FATEC – OURINHOS

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

ENGENHARIA DE SOFTWARE

**AgroPlanner**

**Versão1.0**

**2024**

**FATEC – OURINHOS**

**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**ENGENHARIA DE SOFTWARE**

**AgroPlanner**

**Versão1.0**

**ACADÊMICOS**

Joao Vinicius Latczuk

Projeto Desenvolvido para as disciplinas de Engenharia de Software da Faculdade de Tecnologia de Ourinhos.

**2024**

**Gerenciamento de Configuração**

|  |  |  |
| --- | --- | --- |
| Data | Atividade | Responsável |
| 27/09/24 | Inserção de informações sobre o projeto | Joao Vinicius |
| 27/09/24 | Proposito do sistema / início dos requisitos | Joao Vinicius |
| 03/10/24 | Requisitos funcionais e não funcionais | Joao Vinicius |
| 06/10/24 | Diagrama de pacotes e ou subsistemas | João Vinicius |
| 06/10/24 | Casos de Uso Cadastrais do Subsistema | João Vinicius |
| 23/04/25 | Alterações e implementação dos Protótipos | Joao Vinicius |
| 01/05/25 | Implementação | João Vinicius |
| 02/05/25 | Correção de algumas funcionalidades | João Vinicius |
| 02/05/25 | Revisão de diagramas | João Vinicius |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**SUMÁRIO**

[Plano de Projeto 5](#_Toc1384784293)

[1. Introdução 5](#_Toc1287298321)

[2. Escopo do Projeto 5](#_Toc1316051972)

[3. Descrição Preliminar do Sistema 5](#_Toc1804375633)

[4. Modelo de Ciclo de Vida 6](#_Toc1800378413)

[5 Estrutura da Equipe do Projeto 8](#_Toc692241267)

[Documento de Requisitos 8](#_Toc809259996)

[1. Introdução 8](#_Toc1351818648)

[2. Descrição do Propósito do Sistema 9](#_Toc39579117)

[3. Descrição do Minimundo 9](#_Toc1042552456)

[4. Requisitos de Usuário 10](#_Toc770841209)

[Documento de Especificação de Requisitos 13](#_Toc1700408190)

[1. Introdução 13](#_Toc500076069)

[2. Requisitos do sistema 13](#_Toc66752367)

[3. Identificação de Subsistemas 15](#_Toc536055546)

[4. Modelo de Casos de Uso 16](#_Toc90509281)

[4.1 Subsistema AgroPlanner 21](#_Toc1102379413)

[5. Prototipação de Telas 21](#_Toc2085938152)

[5.1 Cadastrar Usuário 22](#_Toc359071540)

[5.2 Login Usuário 23](#_Toc1125610773)

[5.3 Página Inicial 24](#_Toc1370818533)

[5.4 Página do calendário 25](#_Toc1504229452)

[5.5 Fórum 26](#_Toc512605290)

[5.6 Página do conteúdo. 27](#_Toc447486705)

[6. Regras de Negócio 27](#_Toc1597837890)

[7. Implementação 28](#_Toc1608543945)

[7.1 Estrutura de Pastas e Arquivos 28](#_Toc1889906160)

[7.2 Tecnologias Utilizadas 29](#_Toc1736533625)

[7.3 Códigos principais 30](#_Toc997851230)

[7.4 Banco de Dados 30](#_Toc270345044)

[31](#_Toc536417395)

[7.6 Telas de Login 31](#_Toc1557049647)

[7.7 Telas de Início 32](#_Toc434549909)

[7.8 Tela de Adicionar de Planta 32](#_Toc1744789734)

[7.9 Tela de Publicação da Planta 33](#_Toc1562137976)

[8. Conclusão 33](#_Toc689770130)

Plano de Projeto

1. Introdução

Este documento apresenta uma versão resumida do plano de projeto da ferramenta AgroPlanner. Este documento está organizado da seguinte forma: a seção 2 apresenta a definição do escopo do projeto; a seção 3 apresenta a descrição preliminar do sistema; a seção 4 apresenta o modelo de ciclo de vida escolhido para construção do sistema, justificando sua escolha e apresentando as principais atividades que serão realizadas em cada etapa, seus insumos, produtos e papéis que irão realizá-las; a seção 5 apresenta a estrutura da equipe de projeto; finalmente, a seção 6 apresenta medidas definidas para cada atividade do projeto.

2. Escopo do Projeto

A seguir são apresentadas as informações básicas de definição do escopo do projeto:

Projeto: AgroPlanner

Justificativa: O produtor "caseiro" hoje em dia não consegue ter uma boa gestão do período de plantio, ou as vezes não possui experiência, e a AgroPlanner vai ajudar ele neste quesito.

Produto: O produto irá indicar ao produtor como e o que ele precisa para realizar o plantio e obter sucesso, além de poder criar seu próprio plano de plantio caso já saiba como fazer, podendo divulgar a outras pessoas.

3. Descrição Preliminar do Sistema

O sistema é um aplicativo desktop destinado a ajudar usuários de diferentes níveis de experiência a plantar e cuidar de suas plantas. O aplicativo fornecerá funcionalidades que facilitam o gerenciamento de cuidados, como lembretes para regar, fertilizar e podar, além de oferecer informações sobre as necessidades específicas de cada tipo de planta. Com uma interface intuitiva e amigável, o site também permitirá que os usuários compartilhem experiências em uma comunidade dedicada, acessem dicas e tutoriais, e utilizem da comunidade para identificação de plantas.

A partir da descrição foram levantados os seguintes requisitos preliminares:

* Cadastro de Usuário:

Permitir que os usuários criem uma conta para personalizar sua experiência.

* Cadastro de Plantas:

Funcionalidade para adicionar diferentes tipos de plantas, incluindo nome, tipo e características.

* Lembretes Personalizados:

Sistema de notificações para alertar os usuários sobre as necessidades de cada planta (regar, fertilizar, etc.).

* Calendário de Cuidados:

Exibir um calendário com as tarefas programadas para cada planta.

* Dicas e Tutoriais:

Acesso a um banco de dados com artigos sobre cuidados com plantas.

* Comunidade:

Seção para interação entre usuários, onde podem compartilhar dicas e tirar dúvidas.

* Versão Gratuita e Premium:

Estrutura de monetização que oferece uma versão básica gratuita e uma versão premium com recursos adicionais.

4. Modelo de Ciclo de Vida

Para o desenvolvimento do aplicativo móvel de cuidados com plantas, adotaremos o Modelo Ágil (Scrum). Este modelo é escolhido devido à sua flexibilidade, capacidade de adaptação às mudanças e foco na entrega incremental de funcionalidades.

Justificativa:

Flexibilidade e Adaptação: O desenvolvimento ágil permite ajustar os requisitos conforme feedback dos usuários e stakeholders, essencial para um aplicativo que atende a um público diverso.

Entrega Contínua: Através de sprints curtos, podemos lançar novas funcionalidades e correções rapidamente, melhorando a experiência do usuário de forma contínua.

Colaboração: A interação constante entre a equipe de desenvolvimento e os usuários garante que o produto final atenda às reais necessidades dos usuários.

Atividades

Planejamento do Sprint: Definição das histórias de usuário a serem implementadas no próximo sprint, priorizando as funcionalidades mais críticas, como cadastro de usuários e lembretes personalizados.

Desenvolvimento: Implementação das funcionalidades conforme as histórias de usuário, utilizando práticas de programação ágil e testes contínuos.

Reuniões Diárias: Encontros diários para acompanhamento do progresso, identificação de obstáculos e realinhamento das atividades da equipe.

Revisão do Sprint: Apresentação das funcionalidades desenvolvidas para a equipe e stakeholders, coletando feedback para melhorias.

Retrospectiva: Reflexão sobre o que funcionou bem e o que pode ser melhorado no processo, promovendo uma melhoria contínua.

Insumos

Backlog do Produto: Lista priorizada de requisitos e funcionalidades.

Histórias de Usuário: Descrições das necessidades dos usuários que guiarão o desenvolvimento.

Produtos

Incrementos do Produto: Funcionalidades implementadas e testadas a cada sprint, como o cadastro de plantas e o sistema de lembretes.

Documentação: Documentos de especificação, manuais de usuário e guias de implementação.

Papéis

Product Owner: Responsável por definir e priorizar o backlog do produto, garantindo que as funcionalidades atendam às necessidades dos usuários.

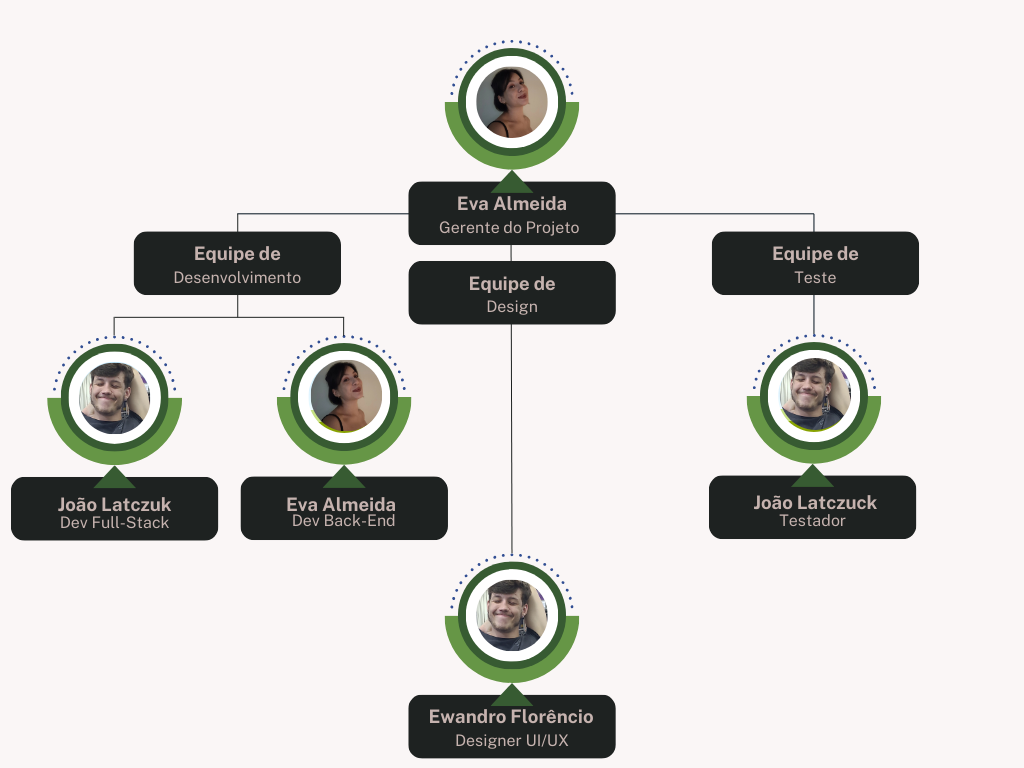
Scrum Master: Facilita o processo ágil, ajuda a equipe a remover impedimentos e garante que as práticas ágeis sejam seguidas.

Equipe de Desenvolvimento: Composta por desenvolvedores, designers e testadores, que trabalham juntos para entregar as funcionalidades planejadas.

Conclusão

O uso do modelo ágil (Scrum) permitirá um desenvolvimento eficiente e centrado no usuário, facilitando a entrega de um aplicativo robusto e adaptável que atenda às necessidades dos amantes de plantas, independentemente de sua experiência. Com foco na colaboração, iteração e feedback contínuo, o aplicativo pode evoluir rapidamente e se tornar uma ferramenta indispensável para o cuidado de plantas.

5 Estrutura da Equipe do Projeto

**Figura 1** – Estrutura da Equipe do Projeto – **Fonte:** Autor

Papéis e Responsabilidades

1. **Gerente de Projeto**: Coordena o projeto, gerencia a equipe e comunica-se com stakeholders.
2. Equipe de Desenvolvimento:
   * **Dev Frontend**: Desenvolve a interface do usuário e a experiência do usuário.
   * **Dev Backend**: Cria a lógica do servidor, banco de dados e integrações.
3. **Designer UI/UX**: Projeta a interface do aplicativo, focando na usabilidade.
4. Equipe de Teste:
   * **Testador**: Realiza testes para garantir a qualidade do aplicativo.

Documento de Requisitos

1. Introdução

Este documento apresenta os requisitos de usuário da ferramenta AgroPlanner e está organizado da seguinte forma: a seção 2 contém uma descrição do propósito do sistema; a seção 3 contém uma descrição do minimundo apresentando o problema; e a seção 4 apresenta a lista de requisitos de usuário levantados junto ao cliente.

2. Descrição do Propósito do Sistema

O sistema tem o propósito de fornecer aos usuários um calendário detalhado de plantio, personalizado de acordo com sua localização, estação do ano e tipo de cultivo. O aplicativo oferece também dicas práticas, notificações para lembrar de tarefas essenciais e orientações passo a passo para garantir um cultivo saldável e produtivo.

3. Descrição do Minimundo

O minimundo do AgroPlanner representa um ambiente digital onde usuários podem criar e gerenciar seus próprios jardins virtuais. Cada planta no sistema possui um perfil detalhado, contendo informações sobre suas necessidades específicas, como quantidade de água, luz solar e tipo de solo. Os usuários podem criar planos de cultivo personalizados, adicionando as plantas que desejam cultivar e configurando lembretes para as tarefas de cuidado, como regar, adubar e podar.

O sistema oferece uma comunidade onde os usuários podem compartilhar suas experiências, através de uma interface intuitiva, o aplicativo permite que os usuários possam trocar dicas e tirar dúvidas. Além disso, a própria comunidade contribuirá na identificação de plantas que irá permitir aos próprios usuários dela descobrir qual espécie é uma planta específica, bastando tirar uma foto dela.

Sendo um ecossistema digital que simula o processo de cultivo, oferecendo ferramentas e informações para auxiliar os usuários em todas as etapas, desde o planejamento até a colheita.

Objetivo:

* Facilitar o planejamento e gerenciamento de cultivos para produtores de todos os níveis de experiência.
* Oferecer informações personalizadas sobre as necessidades de cada planta.
* Criar uma comunidade online para troca de conhecimentos e experiências.
* Auxiliar na tomada de decisões relacionadas ao cultivo, como escolha de plantas, data de plantio e técnicas de cultivo.
* Usuários:
* Produtores rurais: desde pequenos agricultores familiares até produtores comerciais.
* Amadores: pessoas que desejam cultivar plantas em casa, como em hortas ou jardins.
* Estudantes e pesquisadores: interessados em aprender mais sobre agricultura e jardinagem.

Entidades:

* Plantas: Cada planta possui um perfil detalhado com informações como nome científico, nome popular, necessidades de água, luz, temperatura, tipo de solo, pragas e doenças comuns.
* Usuários: Cada usuário possui um perfil com informações pessoais, histórico de cultivo, plantas favoritas e configurações personalizadas.
* Planos de cultivo: Um plano de cultivo é uma lista de plantas que o usuário deseja cultivar, com informações sobre a data de plantio, espaçamento entre as plantas e outras informações relevantes.
* Lembretes: O sistema envia lembretes para o usuário sobre as tarefas de cuidado das plantas, como regar, adubar e podar.
* Comunidade: A comunidade é um espaço onde os usuários podem compartilhar fotos, dicas, dúvidas e experiências relacionadas ao cultivo.

Ações:

* Criar um plano de cultivo: O usuário seleciona as plantas que deseja cultivar e define as configurações do plantio.
* Adicionar plantas: O usuário pode adicionar novas plantas ao seu plano de cultivo.
* Configurar lembretes: O usuário define a frequência e o tipo de lembrete para cada planta.
* Compartilhar experiências: O usuário pode compartilhar fotos e informações sobre suas plantas na comunidade.
* Buscar informações: O usuário pode buscar informações sobre plantas específicas, técnicas de cultivo e outras questões relacionadas à jardinagem.

Informações:

* Dados das plantas: Nome científico, nome popular, família, origem, características, necessidades de cultivo, pragas e doenças.
* Dados do usuário: Nome, e-mail, senha, preferências de cultivo, histórico de plantas cultivadas.
* Dados dos planos de cultivo: Lista de plantas, data de plantio, espaçamento entre as plantas, estágio de desenvolvimento.
* Dados da comunidade: Posts dos usuários, comentários, curtidas.

4. Requisitos de Usuário

Tomando por base o contexto do sistema, foram identificados os seguintes requisitos de usuário:

Requisitos Funcionais

|  |  |  |  |
| --- | --- | --- | --- |
| Identificador | Descrição | Prioridade | Depende de |
| RF1 | Permitir criação de contas de usuário | Alta | - |
| RF2 | Login de usuários existentes para acesso personalizado. | Alta | RF1 |
| RF3 | Permitir atualização de dados pessoais no perfil do usuário. | Média | RF1 |
| RF4 | Usuário pode adicionar diferentes plantas e suas informações. | Alta | RF2 |
| RF5 | Permitir ao usuário capturar informações detalhadas sobre as plantas. | Média | RF4 |
| RF6 | Edição e remoção de plantas cadastradas | Alta | RF4 |
| RF7 | Lembretes para tarefas como regar e fertilizar. | Alta | - |
| RF8 | Notificações para fertilização, poda e outras tarefas. | Média | RF7 |
| RF9 | Usuário pode marcar tarefas como concluídas no calendário. | Média | RF8 |
| RF10 | Permitir busca e filtragem de conteúdo por categoria ou planta. | Média | RF10 |

**Prioridade Alta:** Funcionalidade essencial para o funcionamento básico do sistema.

**Prioridade Média:** Funcionalidade importante, mas que pode ser adicionada após as funcionalidades essenciais.

**Requisitos Não Funcionais**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Identificador | Descrição | Categoria | Escopo | Prioridade | Depende de |
| RNF1 | O usuário deve ser capaz de alterar sua senha sem a intervenção de um administrador. | Usabilidade | Aplicação inteira | Alta | - |
| RNF2 | O controle deve ser a partir da entrada de seu login e senha. | Usabilidade | Aplicação inteira | Alta | RNF1 |
| RNF3 | O sistema deve ser fácil de manter e atualizar. | Manutenibilidade | Aplicação inteira | Alta | - |
| RNF4 | O sistema deve proteger as informações do usuário e do evento por meio de recursos de segurança qualificados. | Segurança | Aplicação inteira | Alta | - |
| RNF5 | O sistema deve ser capaz de lidar com muitos usuários e transações simultâneas, especialmente durante eventos de alta demanda. | Desempenho | Aplicação inteira | Alta | - |
| RNF6 | O sistema deve estar disponível 24 horas por dia, 7 dias por semana, com tempos de inatividade mínimos. | Disponibilidade | Aplicação inteira | Alta | - |
| RNF7 | Interface fácil de navegar para todos os níveis de experiência. | Usabilidade | - | Alta | - |
| RNF8 | Garantir acesso autorizado às informações dos usuários. | Usabilidade | Aplicação inteira | Alta | - |

Documento de Especificação de Requisitos

1. Introdução

Este documento apresenta a especificação dos requisitos da ferramenta AgroPlanner. A atividade de análise de requisitos foi conduzida aplicando-se técnicas de modelagem de casos de uso, modelagem de classes e modelagem de comportamento dinâmico do sistema. Os modelos apresentados foram elaborados usando a linguagem UML. Este documento está organizado da seguinte forma: a seção 2 apresenta os subsistemas identificados, mostrando suas dependências na forma de um diagrama de pacotes; a seção 3 apresenta o modelo de casos de uso, incluindo descrições de atores, os diagramas de casos de uso e descrições de casos de uso; a seção 4 apresenta o modelo conceitual estrutural do sistema, na forma de diagramas de classes; a seção 5 apresenta o modelo comportamental dinâmico do sistema, na forma de diagramas de estado; finalmente, a seção 6 apresenta o glossário do projeto, contendo as definições das classes identificadas.

2. Requisitos do sistema

Tomando por base o contexto do sistema, foram identificados os seguintes requisitos:

Os requisitos funcionais são requisitos que expressam funções ou serviços que um software deve ou pode ser capaz de executar ou fornecer. As funções ou serviços são, em geral, processos que utilizam entradas para produzir saídas.

Requisitos Funcionais

Abaixo estão os requisitos funcionais do sistema, ou seja, as funções que o sistema deve exercer.

Observações: Dados marcados com \* (asterisco) serão considerados campos de preenchimento obrigatório.

**RF01 – Cadastrar Usuário**

1.1. O sistema deve permitir incluir, consultar, excluir e **alterar** usuários.

1.2. O sistema deve permitir a inclusão de um usuário, sendo necessário os seguintes dados e tipo:

1.2.1. Usuário Cliente:

Acesso somente à parte de visualização das plantas disponíveis e de gerenciamento das atividades.

Dados obrigatórios: Nome\*, senha\*, CPF\*, e-mail\*.

1.2.1.1. Cada usuário cliente poderá ter acesso a somente uma conta.

1.2.1.2. O cadastro será feito pelo próprio usuário.

1.2.1.3. O sistema não deve permitir o cadastro de mais de um login e CPF já cadastrados.

1.3. Os usuários poderão alterar seus dados cadastrais quando desejarem, exceto seu login e CPF.

1.4. Para exclusão, deve-se efetuar o login e, então, solicitar a exclusão.

1.5. O sistema deve permitir a consulta de usuários cadastrados, exibindo todos os dados por nome em ordem alfabética.

**RF02 – Manter Login de Usuários**

2.1. O sistema deve permitir que usuários existentes façam login usando seu login e senha.

2.2. O sistema deve validar as credenciais do usuário para garantir o acesso seguro.

2.3. O sistema deve permitir a recuperação de senha por meio de envio de e-mail com instruções.

2.4. Após a validação do login, o sistema deve redirecionar o usuário para a tela inicial personalizada.

**RF03 – Atualização de Dados Pessoais**

3.1. O sistema deve permitir que o usuário atualize suas informações pessoais no perfil.

3.2. Os dados que podem ser atualizados incluem: Nome, e-mail e senha.

3.3. O sistema não deve permitir que o usuário altere seu login ou CPF.

**RF04 – Adicionar Plantas**

4.1. O usuário deve poder adicionar diferentes plantas ao seu perfil.

4.2. Para adicionar uma planta, o usuário deve fornecer os seguintes dados: Nome da planta\*, Espécie\*, Características\*, Cuidados necessários\*.

4.3. O sistema deve validar os dados inseridos antes de permitir a adição.

**RF05 – Captura de Informações sobre Plantas**

5.1. O sistema deve permitir ao usuário capturar informações detalhadas sobre as plantas.

5.2. As informações devem incluir: Espécie, características e cuidados.

**RF06 – Edição e Remoção de Plantas**

6.1. O usuário deve poder editar as informações das plantas cadastradas.

6.2. O sistema deve permitir a remoção de plantas, com confirmação da ação.

**RF07 – Lembretes para Tarefas**

7.1. O sistema deve permitir que o usuário configure lembretes para tarefas como regar e fertilizar.

7.2. Os lembretes devem ser enviados por notificação.

**RF08 – Exibir Notificações**

8.1. O sistema deve exibir notificações para fertilização, poda e outras tarefas.

8.2. As notificações devem ser personalizáveis pelo usuário.

**RF09 – Exibir Visualização de Atividades**

9.1. O sistema deve permitir ao usuário visualizar suas tarefas agendadas (regar, adubar, podar) de forma organizada no calendário.

9.2. O calendário deve ser atualizado automaticamente conforme o usuário adiciona, edita ou remove lembretes.

**RF10 – Busca e Filtro de Conteúdo**

10.1. O sistema deve permitir filtro por tipo de planta e categoria.

3. Identificação de Subsistemas

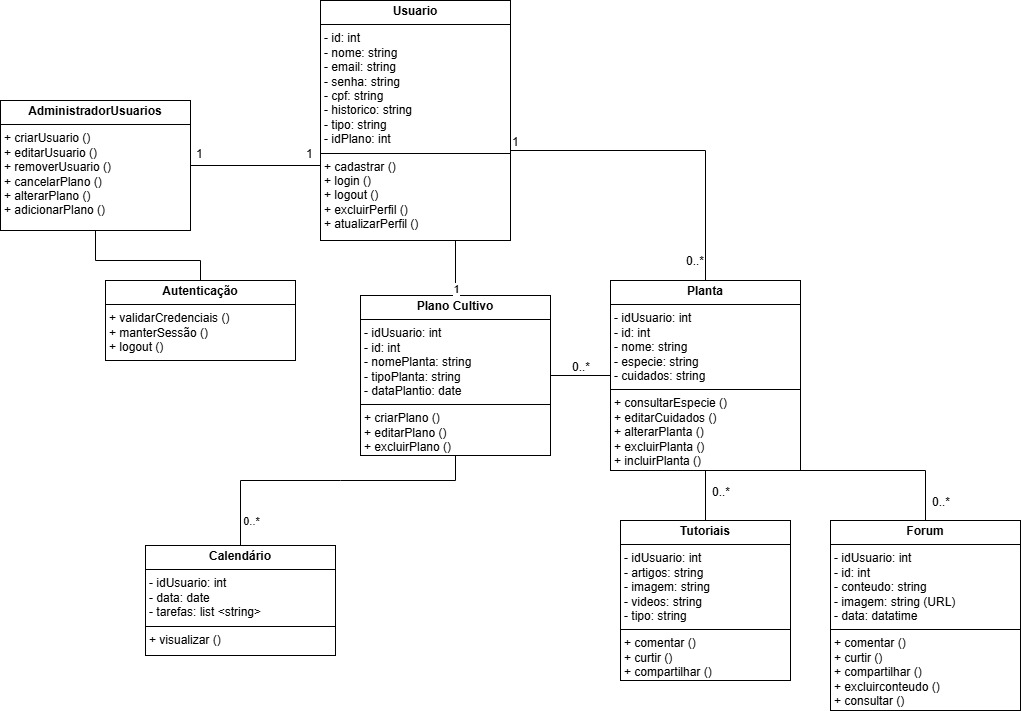
Abaixo os subsistemas identificados no contexto do presente projeto com sua descrição na tabela abaixo.

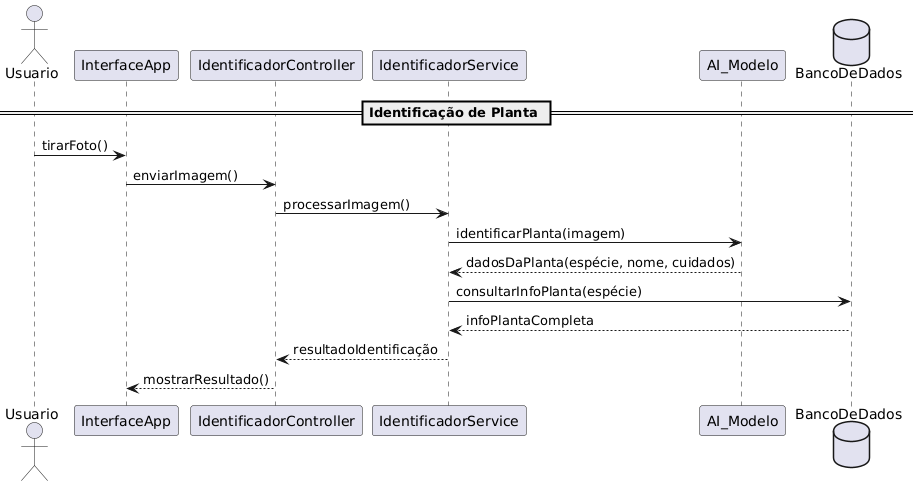
|  |  |
| --- | --- |
| Subsistema | Descrição |
| Autenticação | Gerencia o login e logout dos usuários, autenticação de credenciais e manutenção de sessões ativas. |
| Gerenciamento de usuários | Lida com o cadastro, atualização e exclusão de perfis de usuários, incluindo permissões e roles. |
| Notificações e lembretes | Sistema para alertar os usuários sobre os cuidados que devem ser fornecidos a plantação. |
| Calendário de cuidados | Exibição das tarefas programadas para cada tipo de planta. |
| Dicas e tutoriais | Acesso a artigos, informações e vídeos sobre os cuidados com os diferentes tipos de plantas. |
| Comunidade | Interação entre usuários para que eles possam compartilhar dicas, dúvidas e conselhos. |
|  | Registro do crescimento e desenvolvimento das plantas por meios de fotos e notas. |
| Monetização | Controle das versões gratuita e premium do aplicativo. |

4. Modelo de Casos de Uso

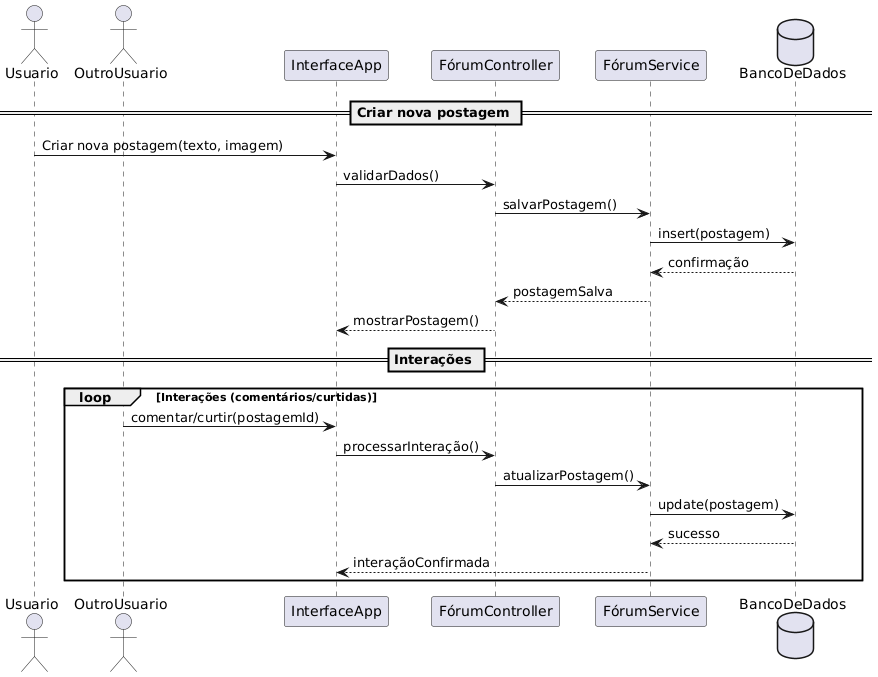
O modelo de casos de uso visa capturar e descrever as funcionalidades que um sistema deve prover para os atores que interagem com o mesmo. Os atores identificados no contexto deste projeto estão descritos na tabela abaixo.

|  |  |
| --- | --- |
| Ator | Descrição |
| Usuário | Usuário do sistema que interage com as funcionalidades do aplicativo AgroPlanner, como cadastro de plantas, lembretes e calendário de cuidados. |
| Administrador | Gerencia as permissões de usuários, mantém o controle de acessos e gerência os dados do sistema. |

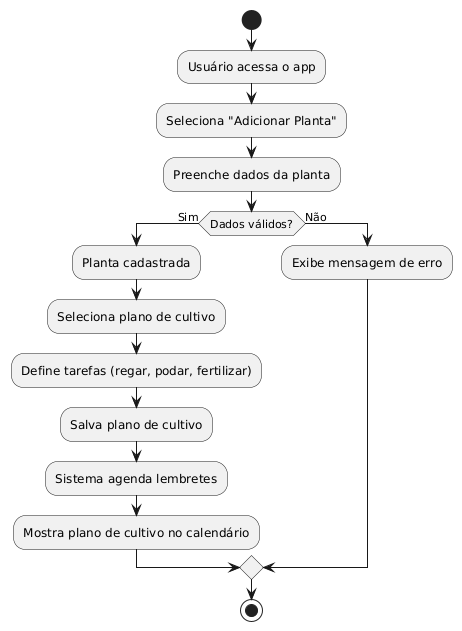
A seguir, são apresentados os diagramas de casos de uso e descrições associadas, organizados por subsistema.

**Figura 2** – Diagrama de classe do sistema todo – **Fonte:** Autor

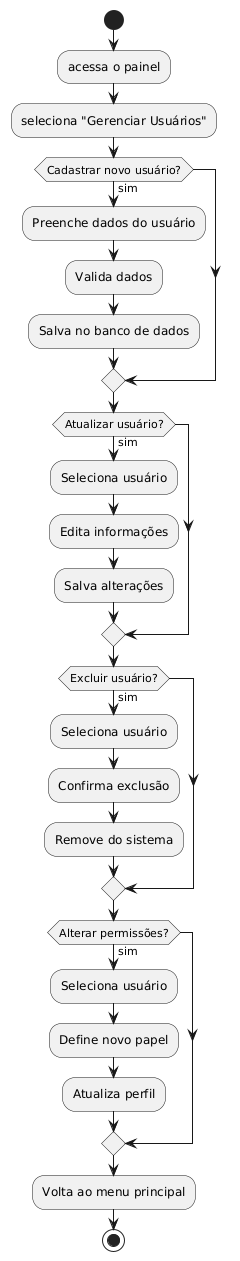
**Figura 3** – Diagrama de sequência da Identificação de Planta – **Fonte:** Autor



**Figura 4** – Diagrama de sequência do Fórum – **Fonte:** Autor



**Figura 5** – Diagrama de atividade para usuário adicionar planta – **Fonte:** Autor



**Figura 6** – Diagrama de atividades do “administrador” do sistema – **Fonte:** Autor

4.1 Subsistema AgroPlanner

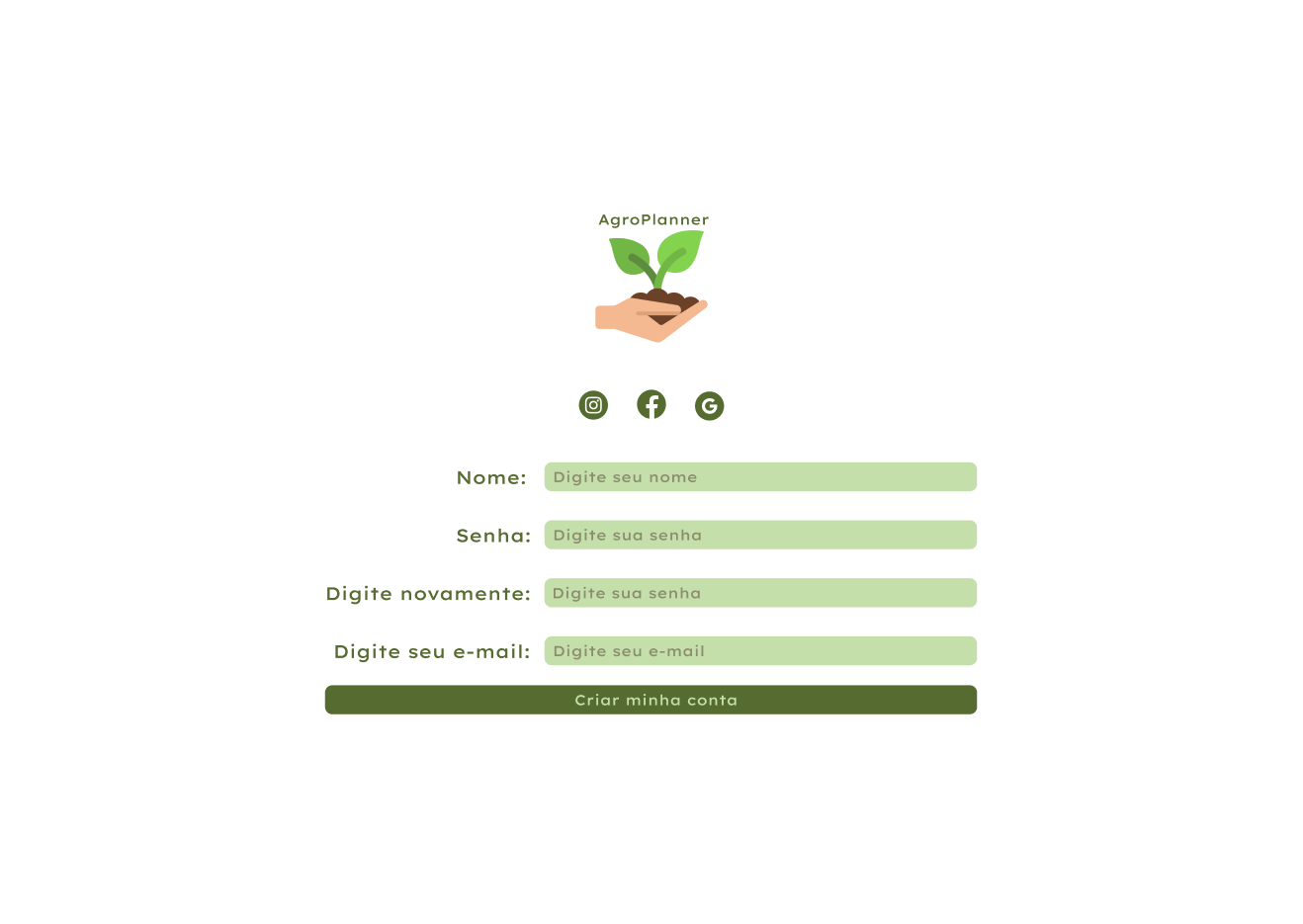
Abaixo são apresentadas as descrições de cada um dos casos de uso identificados. Os casos de uso cadastrais envolvendo inclusão, alteração, consulta e exclusão são descritos na tabela abaixo, segundo o padrão da organização.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Identificador | Caso de uso | Ações Possíveis | Observações | Requisitos | Classes |
| UC01 | Cadastrar Planta | Incluir, alterar, consultar, excluir planta | Usuário deve estar logado | RF4,  RF5,  RF6 | Planta |
| UC02 | Cadastrar Usuário | Incluir, alterar, consultar, excluir usuário | Autenticação necessária para exclusão | RF1,  RF2,  RF3 | Usuário |
| UC03 | Gerenciar Lembretes | Criar, alterar, excluir lembretes | Lembretes configuráveis por planta | RF7,  RF8 | Lembretes |
| UC04 | Gerenciar Calendários | Consultar, alterar tarefas no calendário | Relacionado a atividades com plantas | |  | | --- | | RF8, RF9 |  |  | | --- | |  | | Calendários |
| UC05 | Acessar Comunidades | Postar, consultar, excluir postagens na comunidade | Interação entre usuários | RF10 | Comunidade, Post |

1. Prototipação de Telas

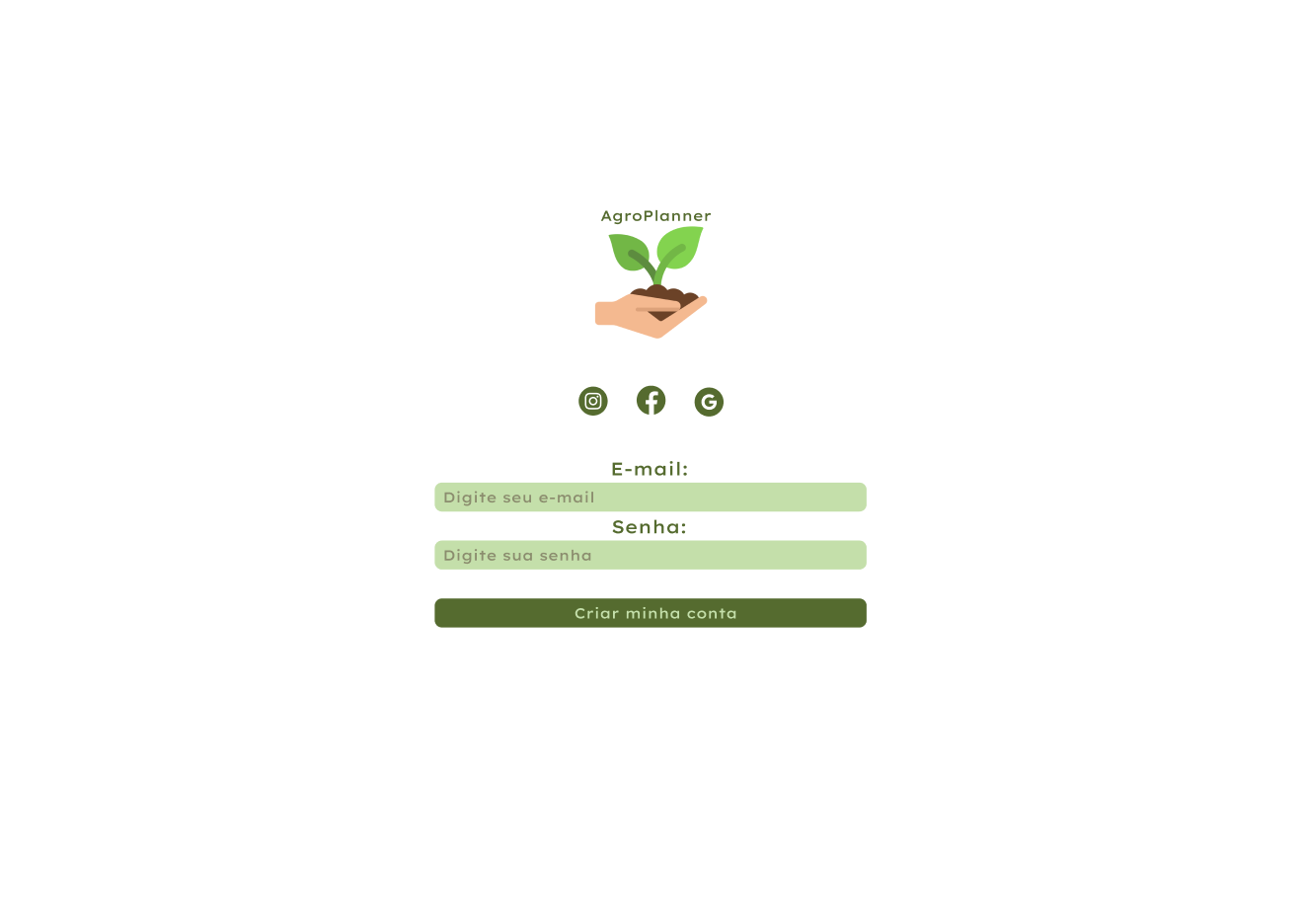
A seguir serão apresentados os protótipos das telas do sistema. Elas poderão sofrer alteração.

5.1 Cadastrar Usuário



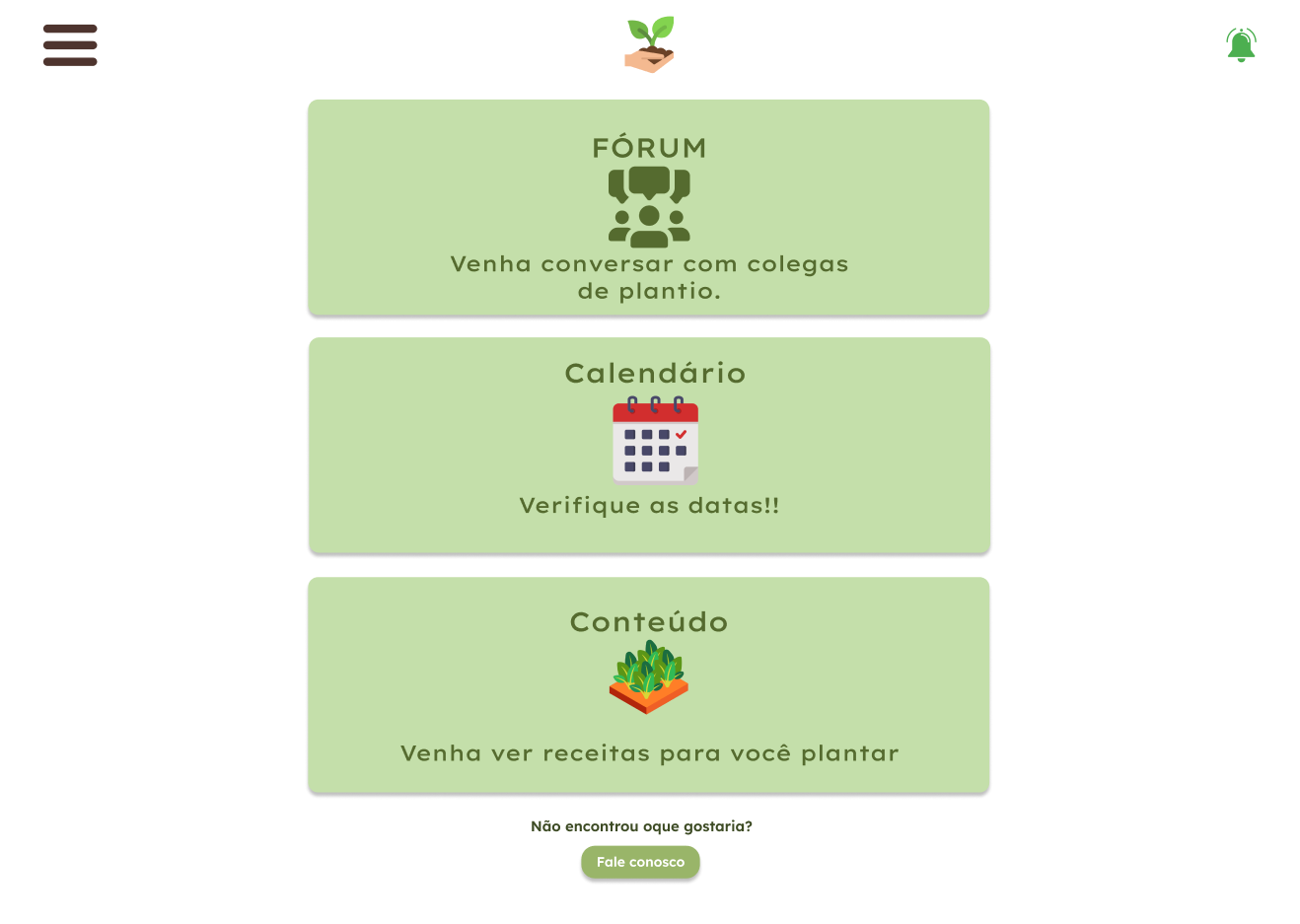
**Figura 7** – Protótipo de tela de cadastro de usuário. – **Fonte:** Autor

5.2 Login Usuário



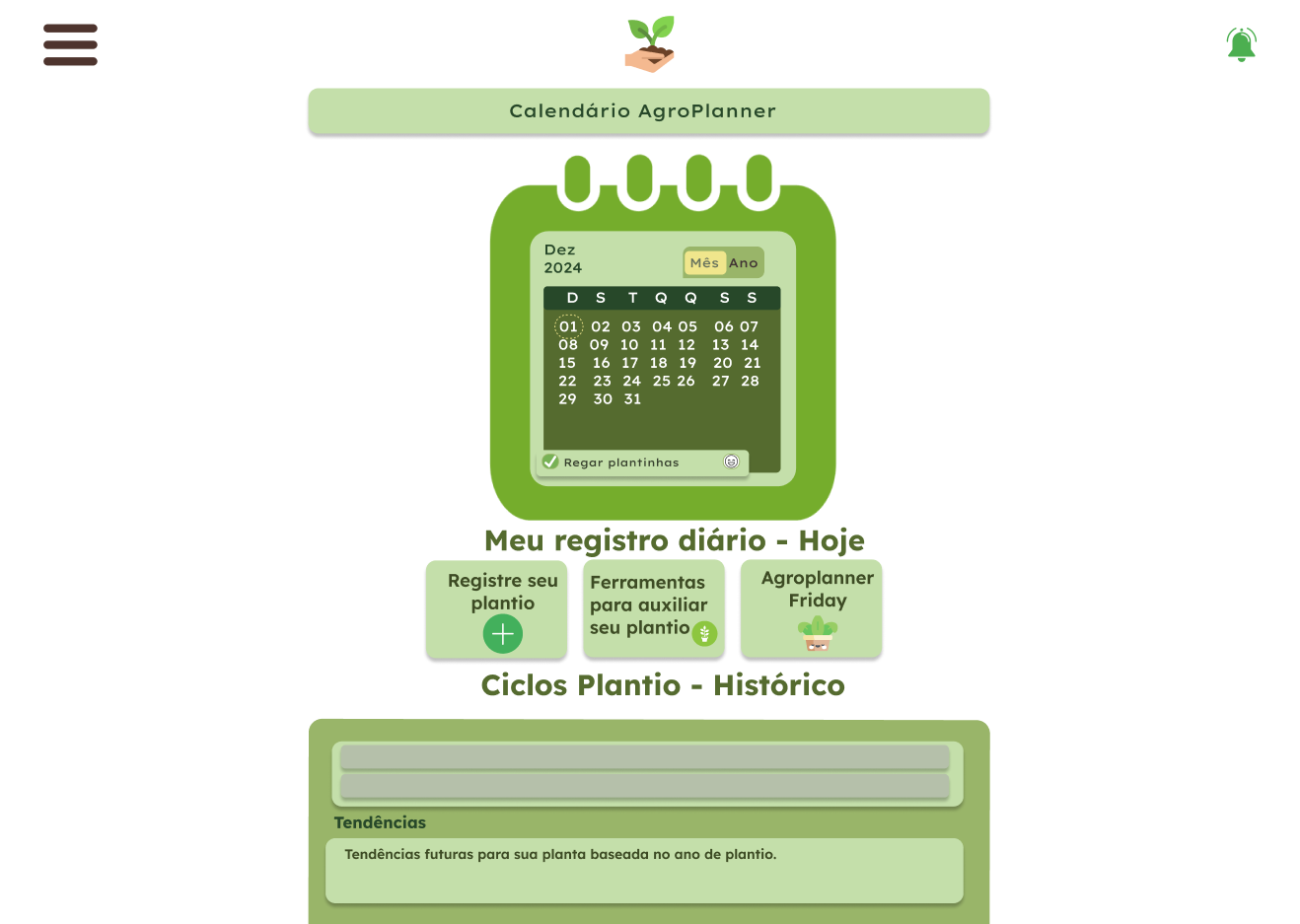
**Figura 8** – Protótipo de tela de login de usuário. – **Fonte:** Autor

5.3 Página Inicial



**Figura 9** – Protótipo de tela inicial. – **Fonte:** Autor

5.4 Página do calendário



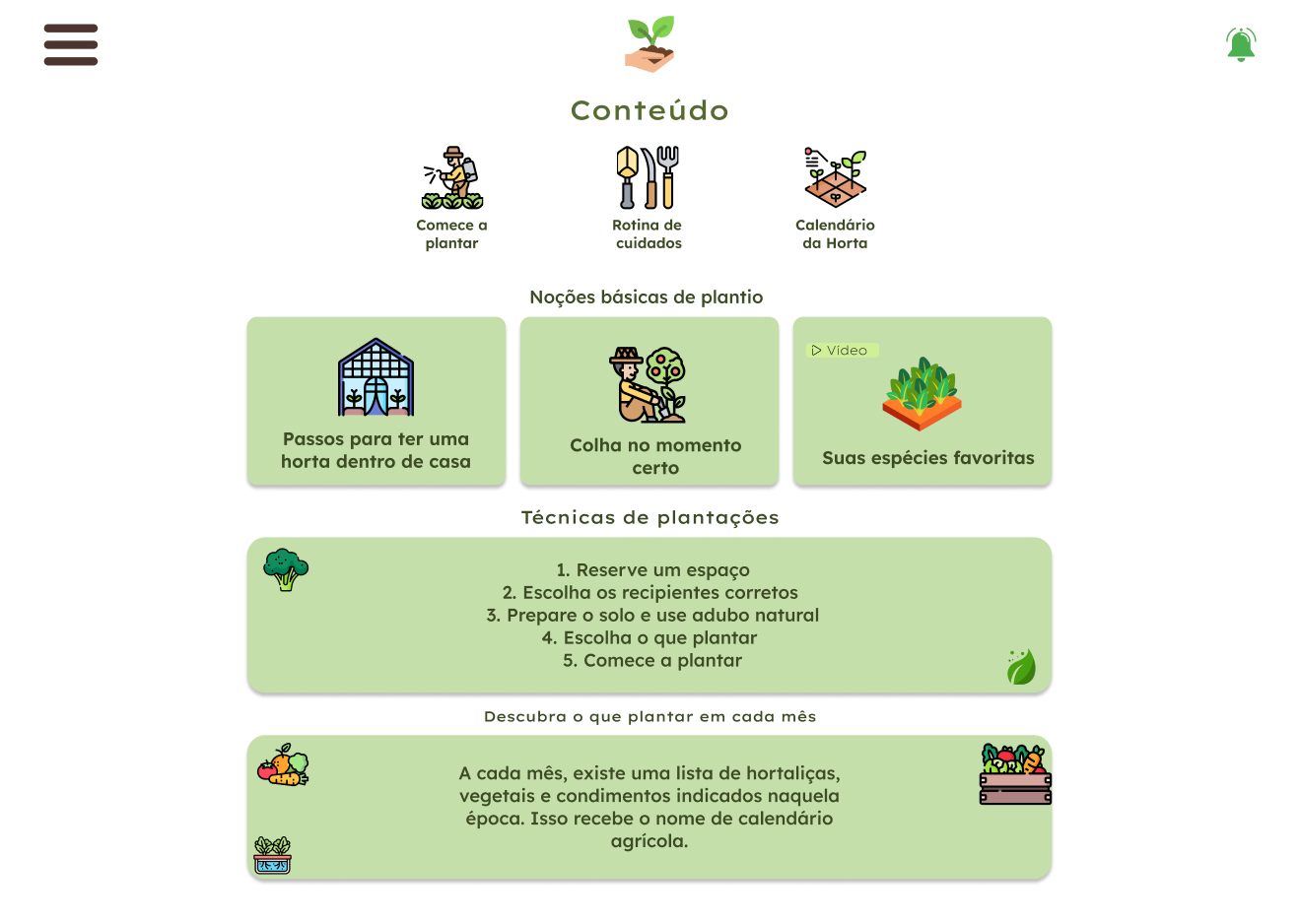
**Figura 10** – Protótipo da página do calendário. – **Fonte:** Autor

5.5 Fórum



**Figura 11** – Protótipo do Fórum. – **Fonte:** Autor

5.6 Página do conteúdo.

.

**Figura 12** – Protótipo da página do conteúdo. – **Fonte:** Autor

1. Regras de Negócio

As regras de negócio do projeto **AgroPlanner** definem critérios essenciais para o funcionamento correto do sistema, garantindo que as ações dos usuários respeitem limites definidos. Elas organizam como informações são cadastradas, alteradas e acessadas, especialmente em áreas como o plano de cultivo, a comunidade e o controle de usuários. Essas regras ajudam a manter a consistência dos dados, a segurança dos perfis e a padronização das operações. Aplicadas em todo o sistema, elas asseguram que o uso do AgroPlanner seja eficiente, confiável e alinhado aos objetivos propostos.

|  |  |
| --- | --- |
| **Identificador** | **Regra** |
| RN01 | Cada usuário pode ter apenas um cadastro com login e CPF únicos. |
| RN02 | Login e CPF não podem ser alterados após o cadastro. |
| RN03 | Usuário precisa estar autenticado para cadastrar, editar ou excluir plantas. |
| RN04 | Apenas usuários logados podem postar na comunidade. |
| RN05 | Cada lembrete deve estar associado a uma planta registrada. |
| RN06 | Planos de cultivo devem conter pelo menos uma planta associada. |
| RN07 | Postagens na comunidade devem ser moderadas em caso de denúncias. |
| RN08 | Um usuário pode excluir apenas postagens feitas por ele mesmo. |
| RN09 | As tarefas no calendário devem respeitar as datas de plantio conforme consta no plano. |

1. Implementação

7.1 Estrutura de Pastas e Arquivos

/agroplanner/

├── src/

│ ├── app.js # Configuração principal da aplicação (Express, middlewares, etc.)

│ ├── config/ # Arquivos de configuração (DB, variáveis de ambiente, etc.)

│ │ └── db.js # Conexão com o banco de dados

│ ├── controllers/ # Lógica de negócio de cada rota (manipuladores de requisição)

│ │ ├── forumController.js

│ │ ├── plantioController.js

│ │ ├── userController.js

│ │ ├── conteudoController.js

│ │ └── notificacaoController.js

│ ├── middlewares/ # Funções middleware reutilizáveis (autenticação, validação, etc.)

│ ├── models/ # Definição dos modelos de dados (Mongoose schemas, Sequelize models)

│ │ ├── Forum.js

│ │ ├── Plantio.js

│ │ ├── User.js

│ │ ├── Conteudo.js

│ │ └── Notificacao.js

│ ├── routes/ # Definição das rotas da API

│ │ ├── authRoutes.js # Rotas de autenticação (login, registro)

│ │ ├── forumRoutes.js

│ │ ├── plantioRoutes.js

│ │ ├── userRoutes.js

│ │ ├── conteudoRoutes.js

│ │ └── notificacaoRoutes.js

│ ├── services/ # Lógica de negócio complexa que pode ser reutilizada por controllers

│ │ ├── uploadService.js # Para lidar com uploads de arquivos

│ │ └── ...

│ └── utils/ # Funções utilitárias diversas (formatadores, validadores, helpers)

├── tests/ # Testes unitários e de integração

├── uploads/ # Diretório para armazenamento de arquivos enviados (imagens, documentos)

├── .env # Variáveis de ambiente (ex: porta, strings de conexão)

├── .gitignore # Arquivos e pastas a serem ignorados pelo Git

├── package.json # Metadados do projeto e dependências

└── README.md # Documentação do projeto

7.2 Tecnologias Utilizadas

Front-end:  
Framework: React  
Gerenciamento de Rotas: React Router  
SPA: Single Page Application  
Integração com back-end: fetch ou axios  
  
Back-end:  
Plataforma: Node.js  
Framework: Express.js  
Banco de Dados: MySQL  
Upload de Arquivos: Multer  
Conexão com DB: mysql

* **React:** Para construir uma interface de usuário dinâmica e modular. A ideia é ter componentes reutilizáveis que tornem a experiência de navegação superfluida.
* **React Router:** Essencial para gerenciar as "páginas" da nossa aplicação, garantindo que o usuário possa navegar sem recarregar a tela a cada clique, característica de uma **SPA (Single Page Application)**. Isso torna a navegação muito mais rápida e agradável.
* **Fetch API ou Axios:** Para a comunicação com o back-end. Prefiro a robustez do Axios para lidar com as requisições HTTP, tornando a troca de dados com o servidor mais simples e eficiente.
* **Estilização (a definir):** Ainda estou avaliando a melhor abordagem para o CSS, buscando algo que nos dê agilidade e um visual moderno, pensando na responsividade para todos os dispositivos.

**No coração da aplicação (Back-end), a base é:**

* **Node.js com Express.js:** Uma combinação poderosa. O Node.js me dá a velocidade e escalabilidade que precisamos, enquanto o Express.js oferece um framework leve e flexível para construir nossa API, gerenciando todas as requisições e respostas.
* **MySQL:** Nosso banco de dados relacional escolhido. É robusto e confiável para armazenar todas as informações cruciais do AgroPlanner, desde dados de plantio até interações no fórum.
* **Multer:** Para facilitar o upload de arquivos, como as imagens de receitas que os usuários poderão compartilhar.
* **Driver mysql:** Para a conexão direta com o banco de dados. Estou explorando se um ORM como o Sequelize pode agregar mais valor para uma gestão de dados ainda mais eficiente no futuro.

7.3 Códigos principais

* **Cadastro de Usuário (userController.js):** Endpoint para criar novos usuários, coletando nome, email e senha para assegurar o acesso à plataforma.
* **Login de Usuário (userController.js):** Lida com a autenticação, verificando as credenciais e retornando os dados do usuário para o front-end, garantindo uma sessão segura.
* **Envio de Pergunta no Fórum (forumController.js):** Permite que os usuários postem novas perguntas, com título e mensagem, fomentando a comunidade e a troca de conhecimentos.
* **Cadastro de Plantio (plantioController.js):** Para que os usuários registrem suas culturas, incluindo data de plantio e observações importantes, auxiliando no gerenciamento agrícola.
* **Cadastro de Conteúdo/Receita (conteudoController.js via conteudoRoutes.js):** Permite o envio de novas receitas, gerenciando o upload de imagens e definindo um status inicial pendente para moderação.

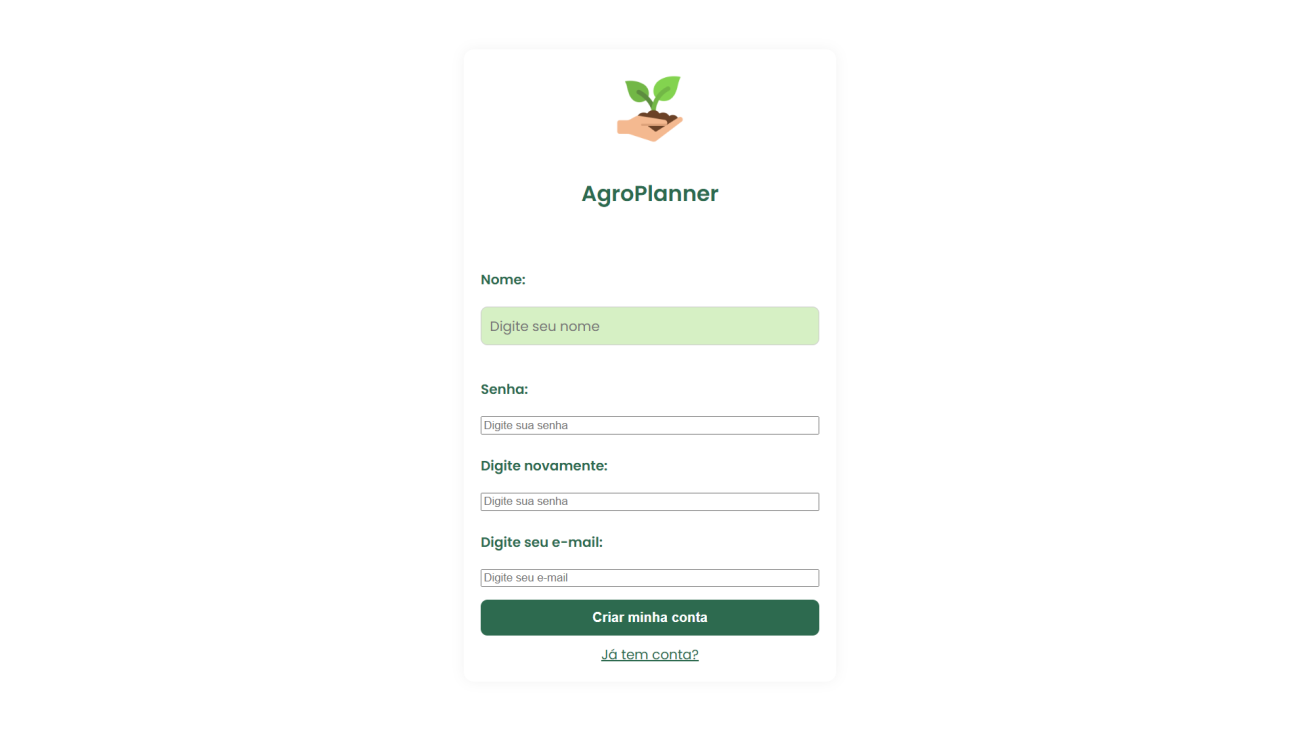
7.4 Banco de Dados

usuarios (id, nome, email, senha)  
forum (id, titulo, mensagem)  
plantios (id, data, cultura, observacoes)  
conteudos (id, nomePlanta, categoria, epoca, temperatura, solo, rega, sol, instrucoes, id\_autor, status, data\_publicacao)  
imagens\_conteudo (id\_conteudo, url)  
notificacoes (id, id\_usuario, mensagem, criado\_em)

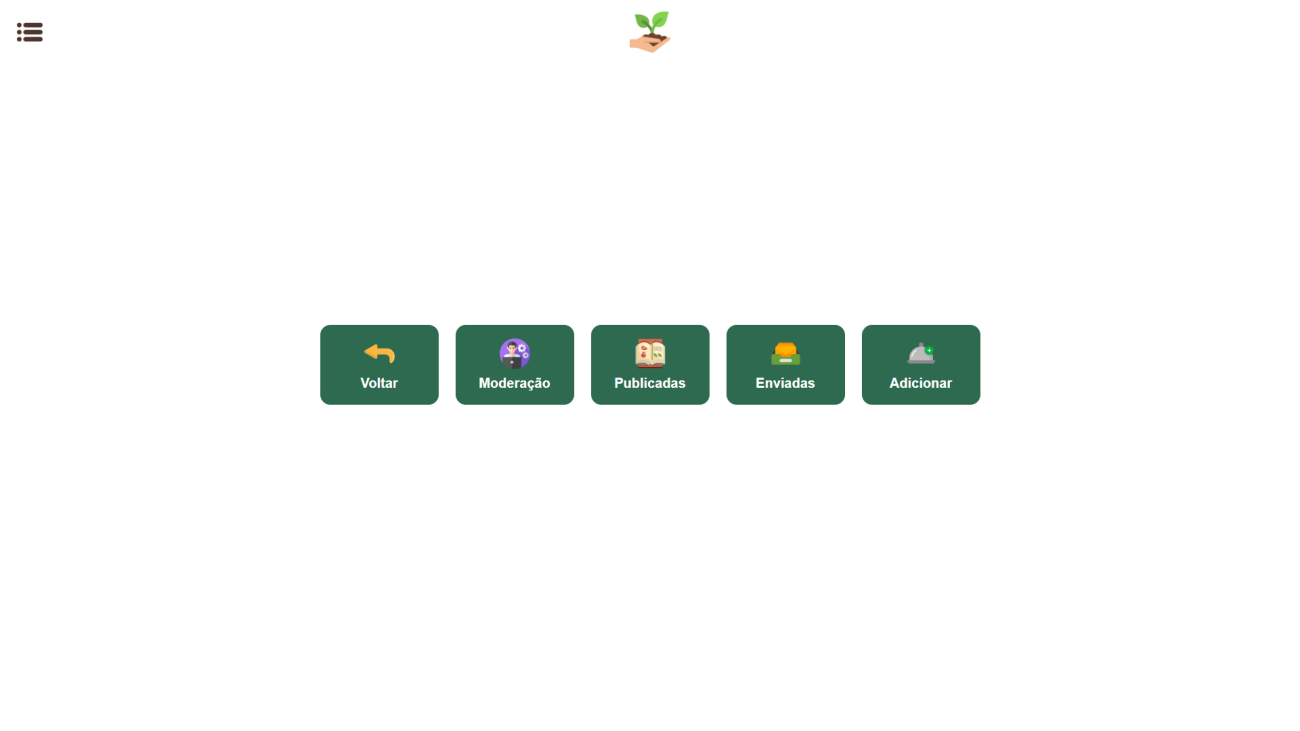
**7.5 Telas de Login**

### 

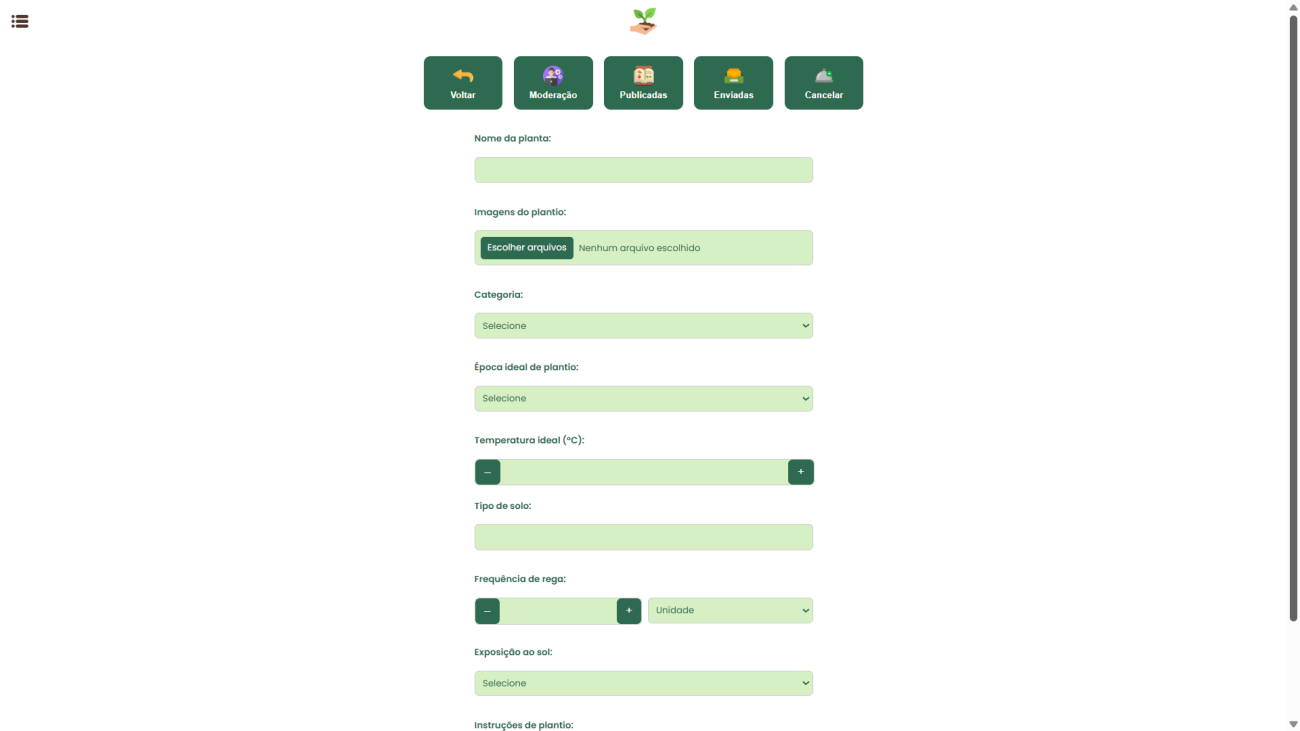
7.6 Telas de Login



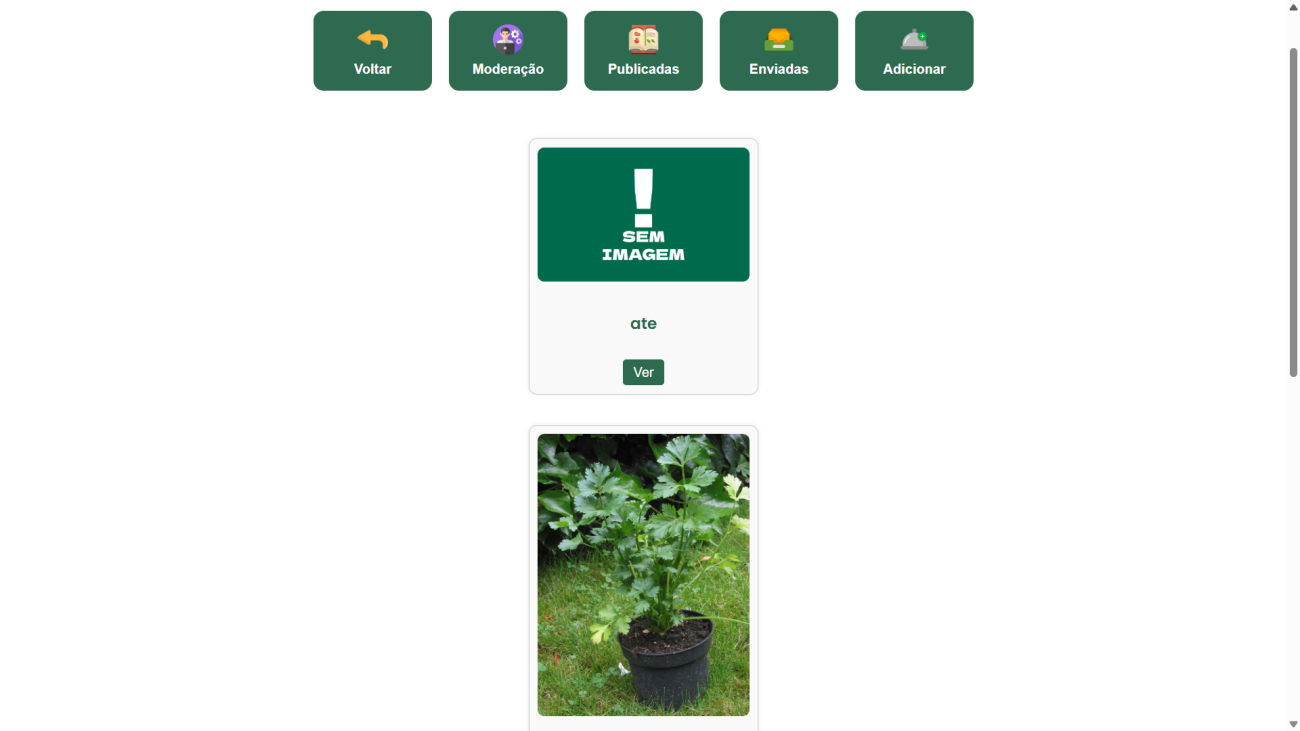
7.7 Telas de Início



7.8 Tela de Adicionar de Planta



7.9 Tela de Publicação da Planta



### **8. Conclusão**

O AgroPlanner cumpre seu objetivo de auxiliar o gerenciamento de plantios, compartilhamento de dicas e suporte comunitário via fórum.  
Melhorias futuras incluem: autenticação com JWT, proteção de uploads, hash de senha, testes automatizados e front-end completo.

**9. Anexos**

Link de repositório do GIT: <https://github.com/joaolatczuk/ES>

Dicionário de dados:

|  |  |  |  |
| --- | --- | --- | --- |
| **Campo** | **Tipo de Dado** | **Descrição** | **Restrições** |
| id | INT | Identificador único | Chave primária, auto incremento |
| nome | VARCHAR(100) | Nome do usuário | Obrigatório |
| email | VARCHAR(100) | E-mail do usuário | Único, obrigatório |
| senha | VARCHAR(100) | Senha do usuário | Obrigatório |
| **Campo** | **Tipo de Dado** | **Descrição** | **Restrições** |
| id | INT | Identificador da pergunta | Chave primária, auto incremento |
| titulo | VARCHAR(255) | Título da pergunta | Obrigatório |
| mensagem | TEXT | Conteúdo da pergunta | Obrigatório |
| **Campo** | **Tipo de Dado** | **Descrição** | **Restrições** |
| id | INT | Identificador do registro | Chave primária, auto incremento |
| data | DATE | Data do plantio | Obrigatório |
| cultura | VARCHAR(100) | Nome da cultura plantada | Obrigatório |
| observacoes | TEXT | Observações gerais | Opcional |
| **Campo** | **Tipo de Dado** | **Descrição** | **Restrições** |
| id | INT | Identificador do conteúdo | Chave primária, auto incremento |
| nomePlanta | VARCHAR(100) | Nome da planta | Obrigatório |
| categoria | VARCHAR(100) | Categoria da planta (ex: hortaliça) | Opcional |
| epoca | VARCHAR(100) | Época ideal para plantio | Opcional |
| temperatura | VARCHAR(50) | Temperatura ideal | Opcional |
| solo | VARCHAR(100) | Tipo de solo | Opcional |
| rega | VARCHAR(100) | Frequência de rega | Opcional |
| sol | VARCHAR(100) | Exposição solar necessária | Opcional |
| instrucoes | TEXT | Instruções de plantio | Opcional |
| id\_autor | INT | Referência ao autor (usuário) | Chave estrangeira |
| data\_publicacao | DATETIME | Data de criação do conteúdo | Automático |
| status | ENUM | Status da receita (pendente, aprovado...) | Padrão: 'pendente' |
| favorita | BOOLEAN | Marca como favorita | Padrão: false |
| **Campo** | **Tipo de Dado** | **Descrição** | **Restrições** |
| id\_conteudo | INT | Referência ao conteúdo | Chave estrangeira |
| url | VARCHAR(255) | Caminho da imagem | Obrigatório |
| **Campo** | **Tipo de Dado** | **Descrição** | **Restrições** |
| id\_conteudo | INT | Referência ao conteúdo | Chave estrangeira |
| url | VARCHAR(255) | Caminho da imagem | Obrigatório |