



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 |
|---------------------------------------|------------|
| Carlos Henrique Duarte da Silva | G81BHC5 |
| João Gabriel Alves dos Santos | R0167J3 |
| Lucas Navimar de Toledo | R009JD0 |
| Luis Henrique Vieira Barros | R024768 |
| Matheus Guilherme Pacheco Courbassier | R026272 |
| Ruan Pablo Galdino Dias Bento | G86JCF-1 |

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

| | RA |
|---------------------------------------|-----------|
| Carlos Henrique Duarte da Silva | G81BHC5 |
| João Gabriel Alves dos Santos | R0167J3 |
| Lucas Navimar de Toledo | R009JD0 |
| Luis Henrique Vieira Barros | R024768 |
| Matheus Guilherme Pacheco Courbassier | R026272 |
| Ruan Pablo Galdino Dias Bento | G86JCF-1 |

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.

**São José dos Campos – SP
JUNHO/2024**

RESUMO

O projeto foi baseado em um sistema de gestão de uma startup de fazenda urbana movida pela necessidade mundial da crescente demanda por produtos alimentícios, aumentado ainda mais nos últimos anos devido ao aumento da população. A startup se responsabilizou pelo compromisso de produzir alimentos orgânicos e saudáveis. O objetivo inicial foi levantar as regras e processos de negócios, fazer uma estimativa de viabilidade, criar valores profissionais da empresa e desenvolver um sistema de gerenciamento para a fazenda urbana. Foi feito o levantamento de requisitos, divisão entre requisitos funcionais e não funcionais, elaboração dos diagramas UML, criação do DER e criação dos scripts SQL, protótipo de telas, planilha de testes e dicionários de dados. Foi adotado a metodologia Scrum para o desenvolvimento do projeto, tendo dividido a equipe em um scrum master, um product owner e quatro integrantes da equipe de desenvolvimento. O grupo também realizou pesquisas sobre os temas de fazenda urbana, segurança alimentar, banco de alimentos, ESG e ODS, com base na pesquisa realizada foi possível compreender a necessidade de uma produção sustentável e como a tecnologia é muito presente atualmente. Entendeu-se que o projeto desenvolvido alcançou as necessidades estabelecidas pelo grupo levando em consideração os aspectos de usabilidade de sistema e viabilidade econômica.

Palavras-Chave: fazenda urbana, startup, dados, ESG, ODS, requisitos, Scrum.

SUMÁRIO

| | |
|--|----|
| 1. INTRODUÇÃO | 5 |
| 1.1. OBJETIVO GERAL | 6 |
| 1.1.1. Objetivos Específicos | 6 |
| 1.2. Fazendas Urbanas: Tipos e Histórico no Mundo..... | 7 |
| 1.2.1. Tipos de fazendas urbanas:..... | 7 |
| 1.2.2. Histórico das fazendas urbanas:..... | 8 |
| 1.3. Segurança Alimentar e Banco de Alimentos | 8 |
| 1.4. ESG (Ambiental, Social e Governança) e ODS (Objetivos de Desenvolvimento Sustentável)..... | 8 |
| 1.4.1 Sobre o ESG | 9 |
| 1.4.2. Sobre o ODS | 9 |
| 1.5. 30ª Conferência da ONU sobre Mudanças Climáticas | 9 |
| 1.6. Gestão de Pessoas | 10 |
| 1.7. Economia e Viabilidade..... | 11 |
| 1.8. Regras de Negócio..... | 11 |
| 2. CARACTERIZAÇÃO DA EMPRESA | 12 |
| 3. DESENVOLVIMENTO DO PROJETO..... | 13 |
| 3.1. Metodologia Utilizada | 13 |
| 3.2. Levantamento de Requisitos..... | 14 |
| 3.3. Planejamento de Sprints | 16 |

| | |
|---|----|
| 3.4. Diagramas UML | 17 |
| 3.4.1. Diagramas de Caso de Uso | 17 |
| 3.4.2. Diagrama de Classe | 21 |
| 3.4.3. Diagrama de Sequência | 23 |
| 3.4.4. Diagrama de Implantação | 26 |
| Figura 10 – Diagrama de Implantação | 26 |
| 3.5. Prototipagem de Telas | 26 |
| 3.6. CRUD (Create Read Update Delete) | 31 |
| 3.7. Banco de Dados | 36 |
| 3.7.1. Estrutura em SQL | 38 |
| 3.7.2. Inserção de Valores | 40 |
| 3.7.3. Seleção de Tabelas | 41 |
| 3.7.4. Dicionário de Dados | 42 |
| 3.8. Planilha de Testes | 44 |
| 4. CONSIDERAÇÕES FINAIS | 45 |

1. INTRODUÇÃO

Com o crescimento populacional e a urbanização acelerada, são colocados grandes desafios à segurança alimentar, que exigem soluções inovadoras para garantir a produção alimentar saudável e sustentável. Neste contexto, a agricultura urbana é uma alternativa viável para satisfazer a procura de alimentos frescos e reduzir o impacto ambiental do transporte e armazenamento de produtos agrícolas. As startups focadas na segurança alimentar desempenham um papel vital neste contexto, impulsionando a adoção de tecnologias avançadas e práticas sustentáveis.

De acordo com o relatório da ONU, A população mundial deve crescer em 2 bilhões de pessoas nos próximos 30 anos, passando dos atuais 7,7 bilhões de indivíduos para 9,7 bilhões em 2050. Isso traz uma grande responsabilidade para suprir a demanda alimentar da população, com qualidade e segurança. Porém, com a redução das áreas agricultáveis e a urgência em revertermos velhas práticas de exploração de recursos, o desafio está em produzirmos alimentos em grande escala, mas de forma sustentável.

Segundo a empresa privada BRK (2023), A resposta para esse problema está no uso da tecnologia e da automação de processos extraordinariamente bem empregados em fazendas urbanas. A principal ideia é utilizar a tecnologia em diversos países, como o Brasil, para diminuir os gastos com transporta e uso de agrotóxicos, além de facilitar o manejo dos funcionários e logística de uma Fazenda Urbana

Com esse objetivo em mente, a fazenda urbana Terras Mil é uma nova iniciativa que combina tecnologia e inovação para otimizar os processos de produção e comércio. Para alcançar eficiência e sustentabilidade, será desenvolvido um sistema de controle abrangente para gerenciar fornecedores, produção e vendas de produtos. O sistema deve atender às necessidades específicas do cliente, que seja possível administrar a fazenda urbana de modo eficiente.

Além disso, a fazenda urbana Terras Mil se compromete a implementar práticas de agricultura regenerativa e técnicas de cultivo vertical para maximizar o uso do espaço urbano disponível, aumentando a produção sem a necessidade de expandir a área física da fazenda. Utilizando o próprio sistema de gerenciamento de produção para monitorar a saúde das plantas e condições do solo em tempo real, a fazenda urbana Terras Mil garante que

cada planta receba os nutrientes e cuidados adequados, minimizando desperdícios e otimizando recursos. Com a gestão estratégica de tarefas entre os funcionários e controle de clientes e fornecedores parceiros, a fazenda urbana Terras Mil se manterá na vanguarda das inovações e produções agrícolas, contribuindo ativamente para a construção de um futuro alimentar mais resiliente e sustentável.

1.1. OBJETIVO GERAL

O objetivo geral deste projeto foi definido como a criação de um sistema de gerenciamento de uma fazenda urbana, sistema esse que tem como finalidade auxiliar o usuário a visualizar e entender os processos que ocorrem na fazenda urbana.

1.1.1. Objetivos Específicos

Com o propósito de atingir o objetivo geral proposto, serão considerados os seguintes objetivos específicos:

- Identificar as Regras de Negócio da fazenda urbana;
- Criar uma lista de requisitos funcionais e não funcionais;
- Utilizar UML para desenvolver o Diagrama de Caso de Uso para cada ferramenta do sistema;
- Desenvolver o Diagrama de Sequência baseando-se no Caso de Uso;
- Criar um protótipo de telas para o sistema;
- Desenvolver o Diagrama de Classe, Modelo Entidade Relacionamento e Diagrama Entidade Relacionamento;
- Criar o Script do banco de dados utilizando SQL, junto com o Dicionário de Dados;
- Desenvolver o CRUD (Create, Read, Update, Delete);

- Elaborar uma planilha de testes;
- Criar o Diagrama de Implantação.

1.2. Fazendas Urbanas: Tipos e Histórico no Mundo

De acordo com o site Blueseeds (2024), fazendas urbanas são espaços de produção agrícola localizados em áreas urbanas ou periurbanas, onde são cultivados alimentos, ervas, flores ou outros produtos agrícolas. Essas fazendas variam em escala e podem ser desde pequenas hortas comunitárias até operações comerciais de grande porte, utilizando técnicas de cultivo sustentáveis e tecnologias inovadoras para otimizar o uso do espaço disponível, sendo mais utilizados os modelos de cultivo vertical e cultivo horizontal.

1.2.1. Tipos de fazendas urbanas:

“Ela pode ser aplicada diretamente no solo, usando hidroponia, canteiros suspensos ou vasos. Com um bom projeto, nenhuma área disponível precisará ser desperdiçada” (Blog Broto, 2023).

Sobre produção urbana, incluindo ou não fazendas urbanas, podemos afirmar que:

Um dos exemplos mais interessantes é o cultivo do meloeiro suspenso. Ele prova que não importa o tamanho da área: a colheita será uma realidade. Inclusive, variedades de melão, como cantaloupe, já são totalmente produzidas em ambiente protegido com estufas, e seus frutos são sustentados por redes até a maturação. Isso permite que as plantas sejam direcionadas de maneira vertical, e o espaço, mais otimizado (Blog Broto, 2023).

Há diferentes classificações de agricultura urbana, de acordo com o espaço e o objetivo da produção. São exemplos:

- Fazendas comerciais: com produção intensiva de médio a grande porte;
- Fazendas institucionais: voltadas a um público mais específico, como escolas, empresas e asilos;
- Agricultura não alimentar: voltada, por exemplo, ao paisagismo;
- Fazendas comunitárias: geralmente, de agricultura familiar;

- Jardins comunitários: planejados para o cultivo de plantas hortícolas, medicinais e ornamentais, e voltados para atividades de lazer e agroturismo.

1.2.2. Histórico das fazendas urbanas:

As fazendas urbanas, como exemplos de sustentabilidade, surgiram na década de 1960, vinculadas ao ambientalismo e aos movimentos em prol de um mundo mais natural, mais justo e mais solidário. De acordo com a fazenda Blueseeds (2024), o conceito mais próximo do que existe hoje, porém, nasceu em 1999 com o professor Dickson Despommier, na Universidade da Columbia, em Nova York e foi somente a partir dos anos 2010 que o movimento ganhou força, em especial nos países que já enfrentavam problemas com a escassez de terras férteis para o cultivo, seja por conta da falta de áreas rurais, como Japão, Taiwan e Cingapura, ou por causa das dificuldades climáticas, como o Canadá.

1.3. Segurança Alimentar e Banco de Alimentos

De acordo com o site gov.br (2023), os Bancos de Alimentos são estruturas físicas e/ou logísticas que ofertam o serviço de captação e/ou recepção e distribuição gratuita de gêneros alimentícios oriundos de doações de setores privados e/ou públicos e os destinam a instituições que atendem público em situação de vulnerabilidade social.

Segundo a Lei Orgânica de Segurança Alimentar e Nutricional – LOSAN (Lei nº 11.346, de 15 de setembro de 2006), por Segurança Alimentar e Nutricional – SAN entende-se a realização do direito de todos ao acesso regular e permanente a alimentos de qualidade, em quantidade suficiente, sem comprometer o acesso a outras necessidades essenciais, tendo como base práticas alimentares promotoras de saúde que respeitem a diversidade cultural e que sejam ambiental, cultural, econômica e socialmente sustentáveis.

1.4. ESG (Ambiental, Social e Governança) e ODS (Objetivos de Desenvolvimento Sustentável)

Segundo o site Ideias (2024), o ESG e ODS podem ser descritos como condutas, iniciativas e ações que as empresas praticam ligadas diretamente aos temas ambientais, sociais e de governança que visam a sustentabilidade a longo prazo e o impacto positivo na sociedade. Em poucas palavras, enquanto ESG é uma abordagem ampla para a sustentabilidade que foca nas práticas empresariais, os ODS são uma estrutura global mais abrangente que aborda os desafios mundiais.

1.4.1 Sobre o ESG

- Ambiental (E): Refere-se às práticas relacionadas ao meio ambiente, como gestão de resíduos, eficiência energética, mudanças climáticas e conservação da biodiversidade.
- Social (S): Envolve questões sociais, como direitos humanos, condições de trabalho, diversidade e inclusão, responsabilidade social corporativa e relações com a comunidade.
- Governança (G): Diz respeito às práticas de governança corporativa, transparência, ética nos negócios, estrutura de liderança e gestão de riscos.

1.4.2. Sobre o ODS

- Os ODS são uma iniciativa global estabelecida pelas Nações Unidas em 2015, composta por 17 objetivos e 169 metas destinados a abordar desafios globais, como pobreza, fome, saúde, educação, igualdade de gênero, água limpa, paz e justiça, entre outros.
- Os ODS extrapolam as práticas internas das empresas. Oferecem uma estrutura abrangente para orientar ações e políticas em níveis global, nacional e empresarial.

1.5. 30ª Conferência da ONU sobre Mudanças Climáticas

De acordo com o site Greenpeace Brasil (2023), a COP30, ou 30ª Conferência da ONU sobre Mudanças Climáticas, é um evento crucial para abordar o desafio do aquecimento global. É onde líderes de todo o mundo se reunirão para buscar soluções, discutir políticas e acordos para reduzir as emissões de gases de efeito estufa e lidar com os impactos das mudanças climáticas.

Na COP30 de 2025, é aceitável que o foco principal seria avaliar o progresso em relação aos objetivos estabelecidos no Acordo de Paris e em conferências climáticas anteriores. Esse é um momento-chave para pressionarmos os governos dos países amazônicos por mudanças que materializem um novo olhar para a Amazônia, um olhar de quem vive e conhece a região, e que seja capaz de superar a concentração de renda e pobreza, conviver com a biodiversidade, respeitar os direitos humanos e considerar os muitos modos de vida e culturas que compõem a Amazônia (Greenpeace Brasil, 2023).

1.6. Gestão de Pessoas

A fazenda urbana Terras Mil é amigável ao trabalhador, sempre auxiliando seus funcionários quando estes encontram dificuldades ao realizar suas tarefas, ou mesmo quando se encontram sem perspectiva para sua carreira profissional. Dentro da empresa é respeitado o desejo dos funcionários de subir de cargo, é enfatizado a meritocracia dentro de nossos portões, maior o esforço, maior a recompensa.

Dentro da fazenda existem muitas funções necessárias para o funcionamento dela, sendo essas ocupadas por profissionais competentes:

- Gerente: Encarregado da gestão geral da fazenda urbana, incluindo planejamento, supervisão de operações, gerenciamento de recursos humanos e financeiros, e tomada de decisões envolvendo assuntos internos.
- Técnicos Agrícolas: Profissionais com conhecimentos em ciências agrícolas, responsáveis pelo planejamento e implementação de práticas agrícolas, monitoramento do solo e das plantas, controle de pragas e doenças, e otimização da produção.
- Agricultores: Encarregados da manutenção diária da fazenda urbana, incluindo plantio, colheita, irrigação e cuidados gerais com as plantas.
- Especialistas em Sistemas de Irrigação: Responsáveis pelo projeto, instalação e manutenção de sistemas do sistema de gotejamento, também responsáveis por pesquisar outras formas de irrigação compatível com a fazenda urbana.
- Engenheiros Elétricos: Encarregados da instalação, manutenção e reparo de equipamentos elétricos, como sistemas de iluminação artificial com lâmpadas UV, sistemas de controle climático e sistemas de automação.
- Equipe de Limpeza: Responsável pela limpeza e manutenção das instalações da fazenda urbana, incluindo áreas de cultivo, estufas, equipamentos e espaços comuns.

- Equipe de Segurança: Encarregada de garantir a segurança das instalações da fazenda urbana, prevenindo roubos, vandalismos e outros incidentes.

1.7. Economia e Viabilidade

A estimativa de viabilidade se iniciou calculando os custos iniciais, incluindo valor da aquisição do terreno da fazenda urbana Terras Mil, construção de estrutura, instalação do sistema de irrigação, construção da estufa, compras de equipamentos, insumos e sementes agrícolas.

Após a estimativa de gastos, é calculado o potencial lucro da fazenda urbana, nesta etapa é levado em conta em consideração o preço de mercado, demanda de certo produto, canais de distribuição disponíveis (clientes, feiras etc.) e a capacidade de produção dos produtos dentro da fazenda.

Calculando o fluxo do plantio é possível deduzir o fluxo de caixa da fazenda ao longo dos anos, considerando os preços sazonais e certos riscos de perda de produção e desvalorização do mercado.

Seguindo esse caminho, é finalmente possível ter certa ideia se a fazenda consegue ou não dar lucro ao proprietário. Lucro este que foi suficientemente satisfatório para os responsáveis pela empresa.

1.8. Regras de Negócio

Regras de negócio são definidas como diretrizes importantes para o funcionamento eficaz de qualquer empresa ou empreendimento, elas ditam o rumo que a fazenda urbana Terras Mil deve seguir. Dito isso, a fazenda urbana Terras Mil possui as seguintes regras de negócio:

- Segurança Alimentar: A fazenda urbana Terras Mil garante que todos os produtos cultivados são da mais alta qualidade, garantindo um consumo seguro e contribuindo com a segurança alimentar.
- Segurança Ambiental: A fazenda urbana Terras Mil Opera de maneira sustentável, minimizando o uso de recursos naturais, como água (irrigação por gotejamento) e energia (captação solar), e implementando práticas agrícolas que não prejudiciais à saúde do solo.

- **Uso Responsável de Insumos:** A fazenda urbana Terras Mil Utiliza insumos agrícolas, como fertilizantes e pesticidas, de forma responsável, buscando alternativas orgânicas e de baixo impacto ambiental.
- **Colaboração com a Comunidade:** A fazenda urbana Terras Mil estabelece parcerias e colaborações com diversos fornecedores, tais como agricultores familiares, organizações sem fins lucrativos e outros negócios locais, para promover a segurança alimentar e o desenvolvimento sustentável.
- **Responsabilidade Social e Inclusão:** A fazenda urbana Terras Mil visa promover a inclusão social e econômica, oferecendo diversas oportunidades de emprego dentro da fazenda urbana e capacitação para membros da comunidade local, especialmente aqueles em situação de vulnerabilidade social e que desejam trabalhar conosco.
- **Inovação e Pesquisa:** A fazenda urbana Terras Mil possui investimentos frequentes em pesquisa e desenvolvimento para aprimorar constantemente as práticas agrícolas, aumentar a produtividade e reduzir o impacto ambiental da fazenda urbana, tal como pragas, fungos etc.
- **Respeito às Regulamentações:** Cumprir todas as regulamentações locais, estaduais e federais relacionadas à produção agrícola, segurança alimentar e proteção ambiental.

2. CARACTERIZAÇÃO DA EMPRESA

Sendo consideradas as regras de negócio apresentadas no tópico anterior, a fazenda urbana Terras Mil se importa com o desenvolvimento sustentável e eficiente dos alimentos e produtos sendo produzidos dentro de suas instalações.

Para seguir os valores da empresa é necessário o auxílio de um sistema que facilite o gerenciamento da fazenda urbana, facilitando a administração e redirecionando a atenção necessária para outros pontos da empresa, foram pensados os processos de negócio

Login e Hierarquia: Necessário um sistema de identificação que distinga funcionários padrão dos funcionários com mais poder de escolha dentro da empresa.

Tarefas: É de grande interesse um sistema que possibilite os funcionários verificar as tarefas pendentes e concluídas que estes possuem, tal como possibilitar que os gerentes criem tarefas novas.

Fornecedor: Um sistema de visualização de fornecedores é essencial para a organização dentro da empresa, tal como a facilidade em editar dados caso o fornecedor não faça mais negócio conosco.

Clientes: É preciso uma função que administre os clientes, indicando o status da compra, dados do cliente, possibilidade de edição e arquivamento do cliente.

Produção: O sistema de produção da fazenda urbana Terras Mil deve possuir calendário informando a estimativa da colheita, informar a produção relacionada aos tipos de produtos disponíveis na fazenda urbana (frutas, hortaliças etc.). A produção é feita tendo como base o histórico de compra de algum cliente da empresa.

3. DESENVOLVIMENTO DO PROJETO

Antes do início do desenvolvimento e levantamento de requisitos e análise de requisitos não funcionais, foi definido a metodologia ágil utilizada no desenvolvimento do projeto, levando em conta a eficiência e velocidade de conclusão das tarefas no desenvolvimento do software.

3.1. Metodologia Utilizada

A metodologia utilizada pela equipe de desenvolvido acabou por sendo a SCRUM, tendo sido escolhida pela adaptabilidade em gerenciar os processos do desenvolvimento e pela adaptabilidade em alterar algo caso ocorra imprevistos.

O SCRUM é composto por um PO (Product Owner), um Scrum Master e o restante da equipe de desenvolvimento, sendo o PO o responsável por ser a “ponte” entre os

desenvolvedores e o cliente do produto, as informações que o PO coleta são importantes para a formação dos requisitos e a construção do backlog do sistema.

O Scrum Master é responsável pela equipe de desenvolvimento, garantindo que os membros não desviem de suas funções ou que deixam de fazer uma tarefa a tempo. Uma vez por dia é dirigida uma reunião breve pelo Scrum Master, onde o dev team deve expressar o progresso realizado durante o dia e sanar certas dúvidas, expressar dificuldades no desenvolvimento etc.

3.2. Levantamento de Requisitos

Para a construção do sistema de gerenciamento da empresa, foi levado em conta os seguintes requisitos:

- Tela de Login;
- Adicionar tarefas;
- Listar tarefas pendentes e completas;
- Editar tarefas;
- Adicionar fornecedores;
- Listar fornecedores;
- Mudar informações de fornecedores;
- Informar produção;
- Adicionar produção;
- Adicionar clientes;
- Lista de Clientes;
- Editar Clientes;
- Gráfico de produção;

- Calendário de produção;
- Protótipo de telas.
- Para entendimento de como o sistema deve ser executado, foi pensado nos seguintes requisitos não funcionais:
- Implementar um sistema de controle de acesso que permita diferentes níveis de acesso e permissões com base na hierarquia dos usuários.
- Garantir que todos os usuários sejam autenticados e autorizados antes de acessar informações sensíveis ou realizar operações críticas.
- O sistema deve responder a solicitações de usuários em um tempo aceitável.
- Garantir que todos os usuários sejam autenticados e autorizados antes de realizar operações críticas.
- A interface do usuário deve ser intuitiva e fácil de usar, minimizando a necessidade de treinamento extensivo.
- Manter documentação detalhada e atualizada do sistema.

3.3. Planejamento de Sprints

Após o levantamento e análise de requisitos, foi separado pela equipe de desenvolvimento e Scrum Master a Sprint Backlog, simbolizando quantas sprints a equipe tem disponível para desenvolver o sistema e o que deve ser feito durante a sprint.

A tabela 1 mostra o backlog desenvolvido pela equipe de desenvolvimento.

Tabela 1 - Sprint Backlog.

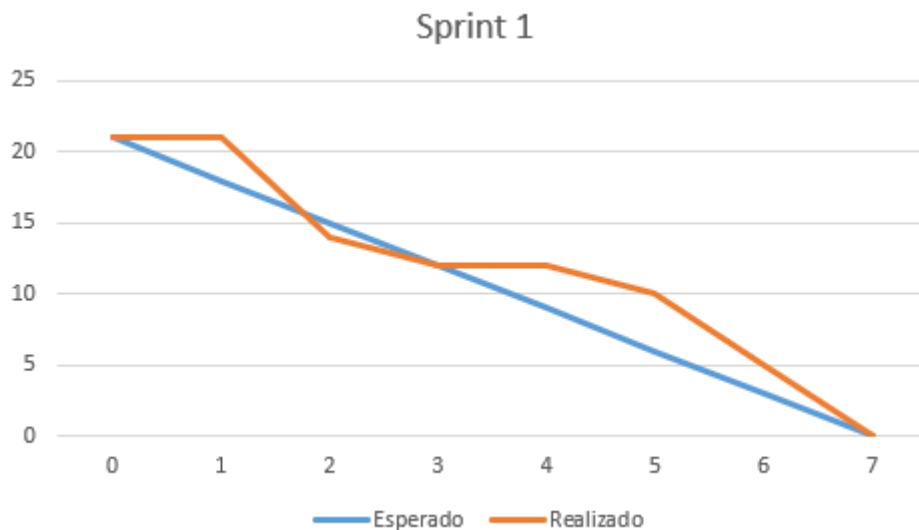
| Sprint 1 | | | | |
|---------------------------------------|-------------|------------|-----------|--------|
| Tarefa | Responsável | Prioridade | Status | Tempo |
| Identificar os requisitos | Henrique | Alta | Concluído | 7 dias |
| Pesquisa sobre fazendas urbanas | Matheus | Média | Concluído | 8 dias |
| Sprint 2 | | | | |
| Tarefa | Responsável | Prioridade | Status | Tempo |
| Tela de Login e Autenticação | Luís | Média | Concluído | 7 dias |
| Sprint 3 | | | | |
| Tarefa | Responsável | Prioridade | Status | Tempo |
| Diagrama de caso de uso | Henrique | Alta | Concluído | 7 dias |
| Diagrama de classe | Matheus | Alta | Concluído | 7 dias |
| Tela dos fornecedores | Luís | Média | Concluído | 7 dias |
| Sprint 4 | | | | |
| Tarefa | Responsável | Prioridade | Status | Tempo |
| Tela de tarefas diárias | Luís | Média | Concluído | 7 dias |
| Tela de tarefas semanais | Luís | Média | Concluído | 7 dias |
| Sprint 5 | | | | |
| Tarefa | Responsável | Prioridade | Status | Tempo |
| Tela de clientes | Luís | Média | Concluído | 7 dias |
| Tela de registro de clientes | Luís | Média | Concluído | 7 dias |
| Sprint 6 | | | | |
| Tarefa | Responsável | Prioridade | Status | Tempo |
| Tela informações Fornecedores | Luís | Média | Concluído | 7 dias |
| Tela Produção | Luís | Média | Concluído | 7 dias |
| Sprint 7 | | | | |
| Tarefa | Responsável | Prioridade | Status | Tempo |
| Tela com informações sobre a produção | Luís | Média | Concluído | 7 dias |
| Diagrama Entidade-Relacionamento | João | Alta | Concluído | 7 dias |
| Início da Documentação | Lucas | Alta | Concluído | 7 dias |
| Sprint 8 | | | | |
| Tarefa | Responsável | Prioridade | Status | Tempo |
| CRUD | Matheus | Alta | Concluído | 7 dias |
| Script do Banco de Dados | Ruan | Alta | Concluído | 7 dias |
| Diagrama de Sequência | Lucas | Alta | Concluído | 7 dias |
| Diagrama de implementação | Lucas | Alta | Concluído | 7 dias |

Fonte: Autor (2024).

Após a resolução de uma sprint é possível criar o gráfico de Sprint burndown, que referencia os dados de quanto a equipe trabalhou durante o tempo da sprint.

O gráfico 1 representa a sprint burndown realizada na primeira Sprint.

Gráfico 1 – Sprint Burndown.



Fonte: Autor (2024).

É perceptível que, embora a equipe tenha encontrado problemas no começo da sprint, foi possível entregar as 21 tarefas esperadas no final dos sete dias.

3.4. Diagramas UML

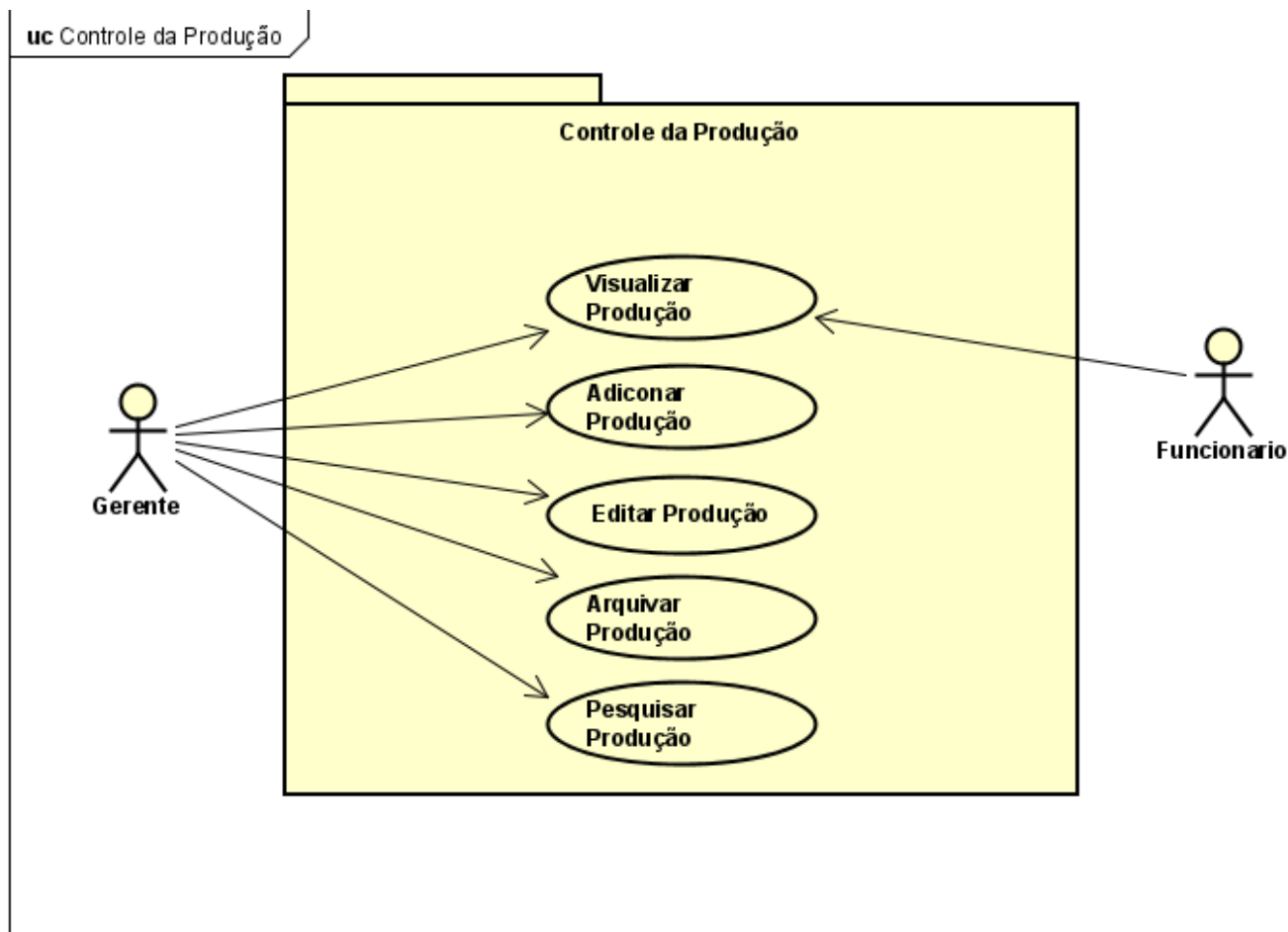
Foi levado em consideração que os diagramas UML são ferramentas essenciais para o desenvolvimento de software, trazendo benefícios significativos em termos de visualização, especificação, construção e documentação, foi considerado para o desenvolvimento do sistema de gerenciamento da fazenda urbana os diagramas UML de caso de uso, sequências, implantação e classe.

3.4.1. Diagramas de Caso de Uso

Foi considerado que um diagrama de caso de uso mostra o comportamento esperado do sistema, porém não indicar a sequência das ações

A figura 1 detalha o caso de uso dos usuários da empresa durante o sistema de controle de produção.

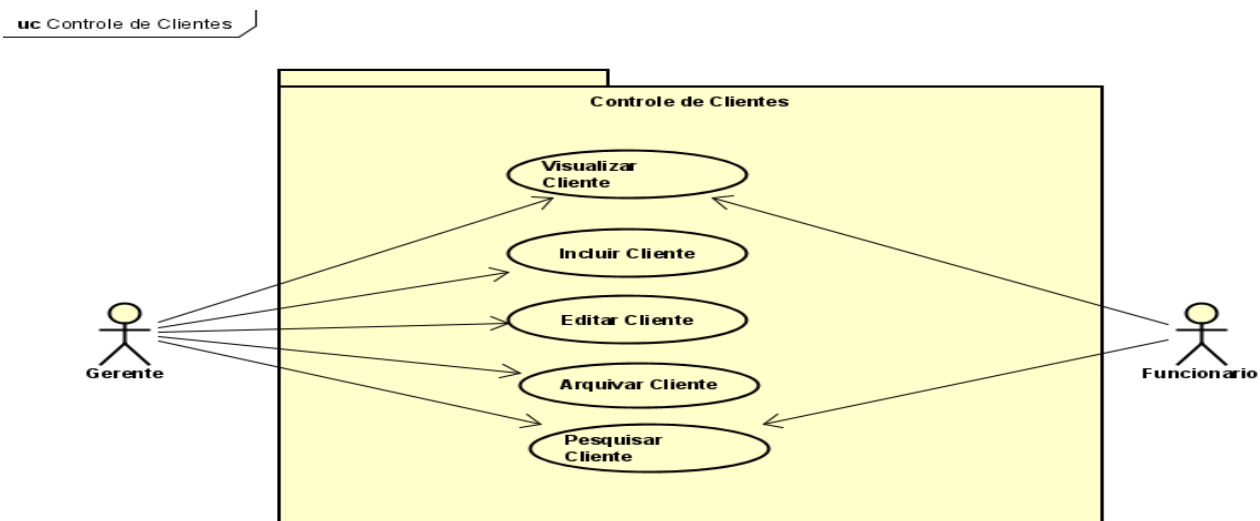
Figura 1 – Caso de uso da produção



Fonte: Autor (2024).

A figura 2 detalha o caso de uso dos usuários da empresa durante o sistema de controle de clientes.

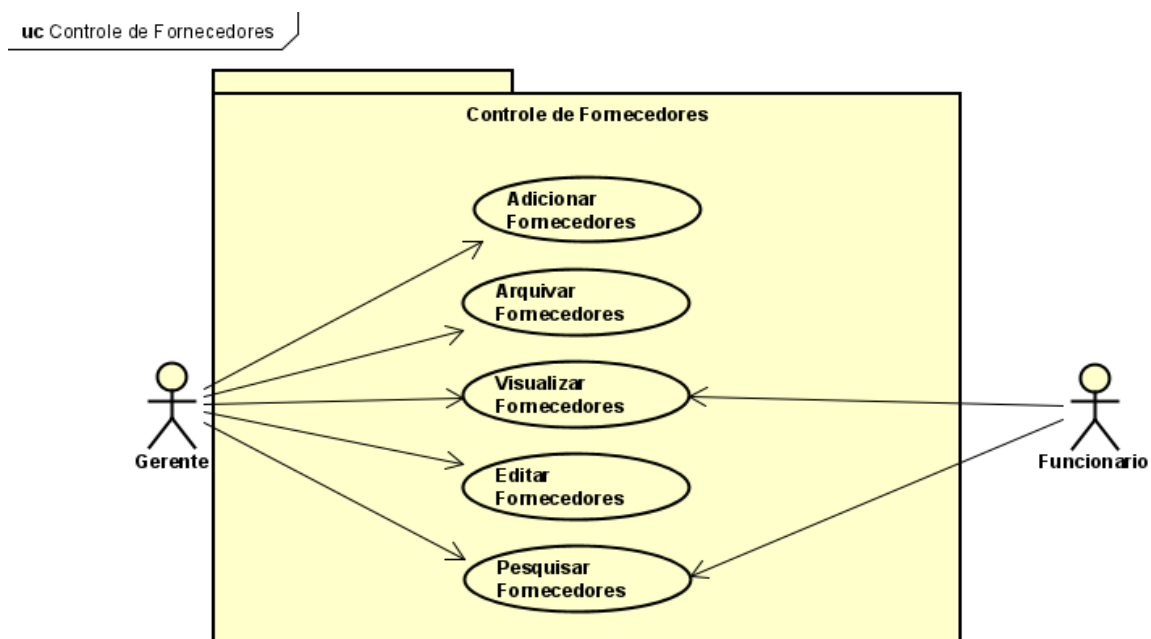
Figura 2 - Caso de uso de controle de clientes.



Fonte: Autor (2024).

A figura 3 detalha o caso de uso do sistema de fornecedores

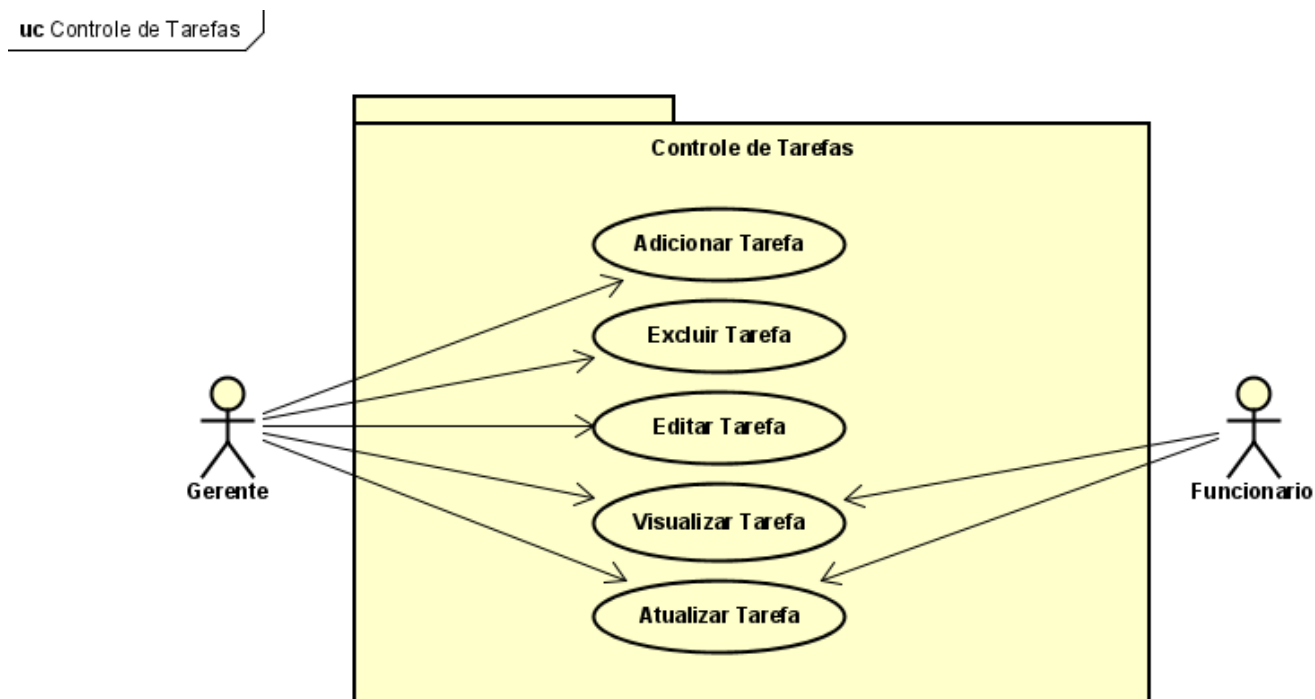
Figura 3 – Caso de uso do controle de fornecedores.



Fonte: Autor (2024).

A figura 4 detalha o caso de uso do sistema de controle de tarefas.

Figura 4 – Caso de uso de tarefas.



Fonte: Autor (2024).

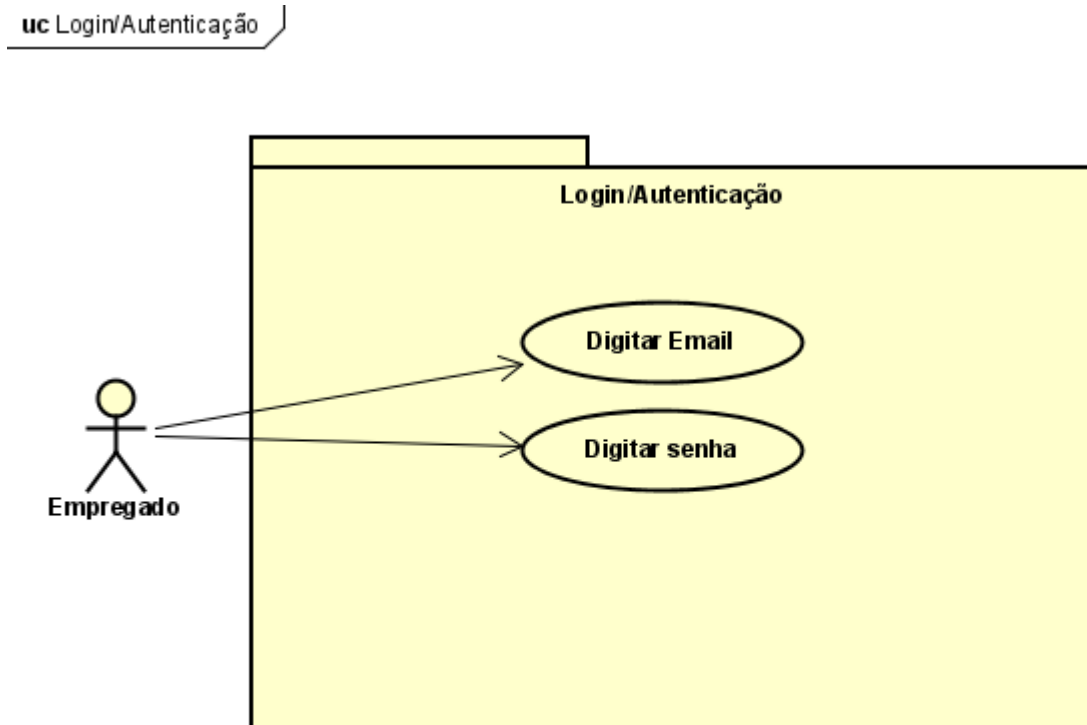
Foi utilizado a lógica de que o gerente deve possuir autoridade máxima dentro do sistema, podendo tomar decisões importantes como adicionar alguma entidade no banco de dados do sistema ou arquivar fornecedores, clientes ou a produção de certa hortaliça ou frutas.

Levou-se em consideração os sentimentos do funcionário comum, tendo em vista que inicialmente o funcionário padrão poderia apenas visualizar as tarefas, sendo incapaz de visualizar qualquer outra parte do sistema, essa ideia porém, não foi considerada mais adiante, tendo a equipe de desenvolvimento entendido que o funcionário deve se sentir parte da empresa, e que privar ele de visualizar certas atividades dentro da empresa leva ao descontentamento seguido de trabalho malfeito.

Entretanto, antes do sistema identificar quem é funcionário ou gerente, é preciso de um sistema de login que faça a distinção entre os dois.

A figura 5 representa o caso de uso do sistema de login.

Figura 5 – Caso de uso do login.



Fonte: Autor (2024).

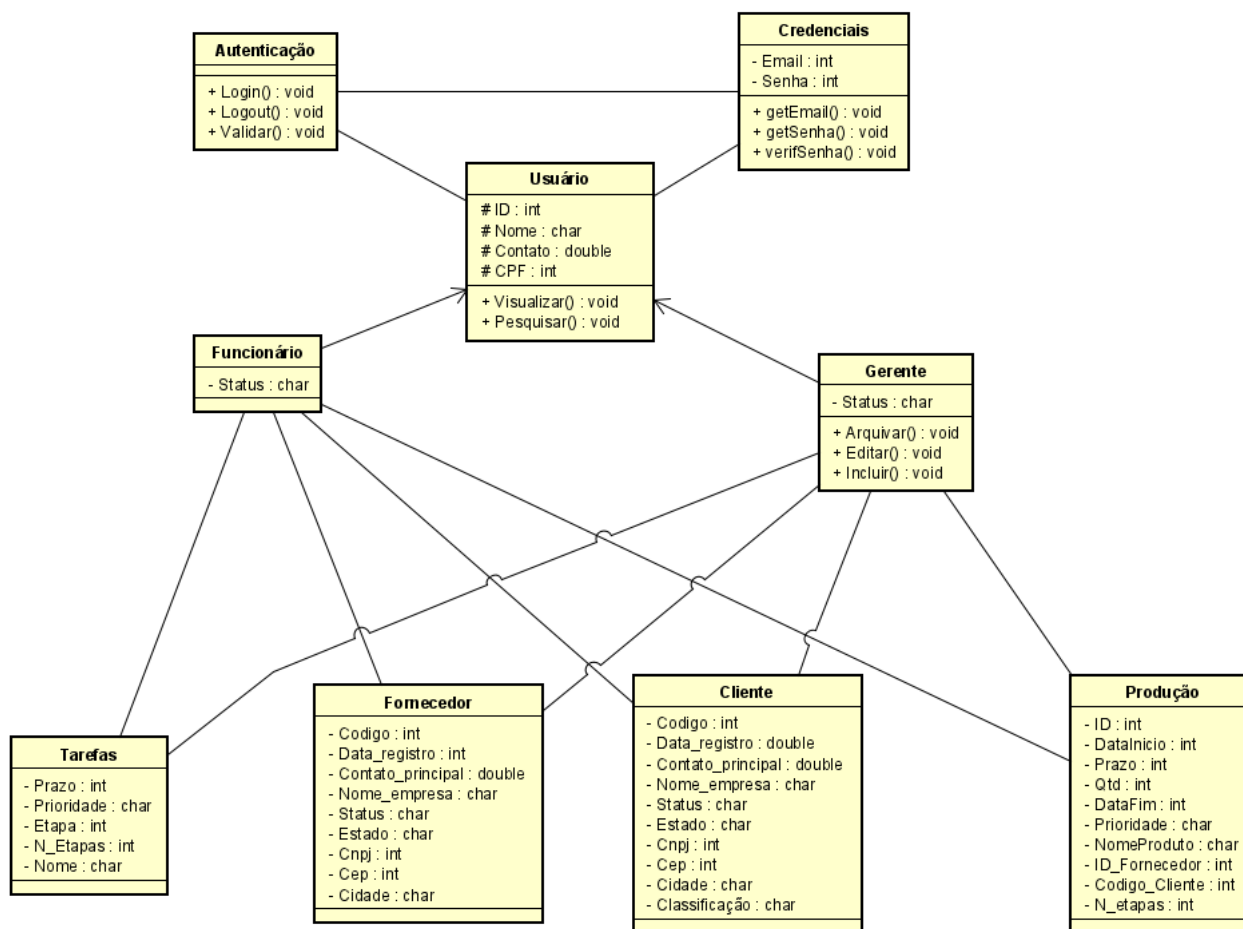
Tendo idealizado todos os diagramas de uso do usuário em cada sistema, é encerrado a necessidade de mais envolvimento nessa etapa dos diagramas UML.

3.4.2. Diagrama de Classe

Após criar todos os diagramas relacionados com o que o sistema deve interagir e comportas, é a próxima etapa fazer um aprofundamento na estrutura do sistema, identificar as classes e relacionamento entre elas, promover heranças e polimorfismos, bem como atribuir métodos e atributos para as classes selecionadas.

A figura 6 mostra o diagrama de classe do sistema de gerenciamento.

Figura 6 – Diagrama de Classe.



Fonte: Autor (2024).

Foi pensado em um sistema de gerenciamento apenas para membros ativos dentro da fazenda urbana Terras Mi, ou seja, sem interação por parte de terceiros.

Tomando como base esse critério, é possível identificar no diagrama acima a classe “usuário”, ou seja, todos os empregados da fazenda urbana Terras Mil são, até o período de autenticação, usuários “iguais”.

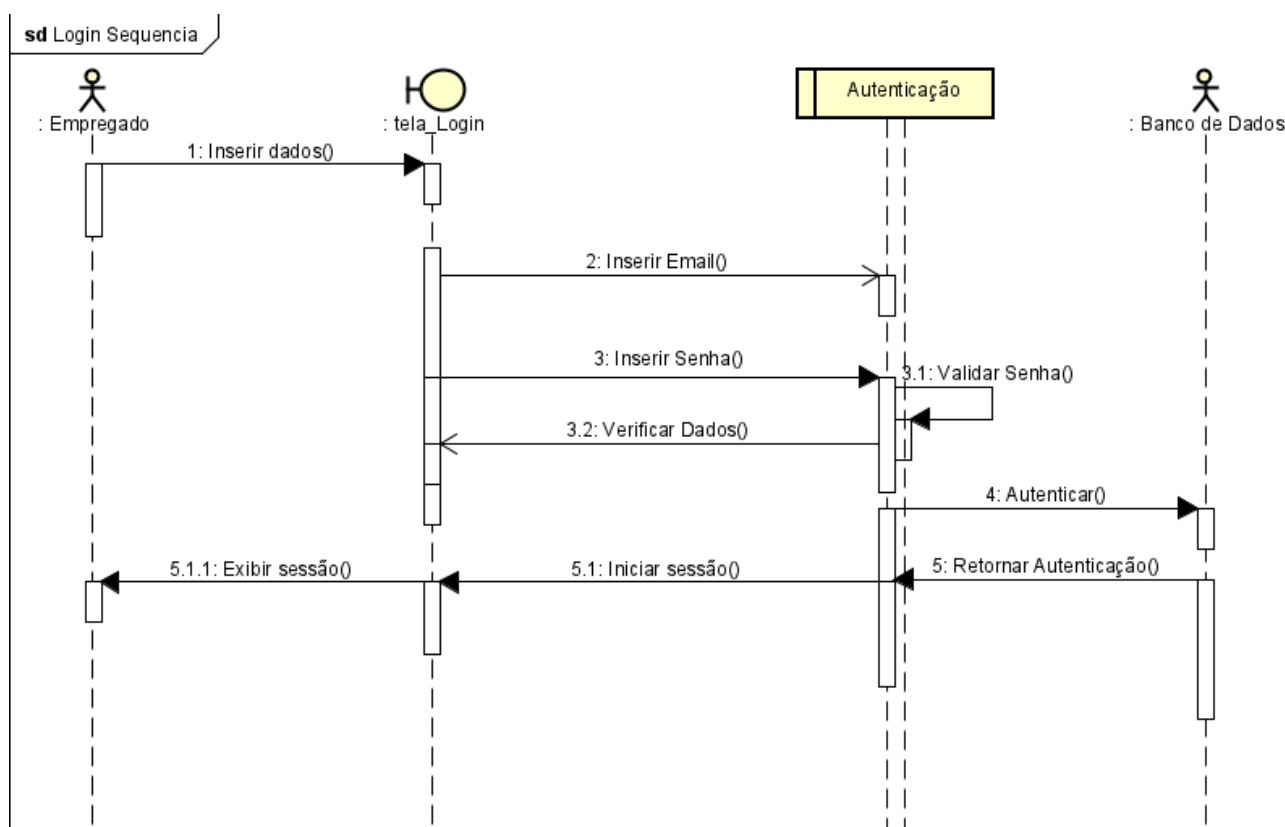
Após a autenticação de cargo o usuário é redirecionado à sua devida hierarquia, sendo ele gerente ou funcionário (a validação será baseada no email do usuário. As classes “gerente” e “funcionário” herdam da classe “usuário” seus principais atributos e métodos, foi levado em consideração a ideia menciona no tópico anterior, o gerente pode tomar vários cursos dentro das classes do sistema enquanto o funcionário padrão apenas visualiza e pesquisa os clientes, fornecedores etc.

3.4.3. Diagrama de Sequência

Os diagramas de sequência representam uma parte mais avançada do UML, pois descreve como e quando um grupo interage como sistema (classe, boundary, ator etc), é usado mais comumente para documentar o trajeto de um trecho do software.

A figura 7 apresenta o diagrama de sequência do sistema de login.

Figura 7 – Sequência do Login.



Fonte: Autor (2024).

É possível visualizar do diagrama o ator “Empregado” que insere os dados na interface de login, a interface de login envia os dados inseridos para a classe “autenticação” que por sua vez é responsável por verificar se o email e senha são válidos e retornar mensagem de invalidez caso os não estejam corretos na validação.

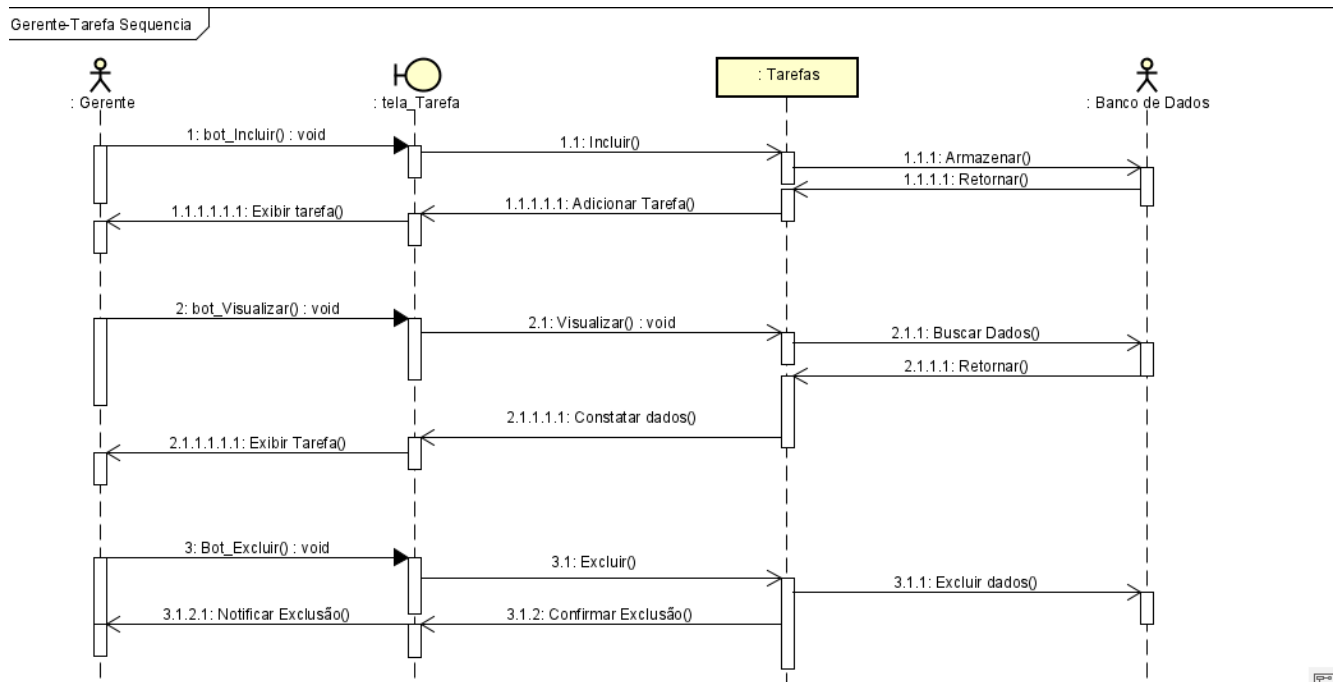
Caso os dados estejam corretos, a classe “autenticação” consulta o banco de dados para confirmação da veracidade dos dados, após os dados serem confirmados é criada uma sessão baseada na hierarquia do funcionário/gerente.

Após a exibição da sessão, o usuário pode entrar nas abas do sistema de gerenciamento, podendo escolher entre as telas de tarefas, clientes, fornecedores e

produção. Independentemente da escolha os diagramas de sequência mal são diferenciados, por isso será documentado apenas o com mais função.

A figura 8 ilustra a primeira metade do diagrama de sequência da tela de tarefas.

Figura 8 – Diagrama de Sequência da tela de tarefas.



Fonte: Autor (2024).

No diagrama o ator “gerente” tem seis opções na interface, sendo elas Incluir, Visualizar, Excluir, Editar, Pesquisa e Atualizar.

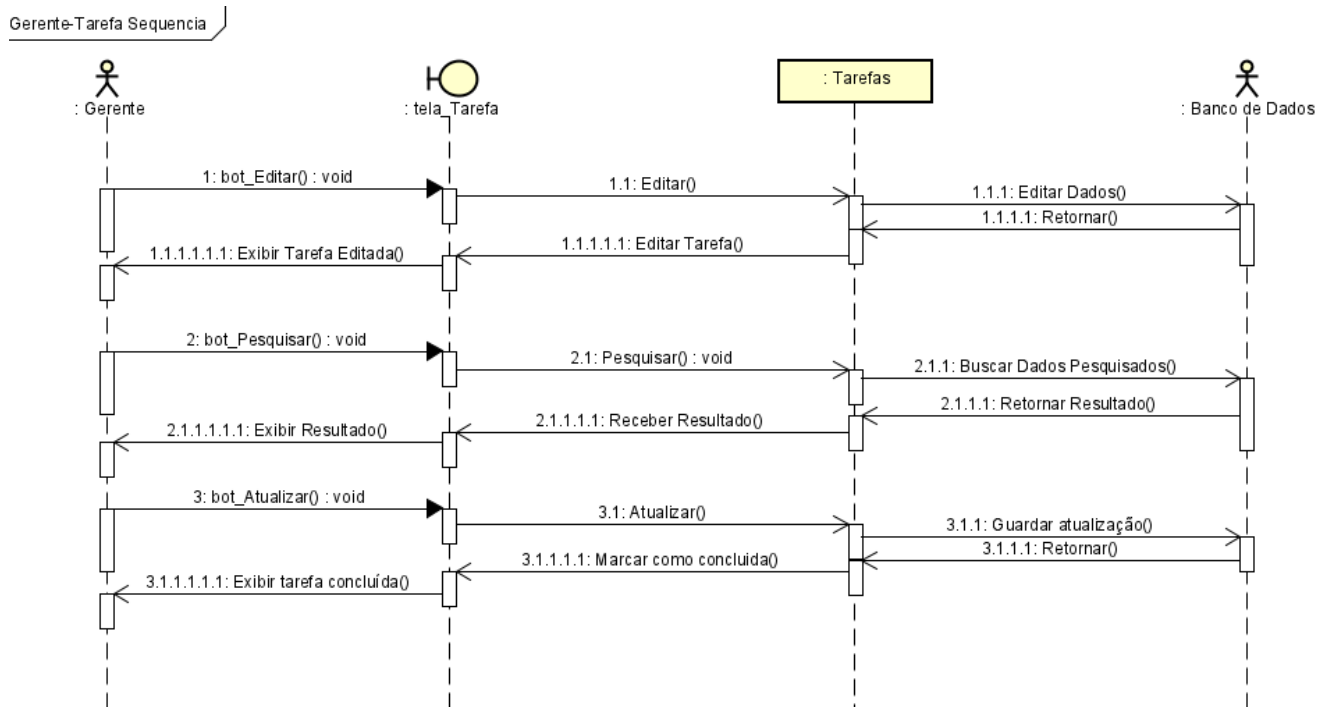
A função “incluir” faz o gerente inserir os dados para adicionar uma nova tarefa no banco de dados, o banco de dados retorna a tarefa adicionada e a exibe para o usuário.

A função “visualizar” busca os dados da tarefa selecionada no banco de dados, o banco retorna os dados e exibe para o usuário.

A função “excluir” apaga os dados da tarefa no banco de dados e devolve uma mensagem avisando que a tarefa foi excluída com sucesso

A figura 9 ilustra a segunda metade do diagrama de sequência da tela de tarefas.

Figura 9 – Diagrama de Sequência da Tela de tarefas.



Fonte: Autor (2024).

A função “editar” busca os dados recentes no banco de dados e os disponibiliza para edição, depois os atualiza e retorna uma mensagem de confirmação para o usuário.

A função “pesquisar” busca os dados inseridos na barra de pesquisa, consulta o banco de dados, retorna o resultado e exibe para o usuário.

A última função, “atualizar”, marca uma caixa de texto na informação da tarefa então armazena os dados e marca a tarefa como concluída.

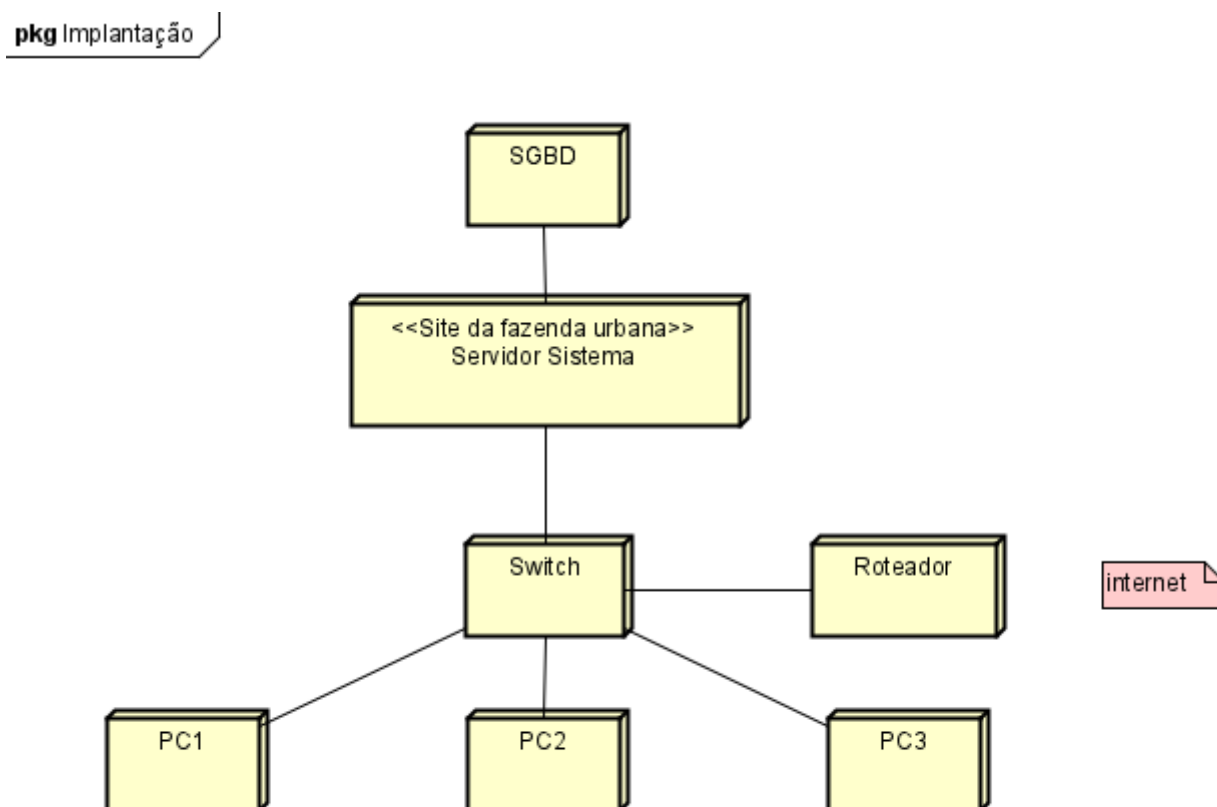
Como mencionado anteriormente, não há necessidade de explicar todos os diagramas de sequencias de todas as telas, tendo em vista que além da função “atualizar”, que é exclusiva da tela de tarefa, todas as funções possuem os mesmos métodos.

3.4.4. Diagrama de Implantação

O diagrama de implantação é o próximo passo após identificar o escopo do projeto, o projeto de sistema de gerenciamento da fazenda urbana Terras Mil foi devido como um sistema desktop.

A figura 10 representa o diagrama de implantação da Terras Mil.

Figura 10 – Diagrama de Implantação.



Fonte: Autor (2024).

Foi considerado um sistema “padrão” de redes para o funcionamento do sistema.

3.5. Prototipagem de Telas

Para a criação do protótipo de telas foi utilizado a ferramenta Figma, foi desenvolvido utilizando como princípio básico os fundamentos da UX e as Dez Heurísticas de Nielsen, buscando ao máximo proporcionar uma boa usabilidade para o usuário.

A figura 10 exibe a tela de login do sistema.

Figura 11 – Tela de Login.



Faça o seu Login
Insira seu email e senha

email@dominio.com

Entrar

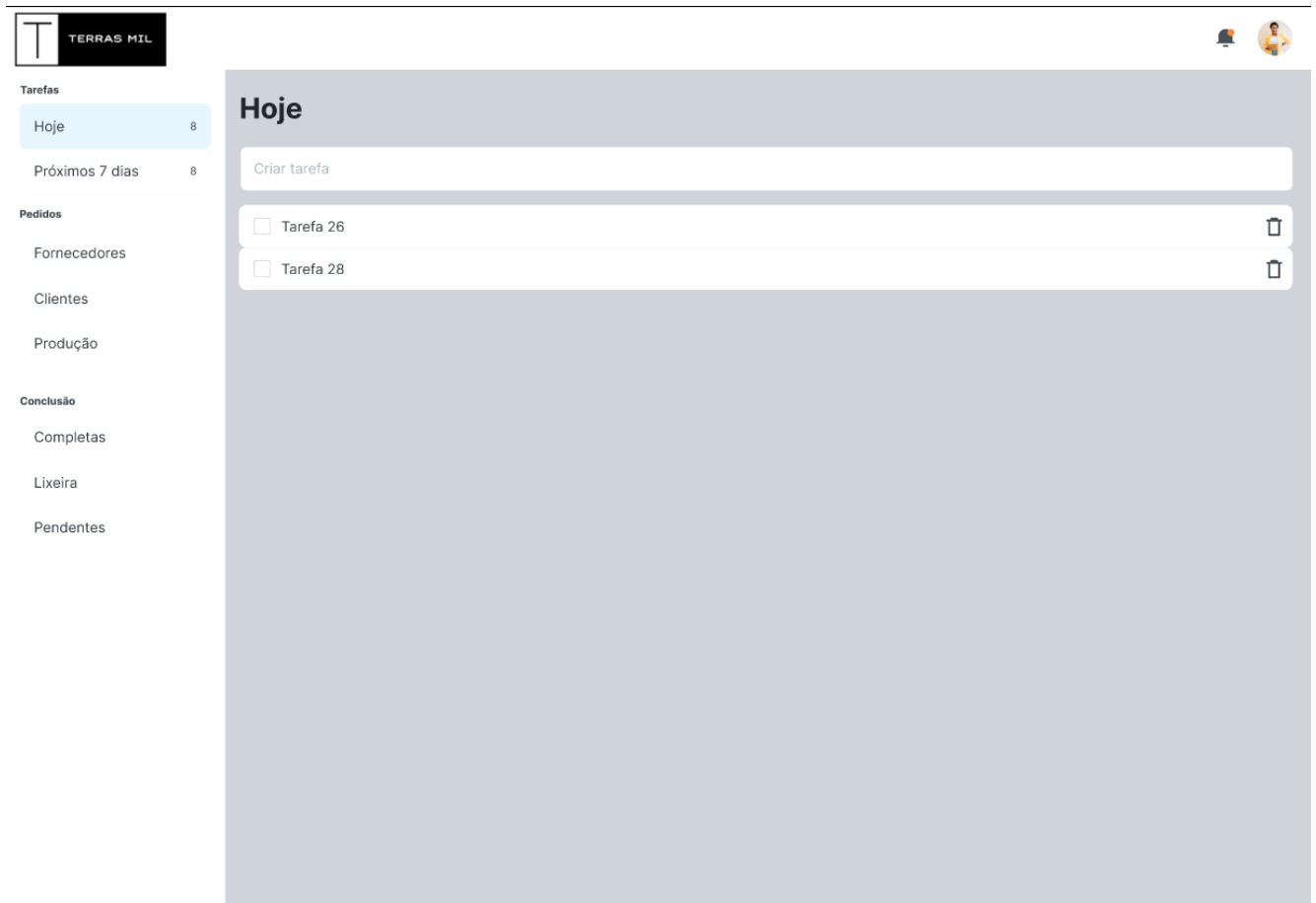
[Perdi meu acesso](#)

Fonte: Ator (2024).

Foi idealizado uma tela de login simples, com opção de auxílio caso o usuário perca a senha, será exibido uma mensagem para o usuário buscar o RH.

A figura 12 mostra a tela inicial do sistema.

Figura 12 – Tela inicial de tarefas.



Fonte: Autor (2024).

Tela inicial que exibe tarefas diárias ou semanais, com opção de navegar para outras funções.

A figura 13 ilustra a tela de fornecedores do sistema.

Figura 13 – Tela de Fornecedores.

TERRAS MIL

Tarefas

Hoje 8

Próximos 7 dias 8

Pedidos

Fornecedores

Clientes

Produção

Conclusão

Completas















Lixeira

Pendentes

Fornecedores

Pesquisar fornecedor

Inserir fornecedor

| Código | Status | Classificação | Entrega | Ações |
|--------|---------|---------------|-----------|---|
| 90567 | Ativo | Tradicional | Pendente |   |
| 90567 | Ativo | Especial | Concluído |   |
| 90567 | Inativo | Especial | Pendente |   |
| 90567 | Ativo | Tradicional | Concluído |   |
| 90567 | Ativo | Especial | Concluído |   |
| 90567 | Ativo | Tradicional | Pendente |   |
| 90567 | Inativo | Tradicional | Pendente |   |

Fonte: Autor (2024).

Tela com opção de inserir novo fornecedor, visualizar fornecedor, arquivar e editar.

A figura 14 ilustra a tela de clientes.

Figura 14 – Tela de Clientes.

TERRAS MIL

Tarefas

Hoje 8

Próximos 7 dias 8

Pedidos

Fornecedores

Cientes

Produção

Conclusão

Completas















Lixeira

Pendentes

Clientes

Pesquisar cliente

Inserir cliente

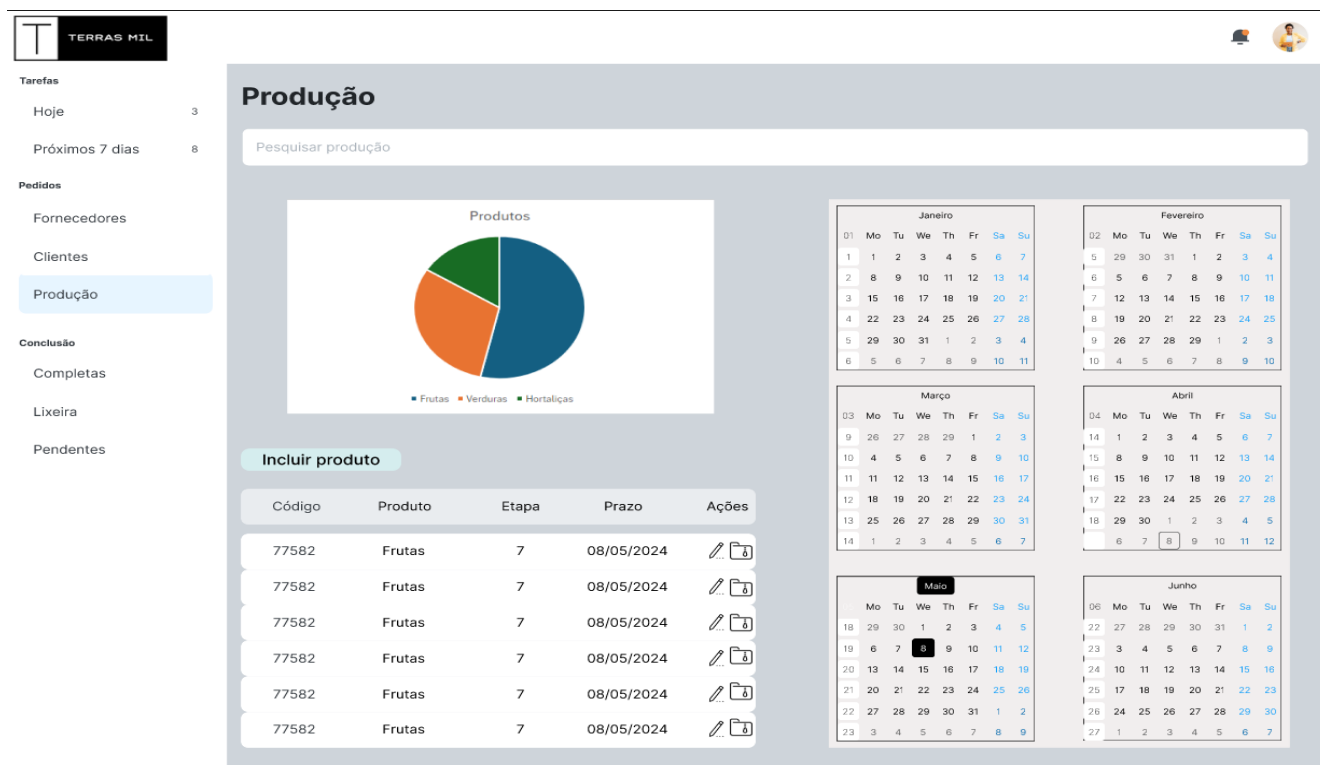
| Código | Status | Classificação | Compra | Ações |
|--------|--------|---------------|----------|---|
| 76218 | Ativo | A | Efetuada |   |
| 76218 | Ativo | A | Efetuada |   |
| 76218 | Ativo | A | Efetuada |   |
| 76218 | Ativo | A | Efetuada |   |
| 76218 | Ativo | A | Efetuada |   |
| 76218 | Ativo | A | Efetuada |   |
| 76218 | Ativo | A | Efetuada |   |

Fonte: Autor (2024).

Tela minimalista com informações dos clientes recorrentes da empresa.

A figura 14 ilustra a tela da produção da fazenda urbana.

Figura 15 – Tela de Produção.



Fonte: Autor (2024).

Tela de produção apresentando gráfico de produção, data estimada da produção e informação da produção.

3.6. CRUD (Create Read Update Delete)

O Crud foi desenvolvido com foco de simular a parte básica do armazenamento, inclusão, alteração e arquivamento dos dados da empresa Terras Mil.

Ele possui um menu principal no qual o administrador (Gerente) poderá selecionar qual dos dados (Cliente, Fornecedor, Tarefas, Funcionário e Produção) ele deseja modificar, arquivar, visualizar e incluir, além de também ter as mesmas permissões para mexer com a parte de produção do site.

A figura 16 representa um trecho do Crud.

Figura 16 – Classe Programa

```
4
5 namespace crudreal
6 {
7     0 referências
8     class Program
9     {
10        0 referências
11        static void Main(string[] args)
12        {
13            List<Funcionario> funcionarios = new List<Funcionario>();
14            List<Tarefas> tarefas = new List<Tarefas>();
15            List<Fornecedor> fornecedores = new List<Fornecedor>();
16            List<Cliente> clientes = new List<Cliente>();
17            List<Producao> producoes = new List<Producao>();
18            Gerente gerente = new Gerente();
19            int opcao = -1;
20
21            while (opcao != 0)
22            {
23                Console.WriteLine("Escolha uma opção:");
24                Console.WriteLine("1. Adicionar Funcionário");
25                Console.WriteLine("2. Pesquisar Funcionário");
26                Console.WriteLine("3. Editar Funcionário");
27                Console.WriteLine("4. Arquivar Funcionário");
28                Console.WriteLine("5. Adicionar Tarefa");
29                Console.WriteLine("6. Pesquisar Tarefa");
30                Console.WriteLine("7. Editar Tarefa");
31                Console.WriteLine("8. Arquivar Tarefa");
32                Console.WriteLine("9. Adicionar Fornecedor");
33                Console.WriteLine("10. Pesquisar Fornecedor");
34            }
35        }
36    }
37 }
```

Fonte: Autor (2024).

Foram criadas classes no VisualStudio com base no nosso diagrama de classe desenvolvido, além de ter uma classe Program na onde seria a “principal” responsável por executar as funcionalidades do sistema.

A figura 17 mostra um trecho dela abaixo.

Figura 17 – (a) primeira metade do Crud. (b) segunda metade do Crud.

```

while (opcao != 0)
{
    Console.WriteLine("Escolha uma opção:");
    Console.WriteLine("1. Adicionar Funcionário");
    Console.WriteLine("2. Pesquisar Funcionário");
    Console.WriteLine("3. Editar Funcionário");
    Console.WriteLine("4. Arquivar Funcionário");
    Console.WriteLine("5. Adicionar Tarefa");
    Console.WriteLine("6. Pesquisar Tarefa");
    Console.WriteLine("7. Editar Tarefa");
    Console.WriteLine("8. Arquivar Tarefa");
    Console.WriteLine("9. Adicionar Fornecedor");
    Console.WriteLine("10. Pesquisar Fornecedor");
    Console.WriteLine("11. Editar Fornecedor");
    Console.WriteLine("12. Arquivar Fornecedor");
    Console.WriteLine("13. Adicionar Cliente");
    Console.WriteLine("14. Pesquisar Cliente");
    Console.WriteLine("15. Editar Cliente");
    Console.WriteLine("16. Arquivar Cliente");
    Console.WriteLine("17. Adicionar Produção");
    Console.WriteLine("18. Pesquisar Produção");
    Console.WriteLine("19. Editar Produção");
    Console.WriteLine("20. Arquivar Produção");
    Console.WriteLine("0. Sair");
    opcao = int.Parse(Console.ReadLine());

    switch (opcao)
    {
        case 1:
            gerente.AdicionarFuncionario(funcionarios);
            break;
        case 2:
            gerente.PesquisarFuncionario(funcionarios);
            break;
        case 3:
            gerente.EditarFuncionario(funcionarios);
            break;
        case 4:
            gerente.ArquivarFuncionario(funcionarios);
            break;
        case 5:
            gerente.AdicionarTarefa(tarefas);
            break;
        case 6:
            gerente.PesquisarTarefa(tarefas);
            break;
        case 7:
            gerente.EditarTarefa(tarefas);
            break;
        case 8:
            gerente.ArquivarTarefa(tarefas);
            break;
        case 9:
            gerente.AdicionarFornecedor(fornecedores);
            break;
    }
}

```

Fonte: Autor (2024).

Segue na figura 18 abaixo um pouco da estrutura da classe Gerente que é a nossa maior classe, com seus atributos e métodos, lembrando que a classe gerente herda da classe pai Usuário.

Figura 18 – (a) Classe Gerente. (b) Estrutura de Classe.

```

1 referência
public void AdicionarTarefa(List<Tarefas> tarefas)
{
    Tarefas tarefa = new Tarefas();
    Console.WriteLine("Digite o Nome da Tarefa:");
    tarefa.Nome = Console.ReadLine();
    Console.WriteLine("Digite o Prazo da Tarefa:");
    tarefa.Prazo = int.Parse(Console.ReadLine());
    Console.WriteLine("Digite a Prioridade da Tarefa:");
    tarefa.Prioridade = Console.ReadLine();
    Console.WriteLine("Digite a Etapa da Tarefa:");
    tarefa.Etapa = int.Parse(Console.ReadLine());
    Console.WriteLine("Digite o Número de Etapas da Tarefa:");
    tarefa.N_Etapas = int.Parse(Console.ReadLine());
    tarefas.Add(tarefa);
    Console.WriteLine("Tarefa incluída com sucesso.");
}

1 referência
public void PesquisarTarefa(List<Tarefas> tarefas)
{
    Console.WriteLine("Digite o Nome da Tarefa a ser pesquisada:");
    string nome = Console.ReadLine();
    Tarefas tarefa = tarefas.FirstOrDefault(t => t.Nome == nome);
    if (tarefa != null)
    {
        Console.WriteLine($"Nome: {tarefa.Nome}, Prazo: {tarefa.Prazo}, Prioridade: {tarefa.Prioridade},");
    }
    else
    {
        Console.WriteLine("Tarefa não encontrada.");
    }
}

< referências
public class Gerente : Usuario
{
    0 referências
    public string Status { get; set; }

    // Métodos CRUD para Funcionário
    1 referência
    public void AdicionarFuncionario(List<Funcionario> funcionarios)
    {
        Funcionario funcionario = new Funcionario();
        Console.WriteLine("Digite o ID do Funcionário:");
        funcionario.ID = int.Parse(Console.ReadLine());
        Console.WriteLine("Digite o Nome do Funcionário:");
        funcionario.Nome = Console.ReadLine();
        Console.WriteLine("Digite o Contato do Funcionário:");
        funcionario.Contato = double.Parse(Console.ReadLine());
        Console.WriteLine("Digite o CPF do Funcionário:");
        funcionario.CPF = int.Parse(Console.ReadLine());
        Console.WriteLine("Digite o Status do Funcionário:");
        funcionario.Status = Console.ReadLine();
        funcionarios.Add(funcionario);
        Console.WriteLine("Funcionário incluído com sucesso.");
    }

    1 referência
    public void PesquisarFuncionario(List<Funcionario> funcionarios)
    {
        Console.WriteLine("Digite o ID do Funcionário a ser pesquisado:");
        int id = int.Parse(Console.ReadLine());
    }
}

```

Fonte: Autor (2024).

Na imagem acima é possível visualizar os métodos para pesquisar e adicionar tarefas, na onde o gerente é capaz de manusear os dados dessa classe. Ele também é capaz de manusear das outras classes, utilizando códigos semelhante ao demonstrado acima.

Na figura 19 abaixo mostra a classe usuário, com seus atributos que são herdados pelas demais classes de usuários, como gerente e o funcionário.

Figura 19 – Classe Usuário.

```
namespace crudreal
{
    4 referências
    public class Usuario
    {
        6 referências
        public int ID { get; set; }
        3 referências
        public string Nome { get; set; }
        3 referências
        public double Contato { get; set; }
        3 referências
        public int CPF { get; set; }

        1 referência
        public virtual void Visualizar()
        {
            Console.WriteLine($"ID: {ID}, Nome: {Nome}, Contato: {Contato}, CPF: {CPF}");
        }

        0 referências
        public virtual Usuario Pesquisar(List<Usuario> usuarios, int id)
        {
            return usuarios.FirstOrDefault(u => u.ID == id);
        }
    }
}
```

Fonte: Autor (2024).

Já na figura 20 abaixo mostra os atributos de uma classe “não usuária”, que é a classe produção, que é mais como um “Módulo” disponível no site que poderá ser acessado e manipulado pelos usuários.

Figura 20 – Módulo do sistema

```
namespace crudreal
{
    11 referências
    public class Producao
    {
        5 referências
        public int ID { get; set; }
        3 referências
        public string NomeProduto { get; set; }
        3 referências
        public int DataInicio { get; set; }
        3 referências
        public int Prazo { get; set; }
        3 referências
        public int Qtd { get; set; }
        3 referências
        public int DataFim { get; set; }
        3 referências
        public string Prioridade { get; set; }
        3 referências
        public int ID_Fornecedor { get; set; }
        3 referências
        public int Codigo_Cliente { get; set; }
        3 referências
        public int N_Etapas { get; set; }
    }
}
```

Fonte: Autor (2024).

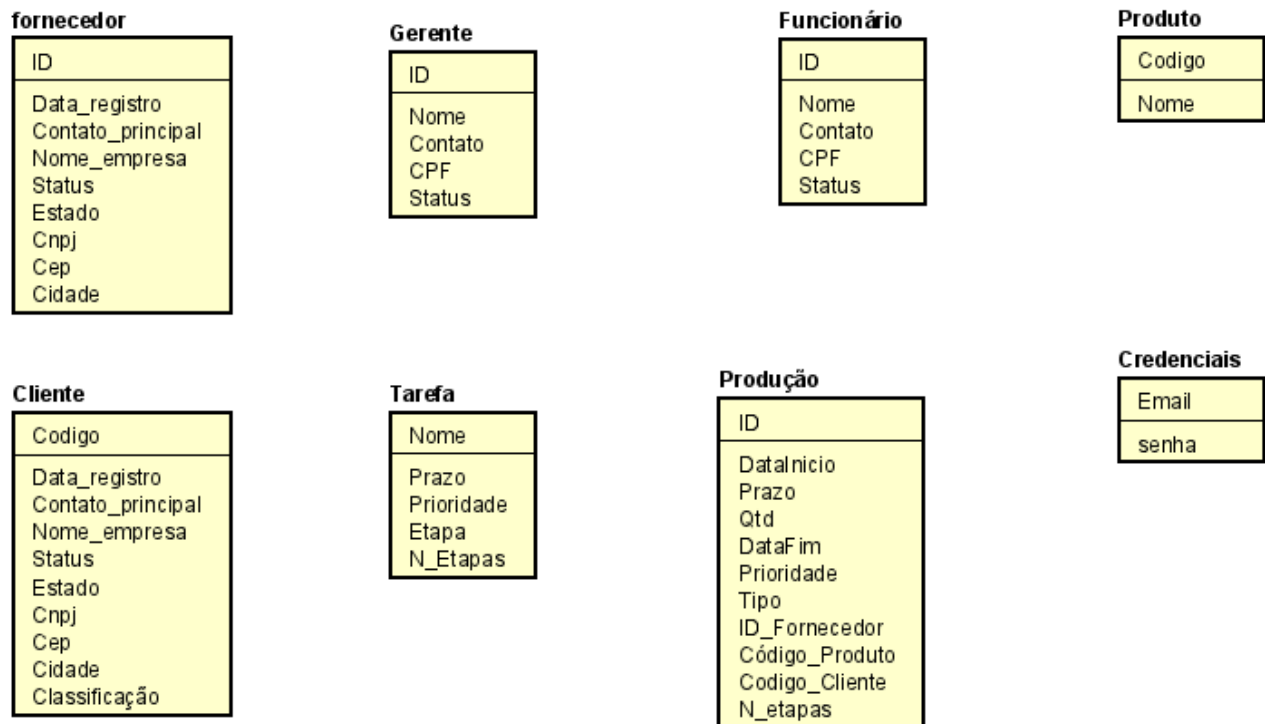
3.7. Banco de Dados

Levou-se em consideração que o projeto deve ter um banco de dados para armazenamento, gerenciamento e facilitar o manejo dos dados cadastrados pelo gerente do sistema.

Foi utilizado como base do banco de dados o diagrama de classe exibido mais acima, também foi feito o MER (Modelo Entidade Relacionamento) do sistema, com as entidades já possuindo atributos e chaves primárias.

A figura 21 ilustra o MER do projeto.

Figura 21 – MER.



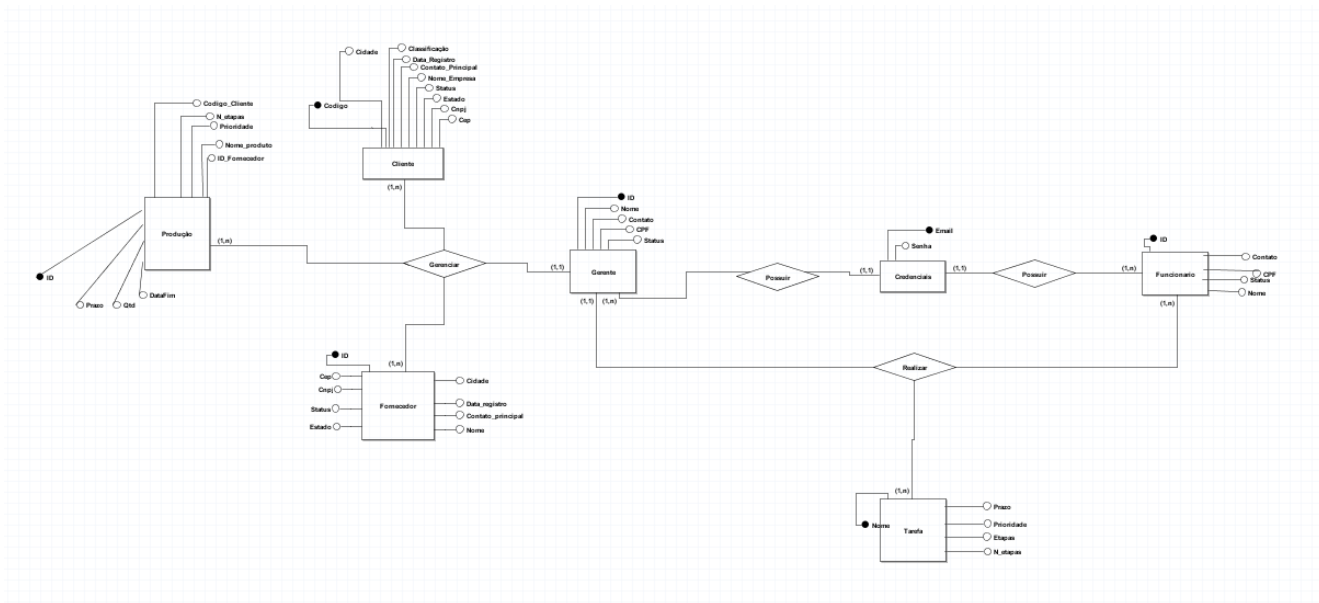
Fonte: Autor (2024).

As tabelas acima se encontram normalizadas até a terceira Norma Formal, ou seja, não possuem atributos multivalorados, não relação com chaves primárias ou relação com atributos fora chaves primárias.

Seguindo após a confirmação do MER existe o DER (Diagrama Entidade Relacionamento), referenciando como as entidades se relacionam, atributos definidos e cardinalidades entre entidades e associações.

A figura 22 ilustra o DER do projeto.

Figura 22 – DER.



Fonte: Autor (2024).

3.7.1. Estrutura em SQL

Tendo realizado os diagramas necessários para a concepção do banco de dados, a próxima etapa é criar o script para criação das tabelas do SQL.

Conforme ilustrado na figura 23 abaixo, a tabela “fornecedor” tem como atributo identificador o ID de fornecedor, categorizado como Primary Key. Também possui os outros dados necessários tanto para cadastro como para visualização das informações do fornecedor.

Figura 23 – Tabela SQL Fornecedor

```
CREATE TABLE FORNECEDOR(  
  ID INT PRIMARY KEY,  
  DATA_REGISTRO DATE,  
  CONTATO_PRINCIPAL VARCHAR(100),  
  NOME_EMPRESA VARCHAR (100),  
  STATUS VARCHAR (50),  
  ESTADO CHAR(2),  
  CNPJ VARCHAR(20),  
  CEP VARCHAR(20),  
  CIDADE VARCHAR(100)  
);
```

Fonte: Autor (2024).

Conforme ilustrado na figura 24 abaixo, as tabelas “gerente” e “funcionário” tem como atributo identificador o ID de empregado, categorizado como Primary Key. Também possui os outros dados informando as características do gerente/funcionário, tal como “status” que informa se o usuário é gerente ou funcionário.

Figura 24 – Tabela SQL Funcionário/Gerente

```
CREATE TABLE GERENTE(  
  ID INT PRIMARY KEY,  
  NOME VARCHAR(20),  
  CONTATO VARCHAR(20),  
  CPF VARCHAR(11),  
  STATUS VARCHAR(10)  
);  
  
CREATE TABLE FUNCIONARIO (  
  ID INT PRIMARY KEY,  
  NOME VARCHAR(20),  
  CONTATO VARCHAR(20),  
  CPF VARCHAR(11),  
  STATUS VARCHAR(10)  
);
```

Fonte: Autor (2024).

Conforme ilustrado na figura 25 abaixo, a tabela “Cliente” tem como atributo identificador o ID de cliente, categorizado como Primary Key. Também possui os outros dados necessários tanto para cadastro como para visualização das informações do cliente.

Figura 25 – Tabela SQL Cliente.

```
CREATE TABLE CLIENTE (  
  CODIGO INT PRIMARY KEY,  
  DATA_REGISTRO DATE,  
  CONTATO_PRINCIPAL VARCHAR(50),  
  NOME_EMPRESA VARCHAR (50),  
  STATUS VARCHAR(10),  
  ESTADO VARCHAR(2),  
  CNPJ VARCHAR (20),  
  CEP VARCHAR (20),  
  CIDADE VARCHAR (20),  
  CLASSIFICAÇÃO VARCHAR (20)  
);
```

Fonte: Autor (2024).

Conforme ilustrado na figura 26 abaixo, a tabela “Tarefa” tem como atributo identificador o nome da tarefa, categorizado como Primary Key. Também possui os outros dados necessários tanto para cadastro como para visualização das tarefas.

Figura 26 – Tabela SQL Tarefa.

```
CREATE TABLE TAREFAS(  
  NOME VARCHAR(20),  
  PRAZO DATE,  
  PRIORIDADE VARCHAR(20),  
  ETAPA VARCHAR (20),  
  N_ETAPAS INT  
);
```

Fonte: Autor (2024).

Conforme ilustrado na figura 27 abaixo, a tabela “Produção” tem como atributo identificador o ID da produção, categorizado como Primary Key. Também possui os outros dados necessários tanto para cadastro como para visualização das produções. Esta tabela também possui duas Foreign Key, relacionando-a com as tabelas “cliente” e “fornecedor”.

Figura 27 – Tabela SQL Produção.

```
CREATE TABLE PRODUÇÃO(  
  ID INT PRIMARY KEY,  
  DATAINICIO DATE,  
  PRAZO DATE,  
  QTD INT,  
  DATAFIM DATE,  
  PREIORIDADE VARCHAR (10),  
  NOMEPRODUTO VARCHAR (100),  
  ID_FORNECEDOR INT,  
  CODIGO_CLIENTE INT,  
  N_ETAPAS INT,  
  FOREIGN KEY (ID_FORNECEDOR)  
  REFERENCES FORNECEDOR(ID),  
  FOREIGN KEY (CODIGO_CLIENTE)  
  REFERENCES CLIENTE (CODIGO)  
);
```

Fonte: Autor (2024).

3.7.2. Inserção de Valores

A figura 28 abaixo ilustra como foi utilizada a função INSERT para inserir dados dentro das tabelas criadas.

Dados inseridos são para demonstração de como o sistema do banco de dados deve receber os dados e inseri-los nas tabelas.

Figura 28 – Função INSERT.

```

INSERT INTO FORNECEDOR (ID, DATA_REGISTRO, CONTATO_PRINCIPAL, NOME_EMPRESA, STATUS, ESTADO, CNPJ, CEP, CIDADE)
VALUES (90567, '2020-04-01', 'PIMTESTE@GMAIL.COM', 'NOME', 'ATIVO', 'SP', '11223344556677', '3655412', 'SJC');

INSERT INTO GERENTE (ID, NOME, CONTATO, CPF, STATUS)
VALUES (59781, 'NOME', 'GERENTE@PIM.COM', '12547859654', 'ATIVO');

INSERT INTO CLIENTE (CODIGO, DATA_REGISTRO, CONTATO_PRINCIPAL, NOME_EMPRESA, STATUS, ESTADO, CNPJ, CEP, CIDADE, CLASSIFICAÇÃO)
VALUES (76218, '2020-04-01', 'EMAILCLIENTE@TESTE.COM', 'NOME', 'ATIVO', 'SP', '224433117766', '3655412', 'SJC', 'A');

INSERT INTO PRODUÇÃO (ID, DATAINICIO, PRAZO, QTD, DATAFIM, PRIORIDADE, NOMEPRODUTO, ID_FORNECEDOR, CODIGO_CLIENTE, N_ETAPAS)
VALUES (77582, '2024-01-17', '2024-05-08', 600, '2024-05-08', 'ALTA', 'MAÇÃ', 90567, 76218, 7);

INSERT INTO CREDENCIAIS (EMAIL, SENHA)
VALUES ('ADMUSUARIO@TESTE.COM', 'SENHA1234');

INSERT INTO TAREFAS (NOME, PRAZO, PRIORIDADE, ETAPA, N_ETAPAS)
VALUES ('NOME', '2024-05-08', 'ALTA', 7, 7);

INSERT INTO FUNCIONARIO (ID, NOME, CONTATO, CPF, STATUS)
VALUES (25568, 'NOME', 'FUNCIONARIO@PIM.COM', '42718119730', 'ATIVO');

```

Fonte: Autor (2024).

3.7.3. Seleção de Tabelas

Foi utilizado a função SELECT FROM para exibir e selecionar os dados inseridos em uma tabela.

A figura 29 ilustra a função SELECT FROM na tabela “Produção”.

Figura 29 – SELECT FROM.

```

SELECT * FROM FORNECEDOR;

SELECT * FROM GERENTE;

SELECT * FROM FUNCIONARIO;

SELECT * FROM CLIENTE;

SELECT * FROM TAREFAS;

SELECT * FROM PRODUÇÃO;

SELECT * FROM CREDENCIAIS;

```

90 %
Resultados Mensagens

| | ID | DATAINICIO | PRAZO | QTD | DATAFIM | PRIORIDADE | NOMEPRODUTO | ID_FORNECEDOR | CODIGO_CLIENTE | N_ETAPAS |
|---|-------|------------|------------|-----|------------|------------|-------------|---------------|----------------|----------|
| 1 | 77582 | 2024-01-17 | 2024-05-08 | 600 | 2024-05-08 | ALTA | MAÇÃ | 90567 | 76218 | 7 |

Fonte: Autor (2024).

É possível visualizar no canto inferior da tela o resultado dos dados selecionados pela função.

3.7.4. Dicionário de Dados

Há também a necessidade de organizar um dicionário de dados, com o propósito de fornecer uma descrição detalhada sobre os atributos especificados em cada entidade.

Pode se dizer que o dicionário de dados é uma parte importante do desenvolvimento, tomando como base o conhecimento que a fazenda urbana Terras Mil se interessa pelo gerenciamento correto dos dados presentes no sistema em questão.

A tabela 2 detalha o dicionário de dados das entidades.

Tabela 2 – (a) fornecedor. (b) Gerente. (c) Funcionário. (d) Cliente. (e) Tarefas. (f) Produção.

| FORNECEDOR | | | |
|-------------------|--|---------------|---------|
| Campo | Descrição | Tipo de Dados | Tamanho |
| ID | Número de identificação de um fornecedor | INT | - |
| DATA_REGISTRO | Data em que o fornecedor foi registrado | DATE | - |
| CONTATO_PRINCIPAL | Contato principal do fornecedor | VARCHAR | 100 |
| NOME_EMPRESA | Nome da empresa do fornecedor | VARCHAR | 100 |
| STATUS | Status do fornecedor 'Ex: 'Ativo', 'Inativo' | VARCHAR | 50 |
| ESTADO | Estado em que o fornecedor se localiza | CHAR | 2 |
| CNPJ | Cnpj do fornecedor | VARCHAR | 20 |
| CEP | Cep do fornecedor | VARCHAR | 20 |
| CIDADE | Cidade em que o fornecedor se localiza | VARCHAR | 100 |

| GERENTE | | | |
|---------|---|---------------|---------|
| Campo | Descrição | Tipo de Dados | Tamanho |
| ID | Número de identificação do gerente | INT | - |
| NOME | Nome do fornecedor | VARCHAR | 20 |
| CONTATO | Contato principal do fornecedor | VARCHAR | 20 |
| CPF | Cpf do gerente | VARCHAR | 11 |
| STATUS | Status do gerente 'Ex: 'Ativo', 'Inativo' | VARCHAR | 10 |

| FUNCIONARIO | | | |
|-------------|---|---------------|---------|
| Campo | Descrição | Tipo de Dados | Tamanho |
| ID | Número de identificação do funcionario | INT | - |
| NOME | Nome do funcionario | VARCHAR | 20 |
| CONTATO | Contato principal do funcionario | VARCHAR | 20 |
| CPF | Cpf do funcionario | VARCHAR | 11 |
| STATUS | Status do gerente 'Ex: 'Ativo', 'Inativo' | VARCHAR | 10 |

| CLIENTE | | | |
|-------------------|---|---------------|---------|
| Campo | Descrição | Tipo de Dados | Tamanho |
| CODIGO | Número de identificação do cliente | INT | - |
| DATA_REGISTRO | Data em que o cliente foi registrado | DATE | - |
| CONTATO_PRINCIPAL | Contato principal do cliente | VARCHAR | 50 |
| NOME_EMPRESA | Nome da empresa do cliente | VARCHAR | 50 |
| STATUS | Status do cliente 'Ex: 'Ativo', 'Inativo' | VARCHAR | 10 |
| ESTADO | Estado em que o cliente se localiza | VARCHAR | 2 |
| CNPJ | Cnpj do cliente | VARCHAR | 20 |
| CEP | Cep em que o cliente se localiza | VARCHAR | 20 |
| CIDADE | Cidade em que o cliente se localiza | VARCHAR | 20 |
| CLASSIFICAÇÃO | Classificação do cliente e 'A', seu nível de prioridade | VARCHAR | 20 |

| TAREFAS | | | |
|------------|---|---------------|---------|
| Campo | Descrição | Tipo de Dados | Tamanho |
| NOME | Identifica nome das tarefas | VARCHAR | 20 |
| PRAZO | Prazo para ser entregue as tarefas | DATE | - |
| PRIORIDADE | Verificar se a tarefa e uma prioridade ou não | VARCHAR | 20 |
| ETAPA | Etapa da tarefa | VARCHAR | 20 |
| N_ETAPAS | Número da etapa ate a sua conclusão | INT | - |

| PRODUÇÃO | | | |
|----------------|---|---------------|---------|
| Campo | Descrição | Tipo de Dados | Tamanho |
| ID | Identificação do produto que está sendo produzido | INT | - |
| DATAINICIO | Data em que começou a produzir o item | DATE | - |
| PRAZO | Prazo para a produção entregar o item | DATE | - |
| QTD | Qtd de itens que estão sendo produzidos | INT | - |
| DATAFIM | Data que ira acabar a produção | DATE | - |
| PRIORIDADE | Prioridade de itens que devem ser produzidos | VARCHAR | 10 |
| NOMEPRODUTO | Nome do produto que esta sendo produzidos | VARCHAR | 100 |
| ID_FORNECEDOR | Id do fornecedor do item da produção | INT | - |
| CODIGO_CLIENTE | Id do cliente que comprou o item que esta sendo produzido | INT | - |
| N_ETAPAS | Número da etapa em que a produção se encontra | INT | - |

Fonte: Autor (2024).

3.8. Planilha de Testes

Para garantir o melhor funcionamento do sistema de gerenciamento, foi criado uma planilha com testes que a equipe de desenvolvimento julgou necessários, tendo tido como base melhorar a usabilidade da interface e checar erros presentes no código de algumas classes.

A tabela 3 ilustra os testes escolhidos para o software da fazenda urbana Terras Mil.

Tabela 3 – Planilha de Testes.

| Teste de Unidade | | | | | | |
|---|---------------------|-----------------------------|-------------|--------------------|--|---------------------------------|
| Requisito/Classe | Nível de Prioridade | Tipo de teste | Responsável | Resultado Esperado | | Comentários |
| Cliente | Alta | Unidade/Unitário | João | Sim ou Não | | # |
| Vendas | Alta | Unidade/Unitário | Matheus | Sim ou Não | | |
| Produção | Alta | Unidade/Unitário | Matheus | Sim ou Não | | |
| Fornecedor | Alta | Unidade/Unitário | João | Sim ou Não | | |
| Teste A/B | | | | | | |
| Telas | Nível de Prioridade | Tipo de teste | Responsável | Resultado Esperado | | Comentários |
| Cadastro Fornecedores | Média | Teste A/B | Luís | A > B ou B > A | | Mudança de posição de botão |
| Cadastro de Cliente | Média | Teste A/B | Lucas | A > B ou B > A | | Mudança de cor de barra de nav. |
| Vendas | Média | Teste A/B | Luís | A > B ou B > A | | Mudança de gráficos |
| Produção | Média | Teste A/B | Lucas | A > B ou B > A | | Mudança de exibição |
| Teste de Usabilidade | | Numero participantes | | | | |
| Sistema todo é submetido a teste qualitativo. | | 3 a 5 | | | | |

Fonte: Autor (2024).

Foi decidido utilizar o teste de unidade para checar incongruências e atividades suspeitas no código das classes, as classes escolhidas foram as julgadas mais complexas para o sistema.

O teste A/B foi escolhido por sua versatilidade em compara variáveis da interface, tendo como objetivo testar as diferentes alterações feitas no protótipo da interface.

O teste de usabilidade é realizado por todos os membros da equipe de desenvolvimento para verificação caso o sistema não tenha sido desenvolvido com a usabilidade esperada, ou seja, busca melhorar a usabilidade do sistema caso haja espaço de aprimoramento.

4. CONSIDERAÇÕES FINAIS

Após o desenvolvimento do projeto e documentação é possível dizer que o início do projeto foi sombrio e confuso, com o grupo sendo surpreendido com um tema tão abrangente como um sistema de gerenciamento de fazenda urbana, porém com o passar do tempo o tema começou a ser entendido pelos integrantes do grupo, abrindo caminho para o desenvolvimento do projeto.

Utilizando a metodologia Scrum, a equipe encontrou diversos problemas durante a construção do projeto, indo desde a identificação dos requisitos, pesquisas sobre o tema, até a própria construção da interface, porém estes problemas foram devidamente resolvidos através dos esforços colaborativos da equipe de desenvolvimento, que agiram em conjunto desde a projeção dos primeiros requisitos do sistema de gerenciamento da fazenda urbana Terras Mil.

Apesar da resiliência do grupo em frente dos desafios apresentados pelo tema e construção do tema, todo o projeto teve muitas mudanças ao longo dos encontros do grupo, o estado atual é um consenso entre os membros, porém não representa totalmente o que era esperado pelos membros. Foi necessário “cortar” do desenvolvimentos certas coisas que gostaríamos que estivessem no protótipo final, porém era inviável criar e adicionar no produto, seja por dificuldade ou por não haver necessidade de algo exótico, já que o sistema segue uma temática minimalista.

No fim, apesar das adversidades, o projeto conseguiu sair da imaginação e finalmente ser construído. Por mais que o sistema pudesse ter sido outra versão ideal pela visão dos desenvolvedores, a versão atual satisfaz a maior parte dos usuários que participaram dos testes de desenvolvimento. Também foi uma grande oportunidade de aprendizado para a utilização das ferramentas utilizadas na criação dos diagramas e protótipo de telas, sendo as ferramentas mais utilizadas pelo grupo o Figma, brModelo, Astah, SQL e Visual Studio. É de grande importância ter conhecimento dessas ferramentas, tendo em vista que a maior parte do mercado do trabalho as utiliza em suas operações corriqueiras.

REFERÊNCIAS

BLUESEEDS. **FAZENDAS URBANAS: a agricultura transforma a alimentação nas cidades**. Disponível em: <https://blueseeds.com.br>. Acesso em: 28 abr. 2024.

BROTO. **Guia prático de agricultura urbana: como cultivar alimentos em grandes centros**. Disponível em: <https://blog.broto.com.br/agricultura-urbana/>. Acesso em: 8 abr. 2024.

COODESH. **O que é CRUD?**. Disponível em: <https://coodesh.com/blog/dicionario/o-que-e-crud/>. Acesso em: 28 abr. 2024.

CREATELY. **O Guia Fácil de Diagramas de Implantação UML**. Disponível em: <https://creately.com/blog/pt/diagrama/tutorial-do-diagrama-de-implantacao/#:~:text=Um%20diagrama%20de%20implantação%20é,software%20físico%20de%20um%20sistema..> Acesso em: 10 abr. 2024.

GREENPEACE. **Cúpula da Amazônia: veja as propostas do Greenpeace Brasil**. Disponível em: https://www.greenpeace.org/brasil/blog/cupula-da-amazona-veja-as-propostas-do-greenpeace-brasil/?appeal=21057&utm_source=google&utm_medium=paid&utm_campaign=florestas&utm_content=aq_20230810_grants&utm_term=cop%2030&utm_campaign=&utm_source=adwords&utm_medium=ppc&hsa_acc=7235609613&hsa_cam=196645621. Acesso em: 6 mai. 2024.

IBM. **Diagramas de Seqüência**. Disponível em: <https://www.ibm.com/docs/pt-br/rsm/7.5.0?topic=uml-sequence-diagrams>. Acesso em: 28 abr. 2024.

NAÇÕES UNIDAS BRASIL. **População mundial deve chegar a 9,7 bilhões de pessoas em 2050, diz relatório da ONU**. Disponível em: <https://brasil.un.org/pt-br/83427-população-mundial-deve-chegar-97-bilhões-de-pessoas-em-2050-diz-relatório-da-onu>. Acesso em: 9 abr. 2024.

SANEAMENTO EM PAUTA. **Fazendas urbanas: conheça o conceito e descubra seus benefícios.** Disponível em: <https://blog.brkambiental.com.br/fazendas-urbanas/#:~:text=Fazendas%20urbanas%20são%20espaços%20concebidos,estufas%20agrícolas%20anexas%20às%20edificações..> Acesso em: 15 abr. 2024.

FICHA DE CONTROLE DO PIM

Ano: 2024

Período: 2º/3º

Coordenador: Prof Roberto Cordeiro Waltz

Tema (Identificação da startup): Sistema de gerenciamento da fazenda urbana Terras Mil.

Alunos

| RA | Nome | E-mail | Curso | Visto do aluno |
|----------|---------------------------------------|-------------------------------|-------------------|----------------|
| G81BH5C | Carlos Henrique Duarte da Silva | carlos.silva818@aluno.unip.br | CST em ADS | Lucas |
| R0167J3 | João Gabriel Alves dos Santos | joao.santos811@aluno.unip.br | CST em ADS | Lucas |
| R009JD0 | Lucas Navimar de Toledo | lucas.toledo12@aluno.unip.br | CST em ADS | Lucas |
| R024768 | Luis Henrique Vieira Barros | luis.barros15@aluno.unip.br | CST em ADS | Lucas |
| R026272 | Matheus Guilherme Pacheco Courbassier | matheus.cour@aluno.unip.br | CST em ADS | Lucas |
| G86JCF-1 | Ruan Pablo Galdino Dias Bento | ruan.bento@aluno.unip.br | CST em ADS | Lucas |

Registros

| Data do encontro | Observações |
|-------------------|---|
| 11/03/2024 | Reunião para estudos sobre fazendas urbanas |
| 20/03/2024 | Especificação de requisitos |
| 01/04/2024 | Separação de requisitos funcionais e não funcionais |
| 11/04/2024 | Começo dos diagramas UML |
| 16/04/2024 | Reunião sobre o MER |
| 23/04/2024 | Revisão dos Diagramas UML |
| 25/04/2024 | Início da Documentação |
| 13/04/2024 | Criação do Crud |
| 22/04/2024 | Revisão de todas as partes do projeto |