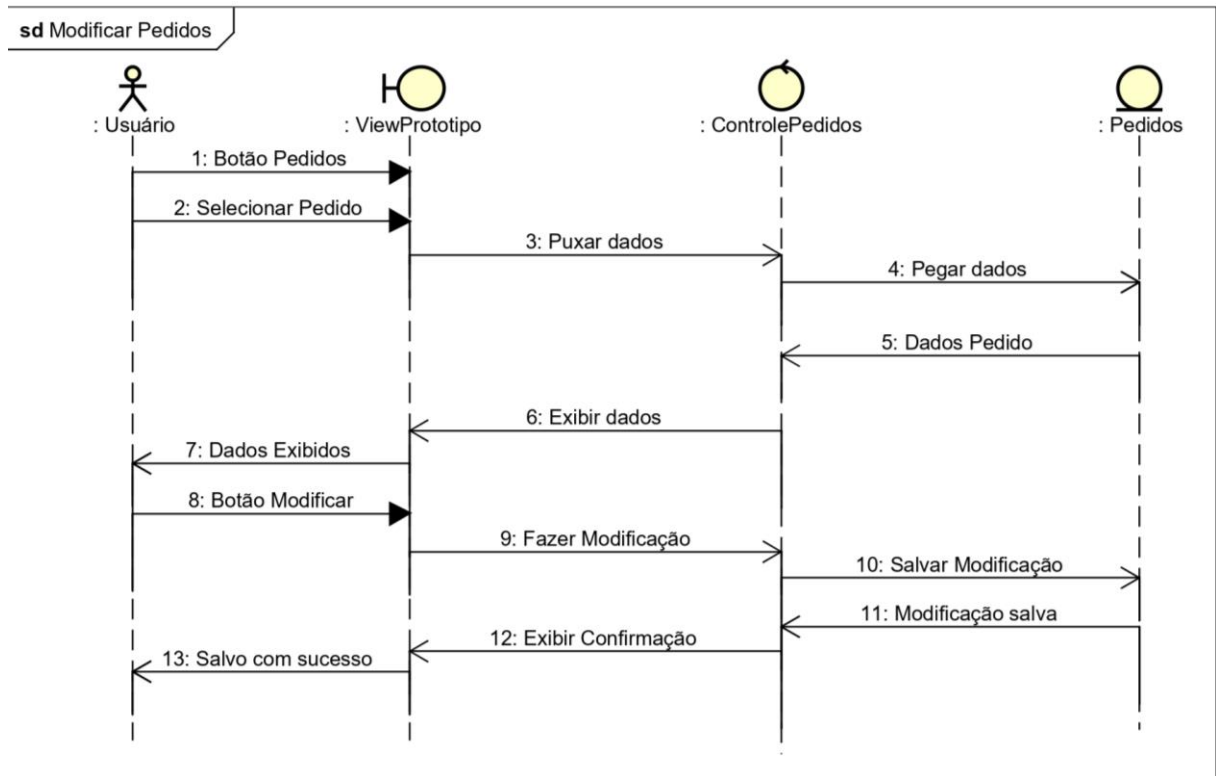
Fonte: O Autor (2024)

Figura 14 – Diagrama de Sequência Modificar Fornecedor



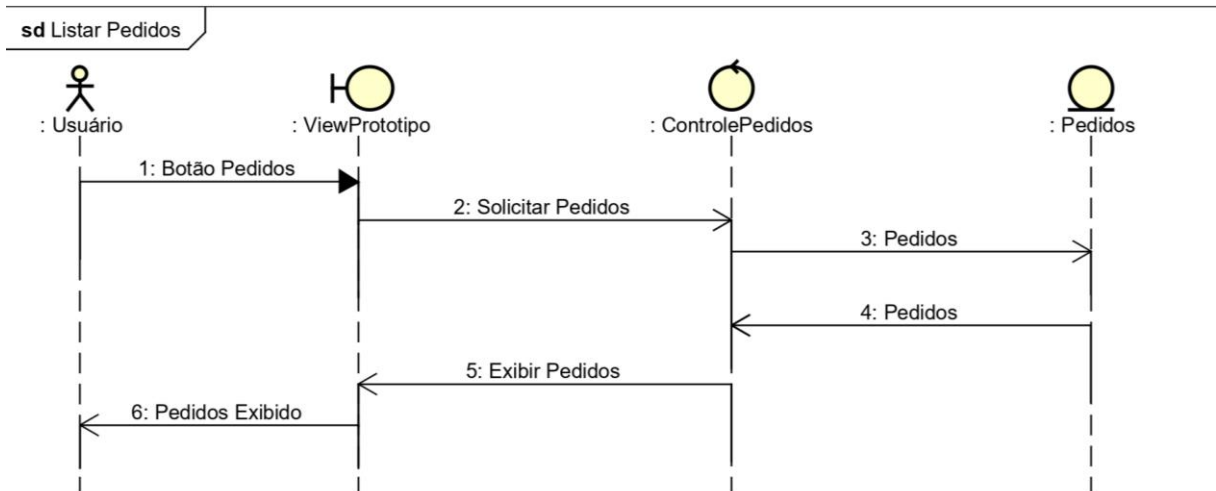
Fonte: O Autor (2024)

Figura 15 – Diagrama de Sequência Modificar Pedido



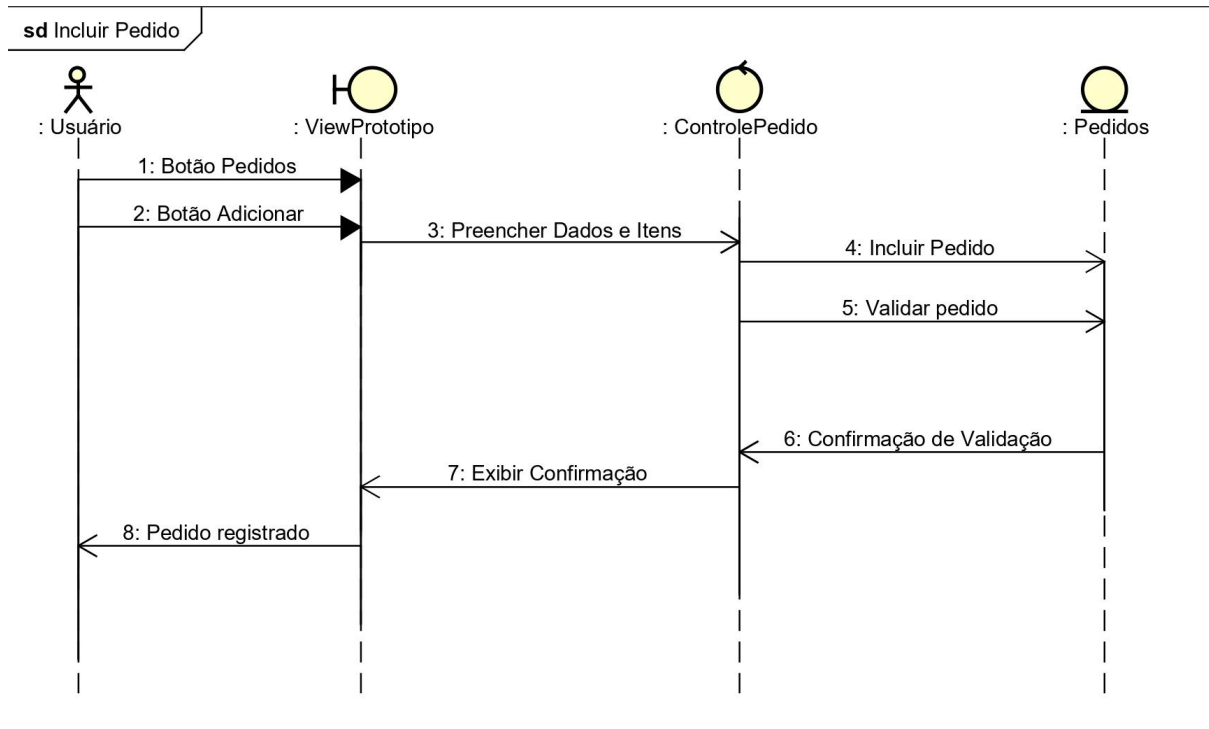
Fonte: O Autor (2024)

Figura 16 – Diagrama de Sequência Modificar Pedido



Fonte: O Autor (2024)

Figura 17 – Diagrama de Sequência Incluir Pedido



Fonte: O Autor (2024)

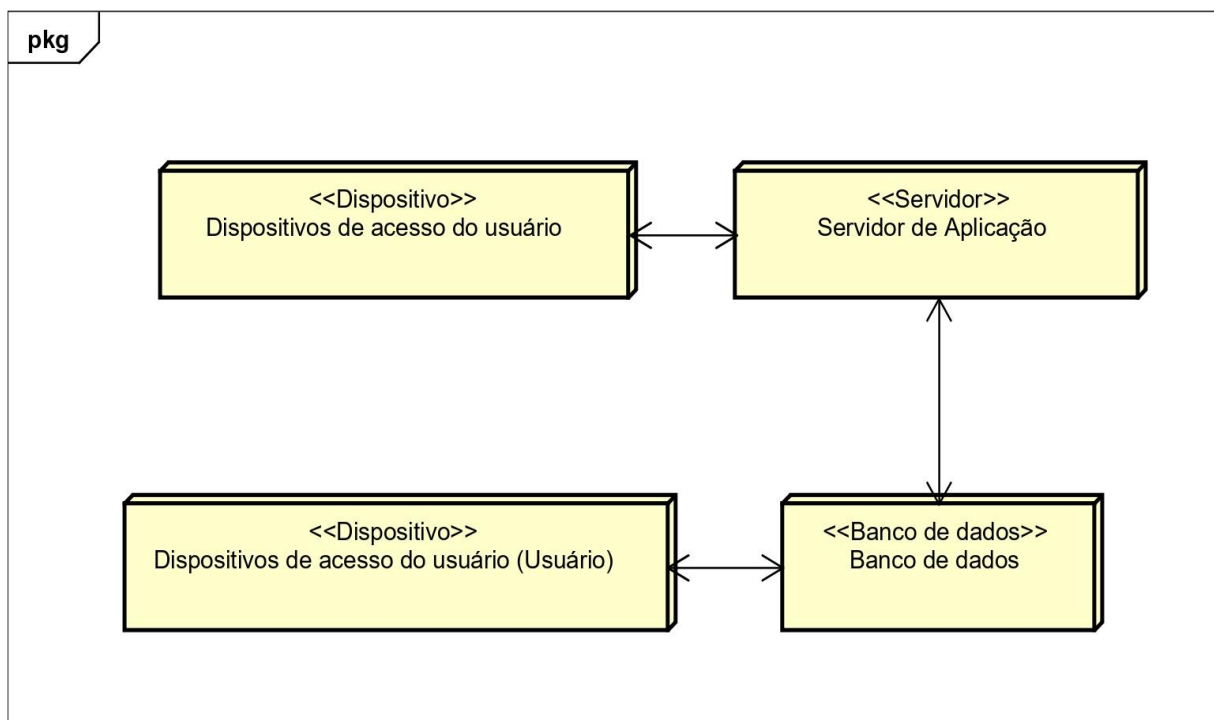
Um diagrama de implementação é uma ferramenta essencial na engenharia de software que ilustra como os componentes do sistema são distribuídos fisicamente nos nodos de processamento. Ele é fundamental para entender como o sistema será implantado, configurado e executado no ambiente real, destacando a estrutura física do sistema, incluindo servidores, dispositivos e outros elementos de hardware, e como os diferentes componentes de software e artefatos estão alocados nesses nodos.

Os nodos no diagrama de implementação representam unidades de processamento ou dispositivos de hardware, como servidores, computadores ou dispositivos móveis. Cada nodo é mostrado como uma caixa e geralmente é etiquetado com o tipo de hardware ou sistema operacional. Dentro desses nodos, são colocados os componentes, que são partes do software que serão executadas nesses dispositivos. Esses componentes podem ser módulos, serviços, aplicações completas ou bibliotecas. Além dos componentes, o diagrama também mostra os artefatos, que são os produtos finais do desenvolvimento, como arquivos executáveis, bibliotecas, arquivos de configuração ou pacotes de instalação.

Artefatos são ligados aos componentes e mostram como o software é realmente armazenado e implantado.

Os diagramas de implementação são fundamentais para o planejamento de implantação, ajudando a garantir que todos os componentes necessários estejam devidamente instalados e configurados nos nodos apropriados. Eles também fornecem uma documentação clara e detalhada da arquitetura física do sistema, o que facilita a manutenção e a atualização. Além disso, servem como uma ferramenta de comunicação eficaz entre as equipes de desenvolvimento, operações e suporte, assegurando que todos tenham uma compreensão comum de como o sistema será implementado e operado no ambiente real.

Figura 18 – Diagrama de Implementação



Fonte: O Autor (2024)

4.3 Banco de Dados

Em qualquer sistema moderno, a gestão e armazenamento de dados são aspectos cruciais que exigem atenção especial durante o desenvolvimento. Em particular, a conformidade com as leis de proteção de dados, como a LGPD (Lei Geral de Proteção de Dados), é uma prioridade inegociável para garantir a segurança e privacidade das informações pessoais dos usuários.

A criação de um banco de dados seguro e eficiente emerge como um dos desafios fundamentais ao implementar funcionalidades de armazenamento de dados em um software. Nesse contexto, a modelagem do banco de dados desempenha um papel crucial. Utilizamos o diagrama de entidade-relacionamento (ER) como uma ferramenta central nesse processo.

O diagrama de entidade-relacionamento é uma abordagem consagrada e amplamente adotada na modelagem de bancos de dados. Ele oferece uma representação visual clara das entidades de dados, seus atributos e os relacionamentos entre elas. Ao utilizar esse diagrama, podemos definir a estrutura do banco de dados de forma precisa e abrangente, considerando as relações entre os diferentes elementos de dados.

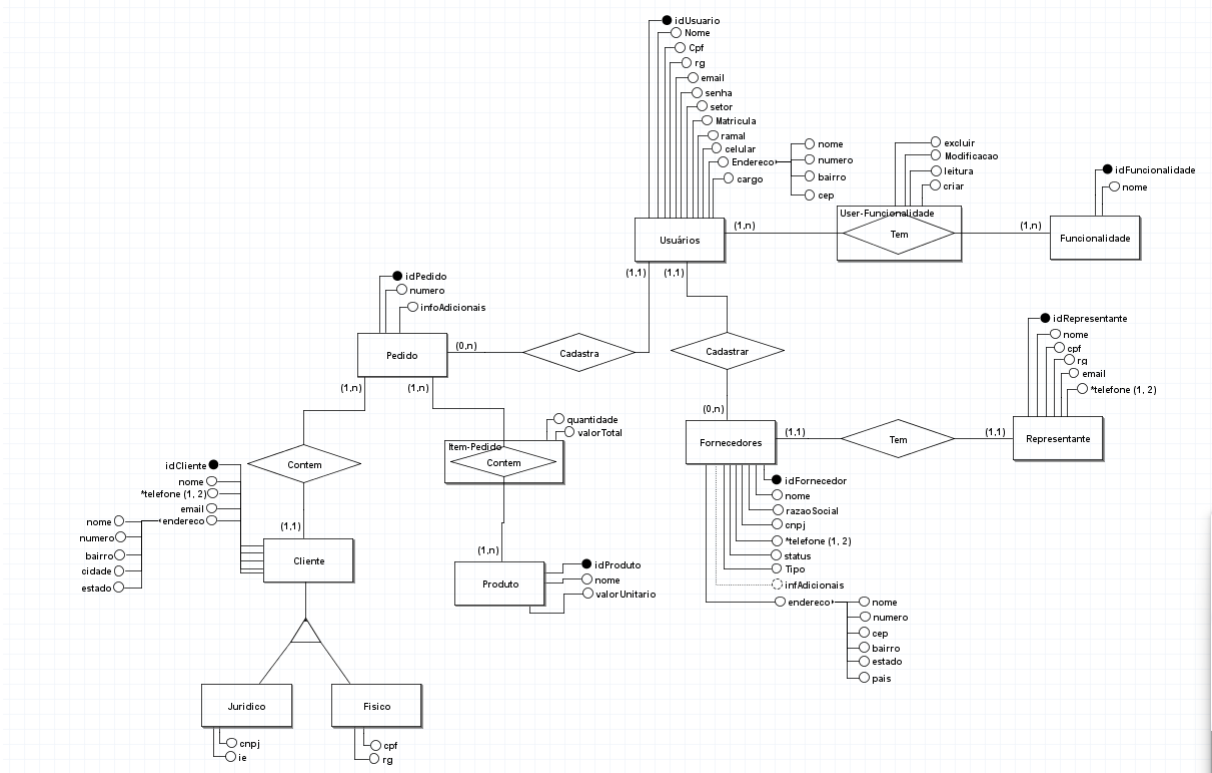
Através do diagrama de entidade-relacionamento, podemos identificar entidades importantes do sistema, como usuários, produtos, pedidos e transações, e mapear suas interações e conexões. Isso nos permite projetar um esquema de banco de dados que seja eficiente, seguro e escalável, atendendo às necessidades específicas do sistema e garantindo a conformidade com as regulamentações de proteção de dados, como a LGPD.

Em suma, a modelagem do banco de dados usando o diagrama de entidade-relacionamento é uma etapa essencial no desenvolvimento de sistemas de software robustos e seguros. Ao garantir uma representação clara e precisa da estrutura de dados, podemos construir sistemas que protegem os dados dos usuários e oferecem uma experiência confiável e segura.

Ao modelar o banco de dados da fazenda, foi essencial compreender os diversos processos que compõem suas operações, a fim de determinar quais dados seriam armazenados no sistema. Esse processo de modelagem de dados envolveu a análise e definição de três tipos de modelos: Conceitual, Lógico e Físico.

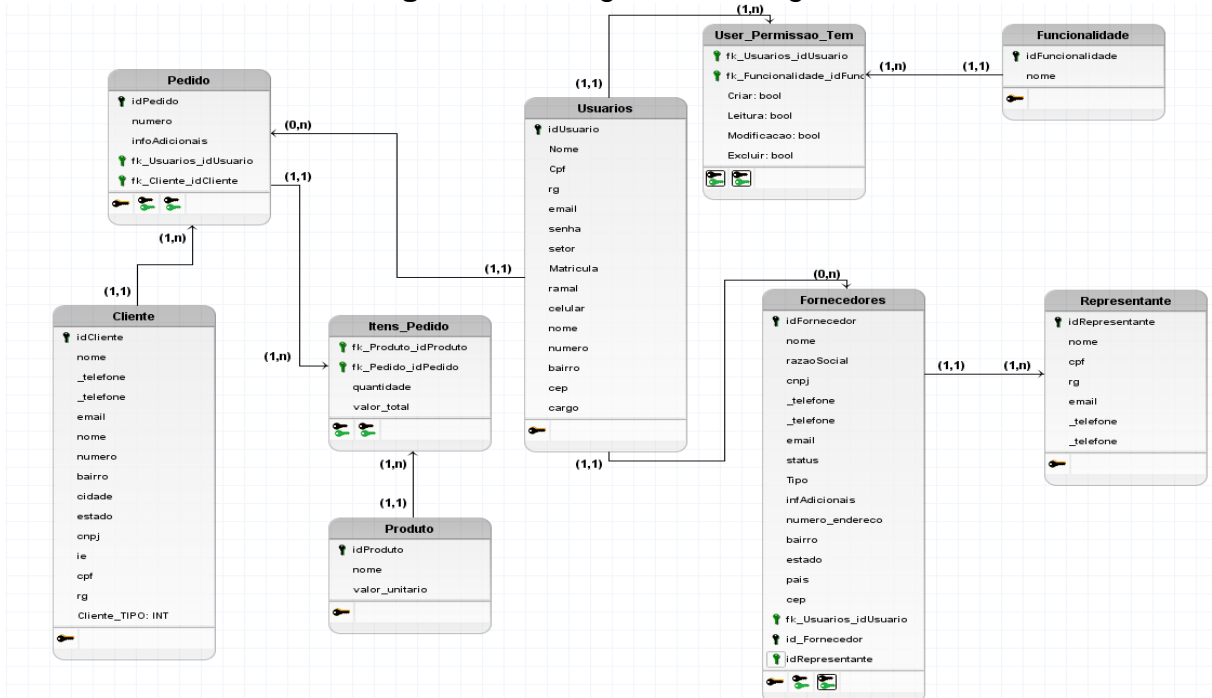
Nesta etapa do estudo, concentramo-nos principalmente nos modelos Conceitual e Lógico, utilizando o software BR Model.

Figura 19 – Diagrama ER Conceitual



Fonte: O Autor (2024)

Figura 20 – Diagrama ER Lógico



Fonte: O Autor (2024)

Preparamos um dicionário de dados para aprimorar a compreensão visual do sistema. Esse recurso é fundamental para facilitar a compreensão por parte daqueles que não estiveram diretamente envolvidos na elaboração do banco de dados, além de oferecer suporte essencial durante processos de manutenção e atualização. A seguir, apresentamos a tabela detalhada do dicionário de dados:

Figura 21 – Dicionário de Dados

Dicionário de Dados			
Tabela	Relacionamento	Nome do Relacionamento	Descrição
Usuários	User-Funcionalidade	Tem	Um usuário dentro de uma funcionalidade pode ter permissões de CRUD
	Fornecedores	Cadastra, Lista, Desativa/Ativa e Modifica	Dentro da funcionalidade fornecedor o usuário pode realizar certas ações de acordo com a sua permissão
	Pedido	Cadastra, Lista, Exclui e Modifica	Dentro da funcionalidade pedido o usuário pode realizar certas ações de acordo com a sua permissão
User-Funcionalidade	Usuários	Tem	Uma ação dentro de uma funcionalidade pode ter vários usuários
	Funcionalidade	Tem	Um usuário pode ter várias ações dentro da funcionalidade
Funcionalidade	User-Funcionalidade	Tem	Uma funcionalidade tem várias ações
Fornecedores	Usuários	Cadastrado	Um fornecedor é cadastrado por um usuário com permissão
	Representante	Tem	Um fornecedor tem 1 representante
Representante	Fornecedores	Representa	Um representante representa um fornecedor
Pedidos	Cliente	Feito	Um pedido é feito por um cliente
Itens-Pedido	Pedidos	Contém	Um produto está em vários pedidos
	Produto	Contém	No pedidos tem vários produtos
Cliente	Pedidos	Faz	O cliente faz pedidos
Produto	Itens-Pedido	Escolhido	Um produto está presente em vários pedidos

Fonte: O Autor (2024)

No entanto, apesar da importância do dicionário de dados, é observado que muitos desenvolvedores subutilizam ou até mesmo negligenciam sua utilização no ciclo de desenvolvimento de software. Este fenômeno pode ser atribuído a uma série de fatores, que merecem nossa análise cuidadosa.

É importante reconhecer que o uso do dicionário de dados pode ser percebido como uma tarefa tediosa e burocrática pelos desenvolvedores, que muitas vezes estão imersos em prazos apertados e demandas urgentes de entrega de software. Como observado por McConnell (2004), em seu livro "Code Complete: A Practical Handbook of Software Construction", os desenvolvedores tendem a priorizar atividades que têm um impacto imediato e tangível no progresso do projeto, relegando tarefas administrativas, como a manutenção do dicionário de dados, para segundo plano.

Após a elaboração dos diagramas e do dicionário de dados, conseguimos obter uma perspectiva mais clara do banco de dados. Com isso, foi possível gerar o script de criação do banco, incluindo as chaves primárias e estrangeiras. A seguir, apresentamos o script completo de criação:

Figura 22 – Script de Criação do Banco de Dados. Parte de Usuários

```
CREATE TABLE Funcionalidade (
    idFuncionalidade int IDENTITY(1,1) PRIMARY KEY,
    nome VARCHAR(100));

CREATE TABLE Usuarios (
    idUsuario int IDENTITY(1,1) PRIMARY KEY NOT NULL,
    nome VARCHAR(255),
    cpf CHAR(14),
    rg CHAR(9),
    email VARCHAR(100),
    senha VARCHAR(15),
    setor VARCHAR (20),
    matricula INT,
    ramal INT,
    celular VARCHAR(20),
    numeroEndereco INT,
    bairro VARCHAR(50),
    cep VARCHAR(10),
    cargo VARCHAR(100)
);

CREATE TABLE User_Permissao_Tem(
    fk_Usuarios_idUsuario int,
    fk_Funcionalidade_idFuncionalidade int,
    criar bit,
    leitura bit,
    modificacao bit,
    excluir bit,
    FOREIGN KEY (fk_Usuarios_idUsuario) REFERENCES Usuarios (idUsuario),
    FOREIGN KEY(fk_Funcionalidade_idFuncionalidade) REFERENCES Funcionalidade (idFuncionalidade)
);
```

Fonte: O Autor (2024)

Figura 23 – Script de Criação do Banco de Dados. Parte de Fornecedores

```
CREATE TABLE Representante(  
    idRepresentante int IDENTITY(1,1) PRIMARY KEY NOT NULL,  
    nome VARCHAR(255),  
    cpf CHAR(14),  
    rg CHAR (9),  
    email VARCHAR(100),  
    telefone1 VARCHAR(20),  
    telefone2 VARCHAR(20)  
);  
  
CREATE TABLE Fornecedores (  
    idFornecedor int IDENTITY(1,1) PRIMARY KEY NOT NULL,  
    nome VARCHAR(255),  
    razaoSocial VARCHAR(255),  
    cnpj CHAR(14),  
    telefone1 VARCHAR(20),  
    telefone2 VARCHAR(20),  
    email VARCHAR(100),  
    _status bit,  
    tipo VARCHAR (30),  
    infAdicionais VARCHAR(255),  
    numeroEndereco INT,  
    bairro VARCHAR(50),  
    estado CHAR(2),  
    pais VARCHAR(30),  
    cep VARCHAR(10),  
    fk_Usuarios_idUsuario int,  
    fk_Representante_idRepresentante int,  
    FOREIGN KEY (fk_Usuarios_idUsuario) REFERENCES Usuarios (idUsuario),  
    FOREIGN KEY (fk_Representante_idRepresentante) REFERENCES Representante (idRepresentante)  
);
```

Fonte: O Autor (2024)

Figura 24 – Script de Criação do Banco de Dados. Parte de Pedidos

```
CREATE TABLE Cliente (  
    idCliente int IDENTITY(1,1) PRIMARY KEY NOT NULL,  
    nome VARCHAR(255),  
    telefone1 VARCHAR(20),  
    telefone2 VARCHAR(20),  
    cpf CHAR(14),  
    rg CHAR(9),  
    cnpj CHAR(14),  
    ie char(9),  
    email VARCHAR(100),  
    numeroEndereco INT,  
    nomeEndereco VARCHAR (150),  
    bairro VARCHAR(50),  
    cidade VARCHAR (30),  
    estado CHAR(2),  
    cep VARCHAR(10),  
    clienteTipo VARCHAR(30),  
);  
  
CREATE TABLE Pedido (  
    idPedido INT IDENTITY(1,1) PRIMARY KEY NOT NULL,  
    numero VARCHAR(30),  
    infoAdicionais varchar(255),  
    fk_Usuarios_idUsuario INT,  
    fk_Cliente_idCliente INT,  
    FOREIGN KEY (fk_Usuarios_idUsuario) REFERENCES Usuarios (idUsuario),  
    FOREIGN KEY (fk_Cliente_idCliente) REFERENCES Cliente (idCliente),  
);
```

Fonte: O Autor (2024)

Figura 25 – Script de Criação do Banco de Dados. Continuação Parte Pedidos

```
CREATE TABLE Produto (  
    idProduto INT IDENTITY(1,1) PRIMARY KEY NOT NULL,  
    nome VARCHAR(255),  
    valorUnitario DECIMAL(5,2)  
);  
  
CREATE TABLE Itens_Pedido (  
    quantidade INT,  
    valor_total DECIMAL(9,2),  
    fk_Produto_idProduto INT,  
    fk_Pedido_idPedido INT,  
    FOREIGN KEY (fk_Produto_idProduto) REFERENCES Produto (idProduto),  
    FOREIGN KEY (fk_Pedido_idPedido) REFERENCES Pedido (idPedido),  
);
```

Fonte: O Autor (2024)

4.4 Protótipo

Para garantir uma avaliação visual precisa e eficaz, utilizamos a ferramenta Figma para desenvolver um protótipo de alta fidelidade do sistema. Esse protótipo foi projetado com o propósito de apresentar uma representação visual completa para o cliente e os stakeholders envolvidos no projeto.

Um dos principais focos do nosso protótipo foi a usabilidade do sistema, levando em consideração diferentes perfis de usuários, incluindo colaboradores idosos ou aqueles com pouca familiaridade com tecnologia. Para isso, dedicamos especial atenção à simplicidade e clareza da interface, garantindo que fosse intuitiva e fácil de usar, mesmo para aqueles menos familiarizados com tecnologia. Ao mesmo tempo, asseguramos que o sistema mantivesse a eficiência e a funcionalidade necessárias para atender às demandas operacionais da fazenda urbana.

Ao zelar pela usabilidade do sistema em nosso protótipo, buscamos proporcionar uma experiência positiva e acessível para todos os usuários, independentemente de seu nível de experiência tecnológica. Isso não apenas facilita a adoção do sistema pelos colaboradores, mas também contribui para a eficiência e produtividade no ambiente de trabalho.

Em resumo, o uso do Figma para desenvolver um protótipo de alta fidelidade nos permitiu apresentar visualmente as funcionalidades e a interface do sistema, enquanto priorizamos a usabilidade e a acessibilidade para todos os usuários. Essa abordagem garante uma avaliação mais precisa e facilita a comunicação com o cliente e os stakeholders, promovendo uma colaboração eficaz no desenvolvimento do projeto.

O projeto tem prevenções contra falhas, isso evita que usuários desatentos acabem dando cliques acidentais e perdendo informações ou o foco no trabalho. Essa função evita muitos problemas de perda de dados importantes.

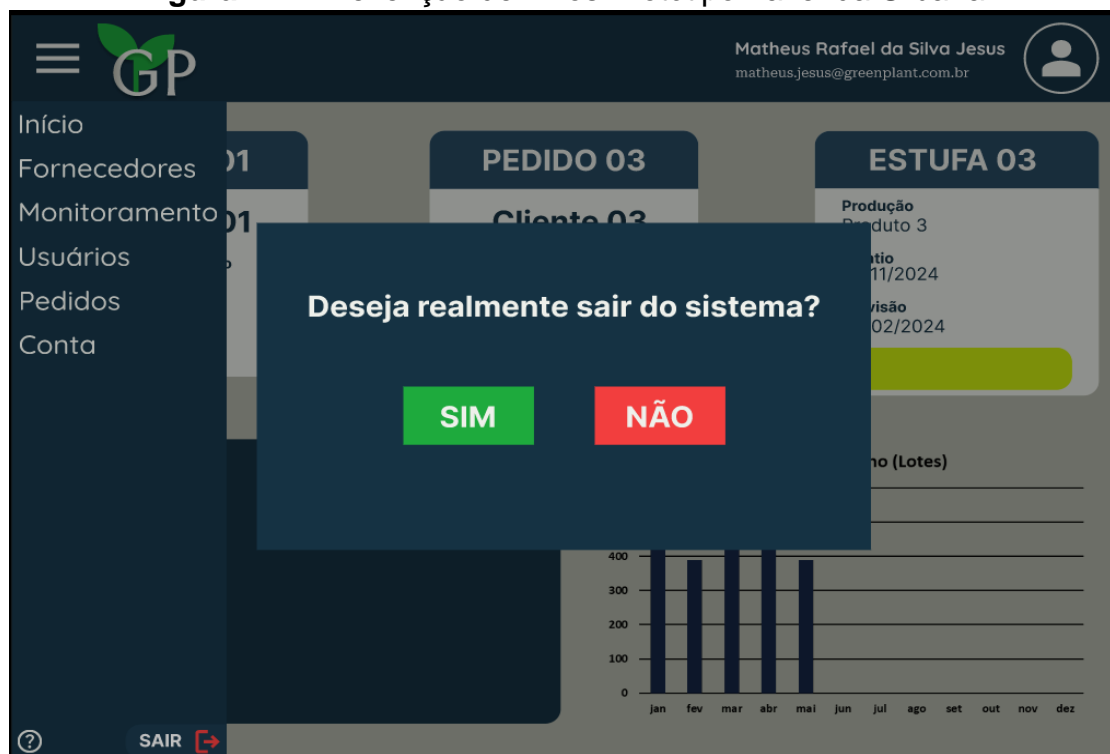
Logo abaixo temos a apresentação da página inicial após o login e uma representação de prevenção de erro. [Figma](#).

Figura 26 – Página Inicial Protótipo Fazenda Urbana



Fonte: O Autor (2024)

Figura 27 – Prevenção de Erros Protótipo Fazenda Urbana



Fonte: O Autor (2024)

4.5 Teste

Para este estágio do projeto, planejamos uma série de testes para o próximo passo, visando aprimorar ainda mais a usabilidade do sistema. Entre esses testes, incluímos o teste A/B, uma abordagem que envolve a comparação de duas versões diferentes do sistema, onde um grupo de usuários experimenta funcionalidades novas, enquanto outro grupo utiliza o sistema típico.

O teste A/B é uma estratégia eficaz para avaliar e comparar o desempenho de diferentes versões do sistema, permitindo-nos identificar quais alterações ou adições resultam em uma melhor experiência do usuário. Durante o teste, monitoramos métricas relevantes, como taxa de conclusão de tarefas, tempo gasto em determinadas atividades e feedback dos usuários.

O outro teste planejado é o teste de unidade. O principal objetivo do teste de unidade é identificar e corrigir possíveis falhas ou bugs no código o mais cedo possível, durante o processo de desenvolvimento. Isso ajuda a prevenir que problemas se propaguem para outras partes do sistema, tornando a depuração mais fácil e eficiente.

Durante o teste de unidade, cada unidade de código é testada em diferentes cenários, incluindo casos de uso normais, borda e excepcionais. Isso permite verificar se a unidade produz os resultados esperados sob diferentes condições de entrada. O desenvolvedor escreve casos de teste específicos para cada unidade de código, simulando diferentes situações e verificando se o comportamento está de acordo com o esperado.

Testes de softwares são uma parte importante da engenharia de software que visa garantir o bom funcionamento e o resultado esperado pelos desenvolvedores. Existem alguns frameworks que realizam testes, entre os mais famosos estão o PyTest, NUnit, Jest e RSpec.

Identificamos a partir do diagrama de classe as principais classes do sistema, com isso planejamos alguns testes de unidade neles, e visando melhorar a usabilidade do sistema, planejamos alguns testes A/B, como mostrar a tabela de testes abaixo:

Tabela 4 – Tabela de Testes

Metódo/Interface	Teste	Prioridade	Resultado Esperado	Resultado Obtido
incluirUsuario	Teste Unidade	ALTA	Incluído com sucesso	
excluirUsuario	Teste Unidade	ALTA	Excluído com sucesso	
incluirFornecedor	Teste Unidade	ALTA	Incluído com sucesso	
ativarDesativarFornecedor	Teste Unidade	ALTA	Salvo com sucesso	
cadastrarPedidos	Teste Unidade	ALTA	Cadastrado com sucesso	
Mudança na opção de sair	A/B	BAIXA	Análise de aceitação	

Fonte: O Autor (2024)

4.6 Prova de Conceito com CRUD

PoC (Proof of Concept, ou Prova de Conceito) é uma aplicação simples desenvolvida para demonstrar a viabilidade de uma ideia ou conceito. Quando se fala de um PoC que faz CRUD (Create, Read, Update, Delete), estamos nos referindo a um sistema básico que permite criar, ler, atualizar e deletar dados. Vamos imaginar um sistema de gerenciamento de usuários para entender melhor.

Na parte de Create (Criar), o sistema deve permitir a criação de novos registros. Por exemplo, um formulário onde o usuário insere seu nome e email. Quando o formulário é enviado, esses dados são salvos em um banco de dados. Esse processo é essencial para adicionar novas informações ao sistema.

A parte de Read (Ler) envolve a leitura e visualização dos dados armazenados. O sistema pode ter uma página que lista todos os usuários cadastrados, exibindo seus nomes e emails. Isso permite que os usuários visualizem e confirmem que seus dados estão corretamente armazenados.

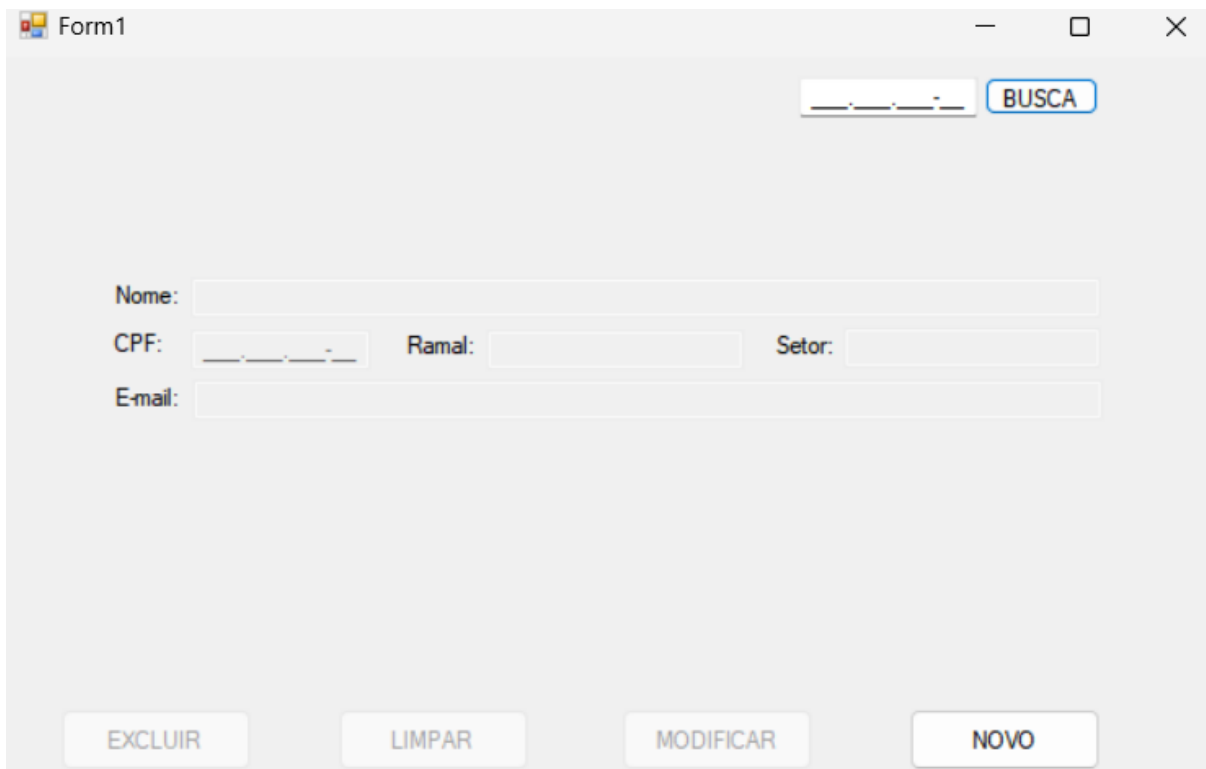
Na etapa de Update (Atualizar), o sistema permite que os dados existentes sejam modificados. Por exemplo, um usuário pode querer atualizar seu email. O sistema deve fornecer uma forma de editar os dados e salvar as mudanças no banco de dados. Isso é crucial para manter as informações atualizadas e corretas.

Por fim, a parte de Delete (Deletar) permite a remoção de registros do banco de dados. Se um usuário quiser excluir sua conta, o sistema deve oferecer uma opção para deletar os dados associados a esse usuário. Isso garante que os dados desnecessários ou indesejados possam ser removidos do sistema.

Um PoC com essas funcionalidades de CRUD demonstra de forma clara e prática como um sistema pode manipular dados básicos, sendo um passo importante para validar a viabilidade de um projeto maior.

Criamos um PoC (Proof of Concept) que implementa funcionalidades de CRUD utilizando o Forms do C#. Com esta aplicação, é possível cadastrar usuários no banco de dados, excluir usuários, modificar dados dos usuários e buscar por usuários específicos. [PoC CRUD](#).

Figura 28 – Form 1 PoC CRUD



Form1

_____-_____-_____- BUSCA

Nome: _____

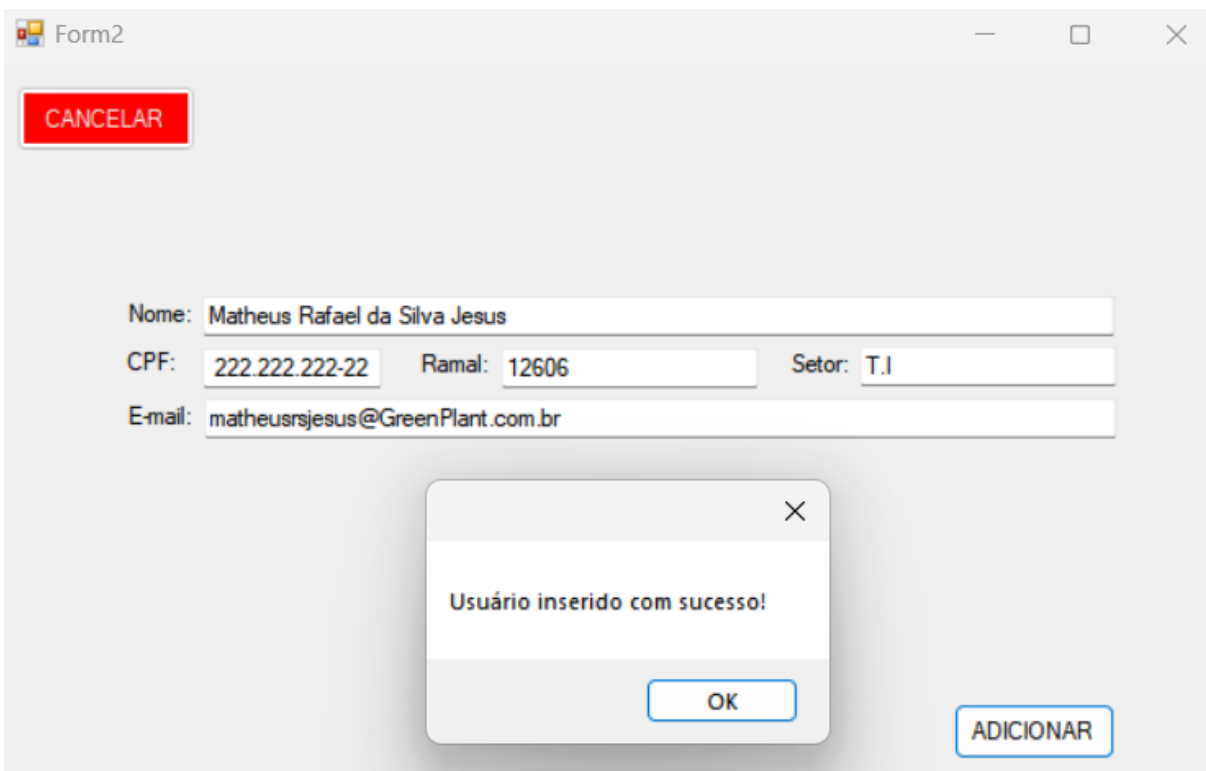
CPF: _____ Ramal: _____ Setor: _____

E-mail: _____

EXCLUIR LIMPAR MODIFICAR NOVO

Fonte: O Autor (2024)

Figura 29 – Form 2 Usuário Incluído



Form2

CANCELAR

Nome: Matheus Rafael da Silva Jesus

CPF: 222.222.222-22 Ramal: 12606 Setor: T.I

E-mail: matheusrjesus@GreenPlant.com.br

Usuário inserido com sucesso!

OK

ADICIONAR

Fonte: O Autor (2024)

Figura 30 – Form 1 Buscar Usuário

Form1

222.222.222-22 BUSCA

Nome: Matheus Rafael da Silva Jesus

CPF: 222.222.222-22 Ramal: 12606 Setor: T.I

E-mail: matheusrjesus@GreenPlant.com.br

EXCLUIR LIMPAR MODIFICAR NOVO

Fonte: O Autor (2024)

Figura 31 – Usuário Incluído no Banco de Dados

	idUsuario	nome	cpf	rg	email	senha	setor	matricula	ramal	celular	numeroEndereco	bairro	cep	cargo
1	6	Matheus Rafael da Silva Jesus	707.070.707-07	NULL	PimUnip01@gmail.com	NULL	TI	NULL	12606	NULL	NULL	NULL	NULL	NULL
2	7	Matheus Rafael da Silva Jesus	222.222.222-22	NULL	matheusrjesus@GreenPlant.com.br	NULL	T.I	NULL	12606	NULL	NULL	NULL	NULL	NULL

Fonte: O Autor (2024)

Figura 32 – Modificação de Dados do Usuário

Form3

CANCELAR

707.070.707-07 **BUSCA**

Nome: Victor Hugo

CPF: 111.111.111-11 Ramal: 12606 Setor: T.I

E-mail: PimUnip01@gmail.com

Atualizado com sucesso!

OK

SALVAR

Fonte: O Autor (2024)

Figura 33 – Usuário Atualizado no Banco de Dados

Results		Messages												
	idUsuario	nome	cpf	rg	email	senha	setor	matricula	ramal	celular	numeroEndereco	bairro	cep	cargo
1	6	Victor Hugo	111.111.111-11	NULL	PimUnip01@gmail.com	NULL	T.I	NULL	12606	NULL	NULL	NULL	NULL	NULL
2	7	Matheus Rafael da Silva Jesus	222.222.222-22	NULL	matheusrjesus@GreenPlant.com.br	NULL	T.I	NULL	12606	NULL	NULL	NULL	NULL	NULL

Fonte: O Autor (2024)

Figura 34 – Mensagem de Confirmação

The screenshot shows a web form titled 'Form1' with a search bar at the top right containing the text '222.222.222-22' and a 'BUSCA' button. Below the search bar are input fields for user information: 'Nome: Matheus Rafael da Silva Jesus', 'CPF: 222.222.222-22', 'Ramal: 12606', 'Setor: T.I', and 'E-mail: matheusrjesus@GreenPlant.com.br'. At the bottom of the form are buttons for 'EXCLUIR', 'NOVO', and an unlabeled button. A modal dialog box titled 'Cuidado' is centered on the screen, displaying a question mark icon and the text 'Deseja realmente excluir esse usuário ?'. The dialog has two buttons: 'Sim' and 'Não'.

Fonte: O Autor (2024)

Figura 35 – Usuário Excluído

This screenshot shows the same 'Form1' as in Figure 34, but with a different modal dialog. The dialog is titled with a close button (X) and contains the text 'Usuário deletado com sucesso!'. It has a single 'OK' button. The background form remains the same, with the search bar, user details, and action buttons ('EXCLUIR', 'LIMPAR', 'MODIFICAR', 'NOVO') visible.

Fonte: O Autor (2024)

4.7 Recursos Humanos

Os recursos humanos necessários para operar uma fazenda urbana incluem diferentes papéis, como gerente da fazenda, agricultores urbanos, especialistas em agricultura, engenheiros agrônomos, técnicos agrícolas, especialistas em sistemas de irrigação, especialistas em energia renovável, equipe de vendas e marketing, administradores de escritório e equipe de limpeza e manutenção. Cada um desses papéis desempenha uma função crucial na gestão, cultivo, comercialização e manutenção da fazenda urbana. Juntos, esses recursos humanos trabalham para garantir o sucesso e a sustentabilidade da fazenda, fornecendo alimentos frescos e saudáveis para as comunidades urbanas.

Os relatórios de gestão são fundamentais para a avaliação e aprimoramento contínuo de negócios em diversos setores. Neste estudo, abordaremos três tipos essenciais de relatórios de gestão: análise de evolução dos negócios, análise de mercado e relatório de desempenho dos funcionários (RH). Compreender esses relatórios é crucial para gestores e empresários que buscam otimizar suas operações e alcançar o sucesso sustentável em um cenário empresarial dinâmico e desafiador.

A análise de evolução dos negócios visa monitorar o crescimento da empresa ao longo do tempo, identificando tendências e oportunidades com base em métricas financeiras e operacionais. Para isso, realizaremos monitoramentos semestrais para acompanhar a progressão da empresa.

A análise de mercado examinará o ambiente externo da empresa, incluindo tendências do mercado, comportamento do consumidor e concorrência, visando identificar oportunidades de crescimento. Para isso, serão realizadas pesquisas mensais para acompanhar o desenvolvimento do mercado.

O relatório de desempenho dos funcionários avalia individualmente cada colaborador, destacando seus pontos fortes, áreas de melhoria e contribuições para os resultados da empresa. Estes relatórios serão elaborados pelos líderes e gerentes, garantindo uma avaliação precisa e detalhada do desempenho da equipe.

5. CONSIDERAÇÕES FINAIS

O presente trabalho abordou a relevância crescente das fazendas urbanas e o desenvolvimento de um sistema de gerenciamento eficiente, destacando a importância desse tema no contexto das cidades modernas. O problema inicial identificado foi a necessidade de otimizar a produção e gestão das fazendas urbanas, que enfrentam desafios significativos como a limitação de espaço, a demanda por alimentos frescos e a sustentabilidade ambiental.

Para solucionar esses desafios, propusemos um sistema de gerenciamento que inclui funcionalidades essenciais como gestão de fornecedores e funcionários, monitoramento de colheitas e plantio, gerenciamento de pedidos. Esse sistema visa melhorar a eficiência operacional das fazendas urbanas, facilitando a tomada de decisões e maximizando a produtividade.

Os impactos econômicos das fazendas urbanas foram explorados, mostrando que essas iniciativas não apenas promovem a segurança alimentar e a sustentabilidade, mas também geram empregos e estimulam o empreendedorismo local. As fazendas urbanas estão em um crescimento significativo, similar ao observado no ecossistema de startups, o que demonstra seu potencial econômico e social.

No entanto, reconhecemos que o estudo possui algumas limitações, como a necessidade de adaptação a diferentes contextos urbanos e a superação de barreiras culturais e financeiras para a adoção de novas tecnologias. Para futuras pesquisas, sugerimos a realização de estudos de caso em diversas cidades para validar e ajustar o sistema proposto a diferentes realidades urbanas.

Em conclusão, as fazendas urbanas representam uma estratégia promissora para enfrentar os desafios urbanos de sustentabilidade e segurança alimentar. O desenvolvimento de sistemas de gerenciamento eficientes é crucial para maximizar seu impacto positivo. Espera-se que este trabalho inspire novas iniciativas e pesquisas no campo de Análise e Desenvolvimento de Sistemas, contribuindo para o avanço das práticas agrícolas urbanas e incentivando empresas de locação a receber mais propostas e contribuições nessa área emergente.

6. REFERÊNCIAS

RIES, Eric; The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses. Crown Currency, 2011.

BLANK, Steve; DORF, Bob. The startup owner's manual: The step-by-step guide for building a great company. John Wiley & Sons, 2020.

MASTERS, Blake; THIEL, Peter. Zero to one: notes on start ups, or how to build the future. Random House, 2014.

Nações Unidas. Sobre o nosso trabalho para alcançar os Objetivos de Desenvolvimento Sustentável no Brasil. <<https://brasil.un.org/pt-br/sdgs>>. Acesso em: 16 mai. 2024.

PESSOA, Leandro; Fazendas Urbanas I – Alternativas a Segurança Alimentar. 2016.

PESSOA, Leandro; Fazendas Urbanas II – Modalidade da Agricultura Urbana para Segurança Alimentar. 2018.

FONSECA, Delmo; Descomplicando o Universo ESG: Guia essencial para uma prática sustentável. 2023.

BOOCH, Grady; UML: guia do usuário. Elsevier Brasil, 2006.

MARTINEZ-ALIER, Juan. Economia ecologica. Universidade Federal do Rio Grande do Sul, Programa de Pós-Graduação em Economia, 1996.

FICHA DE CONTROLE DO PIM

Ano: 2024 Período: 2º/3º Coordenador: Prof Roberto Cordeiro Waltz

Tema (Identificação da startup): Fazendas Urbana GreenPlant

Alunos

RA	Nome	E-mail	Curso	Visto do aluno
N3573A1	Matheus Rafael da Silva Jesus		CST em ADS	
G71GEG3	Victor Hugo Rodrigues Barros Antunes		CST em ADS	

--	--	--	--	--

Registros

[illegible]

--	--