

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA
GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO



**DESENVOLVIMENTO DE UM APLICATIVO PARA GESTÃO DE PEQUENAS
PROPRIEDADES RURAIS**

JOÃO AUGUSTO BATISTA CAMARGO

GOIÂNIA
2023

JOÃO AUGUSTO BATISTA CAMARGO

**DESENVOLVIMENTO DE UM APLICATIVO PARA GESTÃO DE PEQUENAS
PROPRIEDADES RURAIS**

Trabalho de Conclusão de Curso apresentado à Escola Politécnica, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para a obtenção do título de Bacharel em Engenharia da Computação.

Orientador:

Prof. Me. Aníbal Santos Jukemura

Banca Examinadora:

Prof. Me. Rafael Leal Martins

Prof. Me. Geraldo Valeriano Ribeiro

GOIÂNIA
2023

JOÃO AUGUSTO BATISTA CAMARGO

**DESENVOLVIMENTO DE UM APLICATIVO PARA GESTÃO DE PEQUENAS
PROPRIEDADES RURAIS**

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção do título de Bacharel em Engenharia de Computação, e aprovado em sua forma final pela Escola Politécnica, da Pontifícia Universidade Católica de Goiás em 15 de Dezembro de 2023.

Orientador: Prof. Me. Aníbal Santos Jukemura

Prof. Me. Rafael Leal Martins

Prof. Me. Geraldo Valeriano Ribeiro

GOIÂNIA
2023

AGRADECIMENTOS

Em primeiro lugar agradeço a Deus pelo dom da vida e por cada pessoa que colocou em minha vida para me incentivar e apoiar a chegar até aqui. E à virgem Maria, por me conduzir e interceder por mim.

À minha mãe, Luzdalva Rosa de Camargo Batista, por ter me guiado e cuidado de mim em todos os momentos da minha vida, por suas orações e por suas batalhas diárias. Ao meu pai, Mário Batista (in memorian), por ter se dedicado e sempre trabalhado para dar o melhor para mim e para minha família. Aos meus irmãos Patrícia Nayara Leite Bessa, Mário Henrique Camargo Batista, Ana Laura Camargo Batista e Ana Livia Batista Camargo, pela paciência e ensinamentos compartilhados.

À minha esposa, Brígida Cristina Toscano Gonçalves, por ter me incentivado e estar ao meu lado mesmo nos momentos mais difíceis, pelo seu amor, que me ajudou a passar por esses momentos.

Ao meu orientador, Anibal Santos Jukemura, pelas dicas e conselhos, pela paciência e por me guiar nessa etapa da minha formação.

RESUMO

A falta de tempo e de conhecimento induzem a má gestão de um negócio, ambos os fatores estão fortemente presentes no cotidiano de trabalhadores e proprietários rurais, essa dificuldade acaba se tornando um empecilho que pode impedir a expansão do agronegócio por parte do produtor e gerar prejuízos com consequências que vão desde um descontrole financeiro momentâneo à falência do negócio em casos extremos. Com ferramentas devidamente desenvolvidas com base na necessidade desses usuários abrem-se portas para um comércio que até o momento ainda é pouco explorado, garantindo uma possibilidade de gerenciamento facilitado, com economia de tempo e uma visão mais ampla de análise dos indicadores da fazenda, tudo isso sendo pareado a uma interface amigável e simples para máximo aproveitamento do usuário. Com o uso de um aplicativo móvel é possível garantir que a aplicação encaixe nas demandas do produtor rural, pois garante facilidade, rapidez e robustez se adaptando até mesmo às condições geográficas.

Palavras-chave: Gestão, propriedade rural, aplicativo móvel.

ABSTRACT

The lack of time and know ledge leads to poor management of a business, both factors strongly present in the daily life of rural workers and landowners, this difficulty ends up becoming a hindrance that can prevent the expansion of agribusiness by the producer and generate losses with consequences ranging from momentary financial discontrol to bankruptcy of the business in extreme cases. With tools properly developed based on the needs of these users, doors are opened for a trade that is still little explored, guaranteeing a possibility off acilitated management, with time saving sand a broader view of farm indicators analysis, all paired with a friendly and simple interface for maximum user utilization. With the use of a mobile application, it is possible to ensure that the application fits the demands of the rural producer, as it guarantees ease, speed, and robustness, adapting even to geographic conditions.

Key-Words: Management, rural property, mobile application.

LISTA DE ILUSTRAÇÕES

Figura 1 – Descrição de um requisito do trabalhador analisado.....	17
Figura 2 – Caso de uso utilizado no trabalho citado.....	18
Figura 3 – Interface android studio.....	20
Figura 4 –Arquitetura do Flutter.....	21
Figura 5 – Modelo de arquitetura MVVM.....	24
Figura 6 – Interface do astah uml.....	25
Figura 7 – Estrutura de pastas geral	27
Figura 8 – Pastas do diretório lib.....	27
Figura 9 – Diagrama geral de caso de uso (astah)	30
Figura 10 – Tela de login.....	32
Figura 11 – Listagem de lançamento	33
Figura 12 – Formulário de novo lançamento.....	34
Figura 13 – Formulário para editar lançamento.....	35

LISTA DE TABELAS

Tabela 1 – Requisitos funcionais.....	16
Tabela 2 – Tabela manter fluxo de caixa	28

LISTA DE SIGLAS

MVP
IDE
CRUD

Minimun Viable Product
Integrated Development Enviroment
Create, Read, Update e Delete

SUMÁRIO

1. INTRODUÇÃO	12
1.1 Objetivos	14
1.1.1 Objetivos específicos	14
1.2 Justificativa	14
1.3 Organização do documento	15
2. REFERENCIAL TEÓRICO	16
3. PROCEDIMENTOS METODOLÓGICOS	18
3.1 Metodologia de Pesquisa	19
3.2 Ambiente de desenvolvimento	19
3.2.2 Dart e Flutter	20
3.2.4 Por que usar o Flutter?	23
3.2.5 Arquitetura MVVM	23
3.2.7 Astah UML	24
4. DESCRIÇÃO DA APLICAÇÃO	26
4.1 Escopo	26
4.2 Estrutura de pastas da aplicação	26
4.3 Requisitos	28
4.4 Requisitos funcionais	28
4.5 Requisitos não funcionais	28
4.6 Diagrama de caso de uso	29
4.6.1 Diagrama Geral de Caso de Uso	29
4.7 Especificação do caso de uso manter fluxo de caixa	30
4.8 Interface	31
4.8.1 Tela de login	31
4.8.2 Listagem de lançamentos	32
4.8.3 Formulário novo lançamento	33
4.8.4 Formulário de edição de lançamento	34
5. CONCLUSÃO	36
6. TRABALHOS FUTUROS	37
7. REFERÊNCIAS	38

1. INTRODUÇÃO

Diante das mudanças oriundas da globalização e da disseminação da internet, é visto não apenas um Brasil, mas um mundo cada vez mais conectado e sincronizado em relação às informações e a velocidade em que essas são conectadas aos aparelhos (celulares, computadores, ...), tais mudanças trazem impactos à sociedade, tudo acontece muito rápido e quem não acompanhar as alterações vindas com um turbilhão de informações e dados pode “ficar pra trás” e acabar sendo prejudicado principalmente na vertente de negócios.

A falta de gerenciamento é uFm dos motivos que mais levam empresas a falirem. Em propriedades rurais tal evidência não é diferente. Esse fato repercute sobre um mercado pouco explorado e com grande potencial de alcance. A agropecuária familiar tem impacto significativo na produção de produtos de diversas categorias, e é de extrema importância para o Brasil, que os produtores estejam estabilizados e organizados financeiramente para continuarem suprindo a demanda de tais produtos.

Segundo Kay et al. (2014), a dificuldade relacionada à gestão de pequenas propriedades rurais está ligada ao fato de uma mesma pessoa ser responsável por vários setores dentro da propriedade, e em um aspecto mais amplo, essa pessoa ser o próprio dono dessa propriedade, ou até mesmo outra pessoa do grupo familiar.

Com o crescente aumento do uso de smartphones e a facilidade de acesso a esses aparelhos, tem sido cada vez mais comum o uso de aplicativos como ferramenta para facilitar e automatizar tarefas. Na agropecuária apesar de uma menor incidência, existem ferramentas com tecnologias de ponta, acopladas aos mais novos hardwares do mercado, sistemas complexos e de extrema precisão. O foco em grandes empresas e grupos comerciais dificulta a chegada de tecnologias à pequenos produtores, e usuários com pouca informação ou baixa escolaridade, além disso o fato de alguns sistemas serem de interface extremamente complexa aumenta ainda mais essa barreira de entrada.

Com isso, visto que a dificuldade de gerenciar está diretamente associada à falta de tempo, é recomendado o uso de um aplicativo para auxiliar na gestão financeira e organizacional de uma propriedade rural.

Além disso, é preciso destacar que, de acordo com Batalha et al. (2005), as tecnologias e ferramentas introduzidas na agricultura familiar são mal adaptadas ao

cotidiano e não são desenvolvidas para o uso específico de pequenos produtores sendo, por sua vez, deveras complexas e às vezes apresentam muitas funcionalidades obsoletas ou que não refletem a realidade. Em virtude disso, há uma grande demanda gerada pela necessidade por parte do produtor rural, por uma aplicação que o atenda e simplifique a gestão de seu negócio.

Segundo Kay et. Al (2014, p.155) muitas das funções e tarefas de gestão podem contribuir às economias de escala. Compra de insumos, comercialização de produtos, contabilidade, coleta de informações, planejamento e supervisão do trabalho são exemplos disso. A capacidade limitada de gestão sempre foi considerada uma causa clássica de deseconomias de tamanho. À medida que o negócio agropecuário fica maior, torna-se mais difícil o gestor conhecer todos os aspectos de todos os empreendimentos e organizar e supervisionar corretamente todas as atividades.

O agronegócio é extremamente estratégico para a economia brasileira, já que representa pouco mais de 20% do nosso PIB. Mesmo que a maior parte da população não tenha a percepção real desses avanços, o segmento tem sido palco de algumas das mais importantes transformações em tecnologia nos últimos anos. Justificado através de um levantamento da Comissão Brasileira de Agricultura de Precisão (CBAP), 67% das propriedades agrícolas no Brasil já adotaram algum tipo de inovação tecnológica, dentro ou fora do campo (Sociedade Nacional de Agricultura, 2019).

Uma pesquisa realizada pela Embrapa, o Sebrae e o Inpe (Bolfé et al., 2020) com 753 produtores rurais, empresas e prestadores de serviços em agricultura digital de todas as regiões brasileiras observa que 67% desses agricultores e 58% dos prestadores de serviços indicam que o valor do investimento para aquisição de máquinas, equipamentos e/ou aplicativos ainda é o principal desafio para a implantação e a manutenção da transformação digital na propriedade. Dessa forma, percebe-se que, para importante parcela dos agricultores, especialmente pequenos e médios produtores rurais, o processo de transformação digital ainda é considerado oneroso frente à percepção atual dos potenciais benefícios econômicos (EMBRAPA, 2020).

No Censo Agropecuário de 2017, 3.897.408 estabelecimentos atenderam aos critérios e foram classificados como agricultura familiar, o que representa 77% dos estabelecimentos agropecuários levantados pelo censo. Ocupavam uma área de

80,9 milhões de hectares, ou seja, 23% da área total dos estabelecimentos agropecuários brasileiros. Nos estabelecimentos classificados como de agricultura familiar o total de pessoas ocupadas em 30/09/2017 foi de 10,1 milhões de pessoas; 67% do total, nestes estabelecimentos a média de pessoas ocupadas era de 2,6. A agricultura familiar foi responsável por 23% do valor total da produção dos estabelecimentos (IBGE, 2017).

Destaca-se, portanto, que este trabalho pretende analisar e desenvolver uma solução prática afim de servir como ferramenta e suporte de gerenciamento de finanças focado em pequenas propriedades rurais com ênfase em usuários leigos e com pouco contato com tecnologia.

1.1 Objetivos

É objetivo deste trabalho analisar, projetar e desenvolver uma aplicação MVP (Minimum Viable Product) para dispositivos móveis capaz de auxiliar da gestão financeira e organizacional de uma propriedade rural de pequeno porte, através do framework Flutter.

1.1.1 Objetivos específicos

- Levantar requisitos e analisá-los por meio de diagramas e técnicas específicas.
- Desenvolver uma aplicação CRUD (Create, Read, Update, Delete).
- Demonstrar a viabilidade do aplicativo através do framework Flutter e linguagem DART.
- Demonstrar a real utilidade do framework Flutter para o desenvolvimento de aplicativos em dispositivos móveis.

1.2 Justificativa

O crescente avanço da tecnologia, e a demanda decorrente de uma área que vem ganhando cada vez mais valor no Brasil e no mundo, a agropecuária, fazem com que seja necessária a adaptação e adequação dos pequenos produtores rurais, a fim de torná-los aptos a acompanharem o mercado de trabalho, com a capacidade

de poderem minimizar prejuízos, economizando tempo e melhorando os processos dentro de sua propriedade rural. Dentro desse cenário, um aplicativo de gestão de propriedade rural, com foco ao pequeno empresário, apresenta-se como solução viável e de simples acesso.

1.3 Organização do documento

Este trabalho está organizado em 6 capítulos, que descrevem desde a introdução até as referências utilizados no decorrer do desenvolvimento deste trabalho.

- O capítulo 2 contém o referencial teórico, onde é analisado um trabalho acadêmico que serve de base para este trabalho, além da descrição da metodologia utilizada e das ferramentas.
- No capítulo 3 estão inseridas as informações sobre a aplicação CRUD, bem como a interface, componentes e estrutura de desenvolvimento utilizada.
- O capítulo 4, este capítulo é a conclusão, e faz uma análise do desenvolvimento da aplicação junto ao texto, resumindo o que foi feito na prática.
- O capítulo 5 expõe os trabalhos futuros, as partes do trabalho que serão acopladas futuramente e módulos que façam sentido ser implementados para auxiliar devidamente os pequenos produtores rurais.
- O capítulo 6 contém as referências dos artigos, sites, livros e quaisquer informações que serviram de suporte para a construção teórica e prática deste trabalho.

2. REFERENCIAL TEÓRICO

No trabalho de Ataídes (2021), que será usado como parte do referencial teórico, serão apresentados o desenvolvimento e análise de uma tecnologia para resolver problemas específicos de pequenos produtores rurais. As metodologias e processos ligados à Engenharia de Software que também estão presentes neste trabalho serão abordadas neste capítulo, como parte do estudo para desenvolvimento e justificativa.

Neste trabalho, foram feitos levantamentos e especificações de requisitos identificando as necessidades do usuário que permitiram a modelagem das funcionalidades essenciais do sistema, através de tabelas a descrição de cada requisito. Cada item foi descrito juntamente com o nível de prioridade e métodos usados para estabelecer os requisitos funcionais e não funcionais.

A Tabela 1 mostra os requisitos funcionais referentes ao administrador.

Tabela 1 – Requisitos funcionais

	Manter administrador
RF001	✓ Cadastrar administrador
RF002	✓ Consultar administrador
RF003	✓ Alterar administrador

Fonte: Ataídes (2021)

Para os requisitos funcionais, o autor realizou uma análise do problema a ser resolvido, destacando os aspectos envolvidos, e expondo as ações que o sistema irá executar. É por meio dos requisitos funcionais que são identificadas as funcionalidades do sistema, e nesta etapa o autor fez um detalhamento de funcionalidades com níveis de prioridade. Além disso faz uso de tópicos identificados pelo requisito que representam para especificar cada requisito.

A Figura 1 descreve os requisitos funcionais do administrador, sendo o caso de uso “Manter administrador”.

Figura 1 – Descrição de um requisito do trabalho analisado

5.2.2 Manter administrador

Este tópico tem como finalidade mostrar todos os requisitos funcionais do administrador.

[RF001] Cadastrar administrador

Ator: Administrador

Prioridade: Essencial Importante Desejável

Entradas e pré-condições: Informar o nome completo, usuário e senha.

Saídas e pós-condições: O sistema mostrará uma mensagem de confirmação de dados cadastrados com sucesso. O sistema exibirá uma mensagem informando o erro.

[RF002] Consultar administrador

Ator: Administrador

Prioridade: Essencial Importante Desejável

Entradas e pré-condições: O ator deverá estar logado no sistema e o administrador deve estar cadastrado.

Saídas e pós-condições: O sistema mostrará as informações do administrador.

[RF003] Alterar administrador

Ator: Administrador

Prioridade: Essencial Importante Desejável

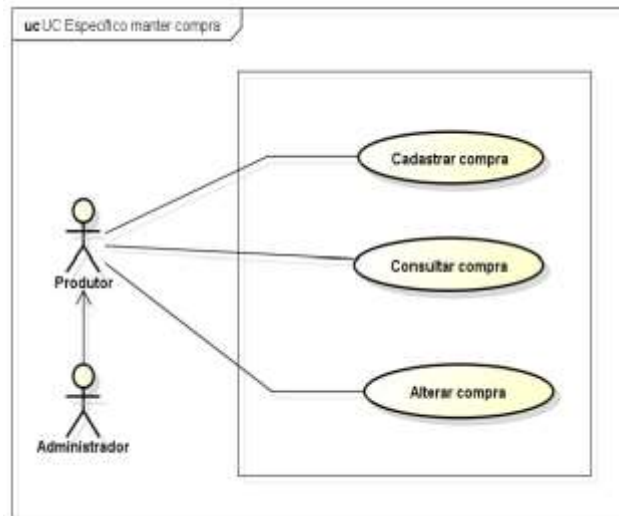
Entradas e pré-condições: O ator deverá estar logado no sistema e o administrador deve estar cadastrado.

Saídas e pós-condições: O sistema mostrará uma mensagem de confirmação de dados alterados com sucesso. O sistema exibirá uma mensagem informando o erro.

Fonte: Ataídes (2021)

O autor faz uso de diagramas de caso de uso na Figura 2 para exemplificar a comunicação no sistema e demonstrar uma ação.

Figura 2 – Caso de uso utilizado no trabalho citado



Fonte: Ataídes (2021)

Os requisitos não funcionais, diferentemente dos funcionais, não estão ligados às funções do sistema em si, mas sim à forma como ele se comporta, e também descrevem aspectos ligados a limitações de hardware e de software

O trabalho analisado consiste em um sistema web desenvolvido com foco em controle de partos de bezerros e saúde animal, destinando a atenção a processos mais complexos e que requerem um certo nível técnico de veterinários ou dos próprios proprietários. Apesar disso, busca atender o mercado que se destina a usuários com baixo e médio nível de conhecimento em tecnologias, além de pequenos produtores, não tendo como foco o amparo à gestão financeira que é o objetivo desse trabalho.

3. PROCEDIMENTOS METODOLÓGICOS

Nesta seção será abordado os métodos de desenvolvimento e de análise da aplicação, bem como as ferramentas usadas em conjunto para garantir e auxiliar na produção que envolve diversos processos do trabalho (Desde a análise de requisitos até o desenvolvimento do código).

3.1 Metodologia de Pesquisa

Este trabalho se classifica como resumo de assunto quanto à sua natureza. “Os resumos de assunto buscam apenas sistematizar uma área de conhecimento, usualmente indicando sua evolução histórica e o estado da arte”. (WAZLAWICK, 2014, p. 21), visto que as pesquisas feitas em diversos sites, fóruns e artigos científicos compõem a maior parte do mesmo. Em relação aos objetivos é classificada como uma pesquisa descritiva e exploratória, as pesquisas descritivas e exploratórias podem ser consideradas um passo prévio para encontrar fenômenos não explicados pelas teorias vigentes segundo Wazlawick. E por fim é também classificada como uma pesquisa bibliográfica e documental, pois utiliza com base trabalhos anteriores de autores que são citados e referenciados, afim de resumir, aplicar e expandir os conhecimentos utilizados.

3.2 Ambiente de desenvolvimento

Como forma de introdução às ferramentas que foram utilizadas para desenvolvimento do MVP visto nessa aplicação, serão citados nos próximos tópicos um breve resumo de cada uma delas.

3.2.1 Android Studio

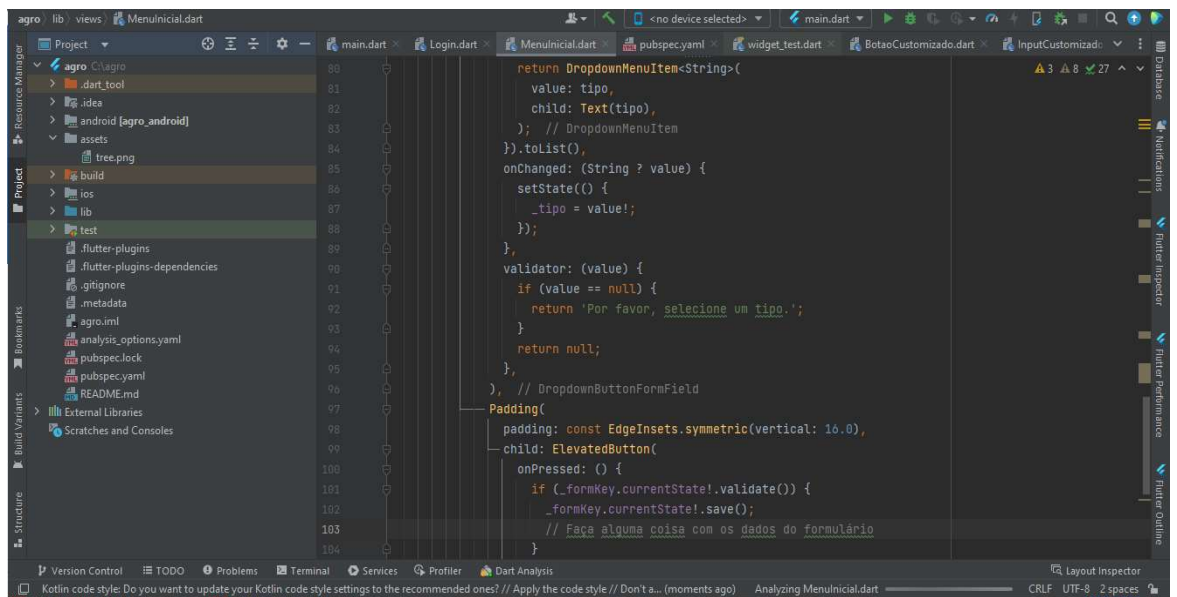
Foi utilizada a IDE Android Studio para desenvolvimento do código, um dos principais motivos para o uso foi a facilidade de uso e o fato de ser uma ferramenta atualizada e com muitos tutoriais disponíveis, além de ser gratuita.

O Android Studio é um ambiente de desenvolvimento integrado (IDE) amplamente utilizado para a criação de aplicativos Android. Desenvolvido pela Google, o Android Studio oferece uma variedade de recursos e ferramentas que auxiliam os desenvolvedores no processo de criação, depuração e teste de aplicativos para a plataforma Android. Com uma interface intuitiva e personalizável, o Android Studio permite uma programação eficiente e produtiva, oferecendo suporte a linguagens como Java e Kotlin. Além disso, a IDE fornece recursos

avançados, como emuladores de dispositivos, depuração em tempo real, gerenciamento de dependências e integração com o SDK do Android (SILVA, 2021).

A Figura 3 mostra a interface da IDE android studio, à esquerda a estrutura de pastas e na parte de cima os botões que indicam ações que podem ser feitas e algumas abas que estão abertas no exemplo

Figura 3 – Interface android studio



Fonte: Autoria própria

3.2.2 Dart e Flutter

A linguagem de programação utilizada foi o dart em conjunto com o framework flutter. Flutter oferece suporte tanto aos dispositivos que utilizam o sistema operacional android quanto aos iOS, economizando tempo no desenvolvimento e menos retrabalho em correções e melhorias futuras.

O Flutter é um framework de desenvolvimento de aplicativos móveis criado pelo Google, que tem se destacado no cenário tecnológico. Com sua abordagem inovadora e eficiente, o Flutter permite aos desenvolvedores criar aplicativos de alta qualidade para múltiplas plataformas, como Android e iOS, a partir de um único código-base. Sua arquitetura baseada em widgets facilita a construção de interfaces de usuário atraentes e responsivas, enquanto sua linguagem de programação, Dart, oferece uma sintaxe concisa e fácil de aprender. Além disso, o Flutter possui uma

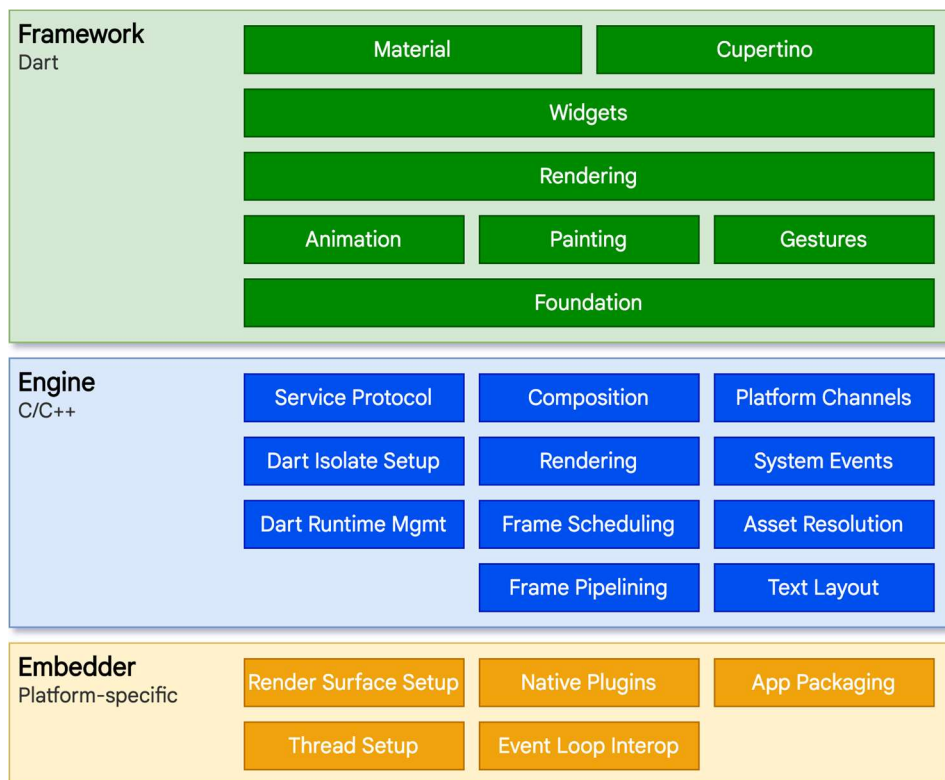
ampla gama de recursos e bibliotecas que agilizam o desenvolvimento e melhoram a produtividade (DOE, 2022).

A linguagem Dart é uma linguagem de programação moderna e robusta que desempenha um papel fundamental no desenvolvimento de aplicativos móveis com o framework Flutter. Criada pelo Google, a Dart apresenta uma sintaxe clara e concisa, facilitando a escrita de código legível e de fácil manutenção. Com recursos como tipagem estática opcional e coleta de lixo eficiente, a linguagem Dart oferece um desempenho ágil e confiável (Smith, 2020).

3.2.3 Visão geral da arquitetura do Flutter

Na Figura 4 é mostrado as camadas em que o flutter está dividido.

Figura 4 – Arquitetura do flutter



Fonte: Documentação do flutter

Na camada do framework, a mais alta da Figura 4, é mostrado os itens mais expostos de toda a arquitetura, em que são mais utilizados por

desenvolvedores, inclusive os mais utilizados, diretamente, para o desenvolvimento deste trabalho.

Material e Cupertino: são bibliotecas que oferecem conjuntos abrangentes de controles que usam a camada de widget primitivas de composição para implementar as linguagens de design Material ou iOS.

Widgets: são uma abstração de composição. Cada objeto de renderização na camada de renderização tem uma classe correspondente na camada de widgets. Além disso, a camada de widgets permite definir combinações de classes que você pode reutilizar.

Rendering: fornece uma abstração para lidar com layout. Com esta camada, você pode construir uma árvore de objetos renderizáveis e manipular esses objetos dinamicamente, com a árvore atualizando o layout.

Foundations: classes e serviços de blocos de construção, como animation, painting e gesture que oferecem abstrações.

O Flutter Engine é o coração do Flutter. É um motor de renderização que é responsável por desenhar a interface do usuário do aplicativo. Ele é construído em C++ e inclui o mecanismo de renderização Skia, que é usado para desenhar a interface do usuário de forma eficiente (DIO,2023).

O Flutter Embedder serve como uma ponte entre o mecanismo Flutter e a plataforma nativa, facilitando a interação da API do incorporador do mecanismo Flutter com o sistema host. Graças ao Flutter Embedder, os desenvolvedores podem escrever aplicativos Flutter com as mesmas funcionalidades flexíveis de design de tela e interface de usuário (UI) no Linux desktop ou em um sistema Linux incorporado, como normalmente fariam para aplicativos Android ou web (Sorathiya,2023).

3.2.4 Por que usar o Flutter?

Ao escolher o flutter como framework, o principal motivo foi a praticidade, uma ferramenta versátil capaz de unir várias plataformas em um único código, o fato de flutter dar suporte ao sistema android e ao ios (sem contar o suporte a web) faz dele uma ferramenta poderosa, que economiza tempo e isso foi primordial para que o flutter fosse o escolhido. Além disso, o flutter tem uma documentação de fácil acesso e extremamente simples, tanto na sua própria documentação quanto nos fóruns de perguntas, é essencial que se tenha essa acessibilidade à informação e à formação quando se trata de uma tecnologia que está em emersão.

3.2.5 Arquitetura MVVM

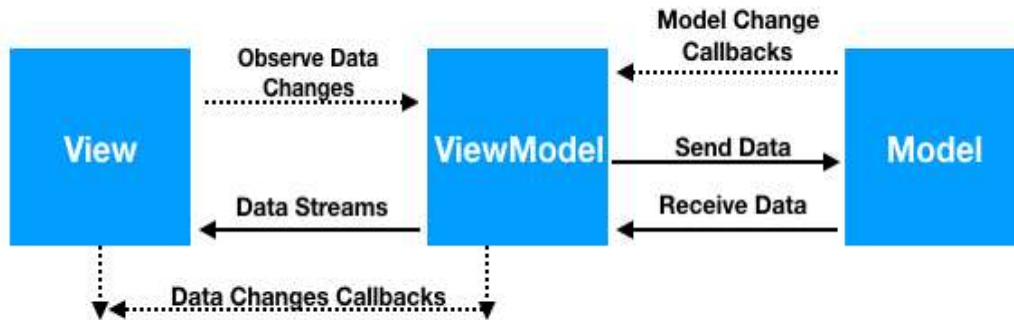
A arquitetura MVVM (Model-View-ViewModel) é um padrão de arquitetura de software que tem ganhado destaque no desenvolvimento de aplicativos, especialmente em aplicações móveis. O MVVM divide a aplicação em três componentes principais: o modelo (model), a visualização (view) e o modelo de visão (viewmodel). O modelo representa os dados e a lógica de negócios da aplicação, a visualização é responsável por exibir a interface do usuário e o modelo de visão atua como um intermediário entre o modelo e a visualização, fornecendo os dados necessários para a interface do usuário. Essa abordagem promove uma separação clara de responsabilidades e facilita a testabilidade, manutenção e evolução da aplicação (DOE, 2022).

A arquitetura MVVM foi extremamente importante ao funcionar como um padrão organizacional, induzindo sempre à um código limpo e organizado. A forma como os dados são modelados pela arquitetura acaba também refletindo na forma como ele é escrito e isso leva à construção de algo que apesar de simples funcione de uma forma visivelmente limpa em que possa ser posteriormente aplicado o princípio da escalabilidade de software, levando em conta a necessidade, caso houver.

Abaixo está a Figura 5, que é uma ilustração da arquitetura MVVM, e mostra como os dados e a interface são interpretados pela camada ViewModel, nessa camada é feito o direcionamento dos dados que serão destinados à camada de

dados (inseridos no sistema), e dos dados que serão mostrados ao usuário pela interface.

Figura 5 - Modelo de arquitetura MVVM.



Fonte: DigitalOcean, 2022.

3.2.6 SQLite

SQLite é uma biblioteca em processo que implementa um mecanismo de banco de dados SQL transacional independente, sem servidor e com configuração zero. O código do SQLite é de domínio público e, portanto, é gratuito para uso para qualquer finalidade, comercial ou privada. SQLite é um mecanismo de banco de dados SQL integrado. Ao contrário da maioria dos outros bancos de dados SQL, o SQLite não possui um processo de servidor separado. SQLite lê e grava diretamente em arquivos de disco comuns. Um banco de dados SQL completo com diversas tabelas, índices, gatilhos e visualizações está contido em um único arquivo de disco. O formato do arquivo do banco de dados é multiplataforma - você pode copiar livremente um banco de dados entre sistemas de 32 e 64 bits ou entre arquiteturas big-endian e little-endian (SQLite, 2023).

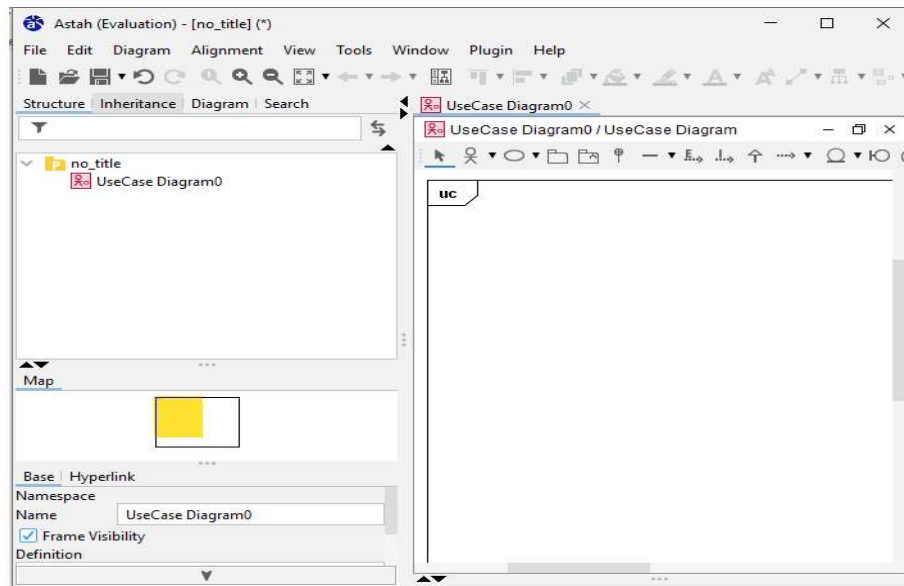
3.2.7 Astah UML

As ferramentas de modelagem do Astah permitem visualizar a essência de suas ideias e designs de software (Astah, 2023). Uma excelente ferramenta para

organizar e visualizar as partes do projeto, além de ser utilizada como forma de documentação.

A Figura 6 mostra a interface da ferramenta astah, que pode ser usada para diversos tipos de diagramas.

Figura 6 – Interface do astah uml



Fonte: Autoria própria

4. DESCRIÇÃO DA APLICAÇÃO

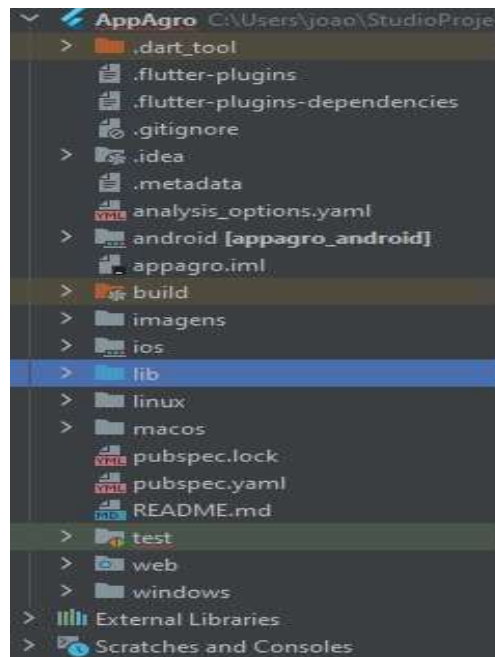
4.1 Escopo

O aplicativo foi pensado com o foco em pequenos produtores, justamente pela falta de contato com a tecnologia e com softwares que auxiliem de forma intuitiva a utilização de sistemas. Nesse sentido, a interface e construção dos formulários e listagens são simples, com campos enxutos e limpos, afim de proporcionar ao usuário algo funcional e de fácil uso.

Muitos sistemas desenvolvidos para o ramo agropecuário contêm diversas informações e até mesmo módulos que confundem visualmente o usuário (produtor) e dificultam o uso da ferramenta em virtude da complexidade apresentada na tela. Ao buscar algo mais objetivo e entregar uma aplicação que seja simples, porém, funcional, o usuário se sente incentivado, e são abertas portas para introdução a novas tecnologias.

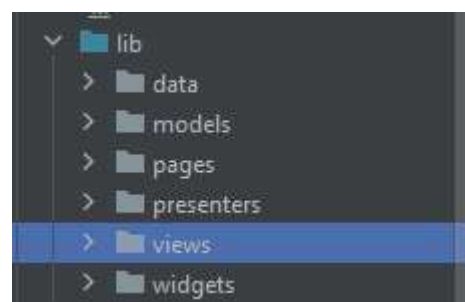
4.2 Estrutura de pastas da aplicação

A Figura 7 contém os diretórios do aplicativo de uma forma geral, são pastas que já estão contidas assim que o projeto é criado no android studio.

Figura 7 – Estrutura de pastas geral

Fonte: Autoria própria

Diferentemente da Figura 7, na Figura 8 estão todas as pastas do diretório lib, que representam as classes de dados, modelos, telas e componentes do projeto.

Figura 8 – Pastas do diretório lib

Fonte: Autoria própria

- Data: Responsável por instanciar o banco de dados SQLite, bem como executar as funções CRUD no banco.
- Models: Esta pasta guarda as classes da aplicação, no momento contém uma classe base e a classe lançamento que é o modelo de dado usado.
- Pages: Diretório que guarda as chamadas para as interfaces da aplicação, as quais tem a função de direcionar, e montar os componentes da aplicação.

- **Presenters:** O arquivo de apresentação, contido nesse diretório, é responsável por fazer a ligação dos dados com a view da aplicação funcionando como um intermediador, que leva e traz informações e dados entre os dados e a interface.
- **View:** São as estruturas visuais, que são preenchidas e mostradas na tela para a comunicação com o usuário.
- **Widgets:** Pasta que guarda os componentes criados manualmente, diferente dos que já existem para o uso dentro do flutter, esses são componentes que podem ser personalizados e modificados conforme a necessidade do projeto

4.3 Requisitos

Neste tópico serão descritos os requisitos funcionais e não funcionais da aplicação (CRUD) implementada.

4.4 Requisitos funcionais

Os requisitos funcionais podem ser observados na Tabela 2, que além de representas os 4 requisitos funcionais desse CRUD (Cadastrar, Consultar, Alterar e Remover), também representa o caso de uso específico “Manter fluxo de caixa”.

Tabela 2 – Tabela Manter fluxo de caixa

	Manter fluxo de caixa
RF01	Cadastrar lançamento
RF02	Consultar lançamento
RF03	Alterar lançamento
RF04	Remover lançamento

Fonte: Autoria própria

4.5 Requisitos não funcionais

Requisitos não funcionais. São restrições aos serviços ou funções oferecidas pelo sistema. Incluem restrições de timing, restrições no processo de

desenvolvimento e restrições impostas pelas normas. Ao contrário das características individuais ou serviços do sistema, os requisitos não funcionais, muitas vezes, aplicam-se ao sistema como um todo (Sommerville, 2011).

Os requisitos não funcionais dessa aplicação a princípio são: Compatibilidade do aplicativo essencialmente no sistema operacional Android, e funcionar sem acesso à internet. A explicação por trás dessas exigências é o fato do aplicativo focar em acessibilidade, geralmente pequenos produtores utilizam celulares com sistema android devido ao baixo custo e nem todos tem o acesso à internet ou tem um acesso limitado.

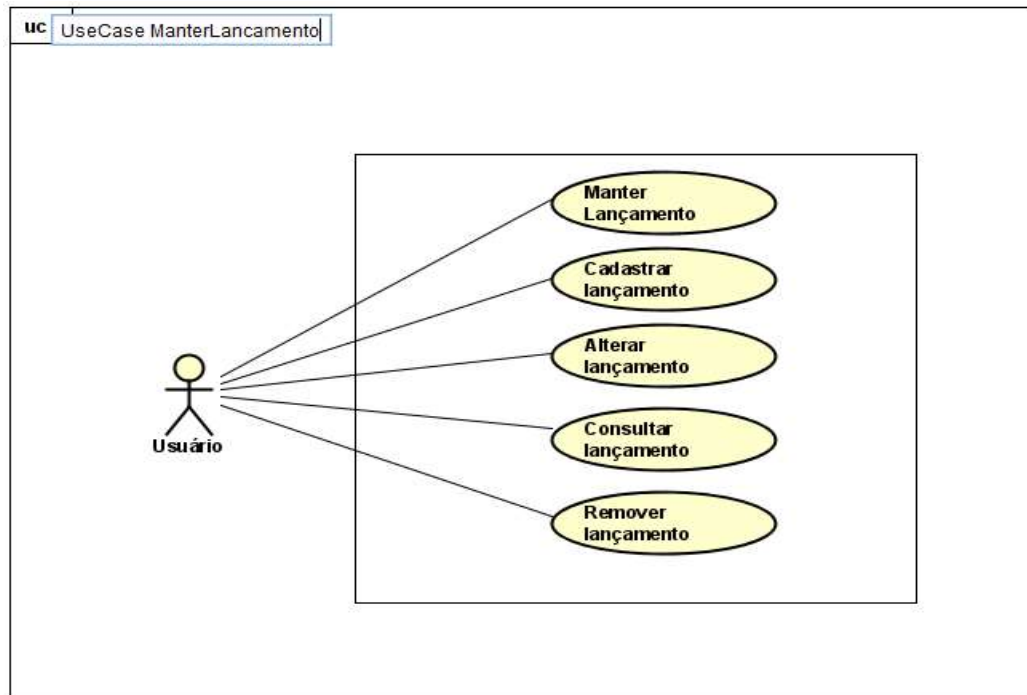
4.6 Diagrama de caso de uso

Os casos de uso são documentados por um diagrama de casos de uso de alto nível. O conjunto de casos de uso representa todas as possíveis interações que serão descritas nos requisitos de sistema. Atores, que podem ser pessoas ou outros sistemas, são representados como figuras 'palito'. Cada classe de interação é representada por uma elipse. Linhas fazem a ligação entre os atores e a interação (Sommerville, 2011).

4.6.1 Diagrama Geral de Caso de Uso

Na Figura 9 está inserido o diagrama geral de caso de uso, criado com o software astah. O diagrama representado descreve as ações da aplicação.

Figura 9 – Diagrama geral de caso de uso (astah)



Fonte: Autoria própria

4.7 Especificação do caso de uso manter fluxo de caixa

Este tópico tem como finalidade descrever os requisitos funcionais relacionados ao fluxo de caixa.

[RF01] Cadastrar lançamento

Ator: Usuário

Prioridade: Essencial

Entradas e pré-condições: Clicar no botão com ícone '+' na barra principal.

Saídas e pós-condições: Após inserir o lançamento e clicar em adicionar o aplicativo será redirecionado à tela de listagem (home).

[RF02] Consultar lançamentos

Ator: Usuário

Prioridade: Essencial

Entradas e pré-condições: É a primeira tela após a tela de login, sendo o home da aplicação, onde está contida a listagem de todos os itens.

Saídas e pós-condições: Não se aplica, pois, a listagem está na tela principal, mostrando todos os lançamentos.

[RF03] Alterar lançamento

Ator: Usuário

Prioridade: Essencial

Entradas e pré-condições: Clicar no botão com ícone de edição, formulário será mostrado, informar as alterações nos campos do formulário, clicar em editar.

Saídas e pós-condições: Redirecionado a listagem de lançamentos.

[RF04] Remover lançamento

Ator: Usuário

Prioridade: Essencial

Entradas e pré-condições: Clicar no botão com ícone de lixeira.

Saídas e pós-condições: Continua na tela de listagem de lançamentos, porém, sem o lançamento que já foi excluído.

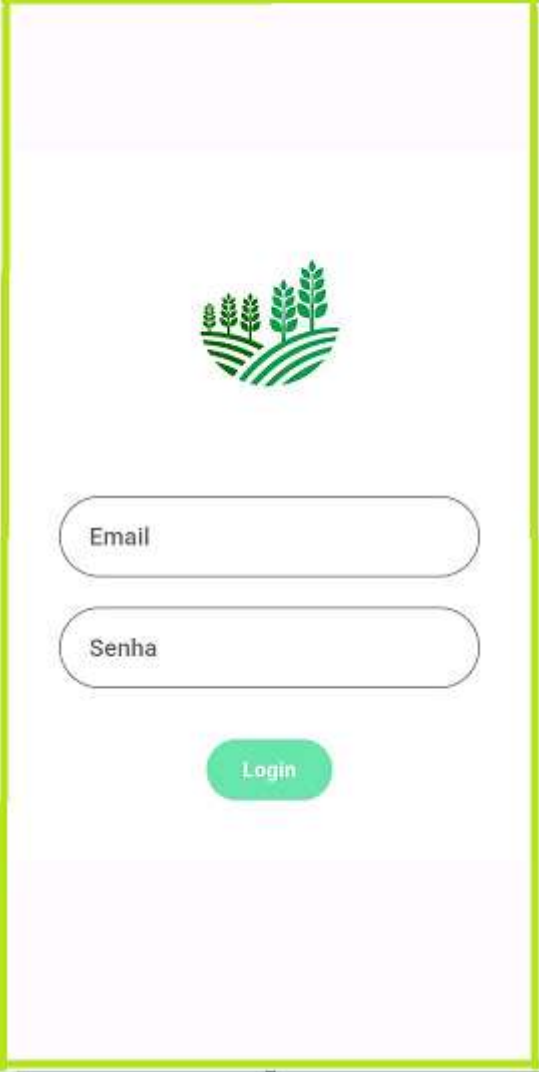
4.8 Interface

O CRUD criado tem 3 telas, que servem de interface para a interação do usuário, além disso foi feita uma tela de login.

4.8.1 Tela de login

A tela de login mostrada na Figura 10, é composta de uma logo simbolizando o agronegócio e as cores também foram pensadas valorizando o verde, pois passa um ar de equilíbrio e harmonia.

Figura 10 – Tela de login



A tela de login apresenta um design limpo com uma barra decorativa superior e inferior em tom de rosa claro. No centro, há um ícone verde que representa plantas e colheita. Abaixo do ícone, existem dois campos de entrada de texto arredondados: o primeiro rotulado 'Email' e o segundo rotulado 'Senha'. Abaixo dos campos, há um botão verde arredondado com o texto 'Login' em branco.

Fonte: Autoria própria

4.8.2 Listagem de lançamentos

A tela de listagem mostrada na Figura 11 é composta por uma barra inicial com o título do módulo lançamento e um botão para adicionar novo lançamento, abaixo da barra cada item é um lançamento que foi adicionado anteriormente.

Figura 11 – Listagem de lançamentos

Fonte: Autoria própria

4.8.3 Formulário novo lançamento

Após clicar no botão '+', mostrado na Figura 11, o usuário é direcionado para o modal de novo lançamento (Figura 12), composto pelos campos tipo, categoria, valor e descrição. Além disso possui o botão para inserir o novo lançamento.

Figura 12 – Formulário de novo lançamento



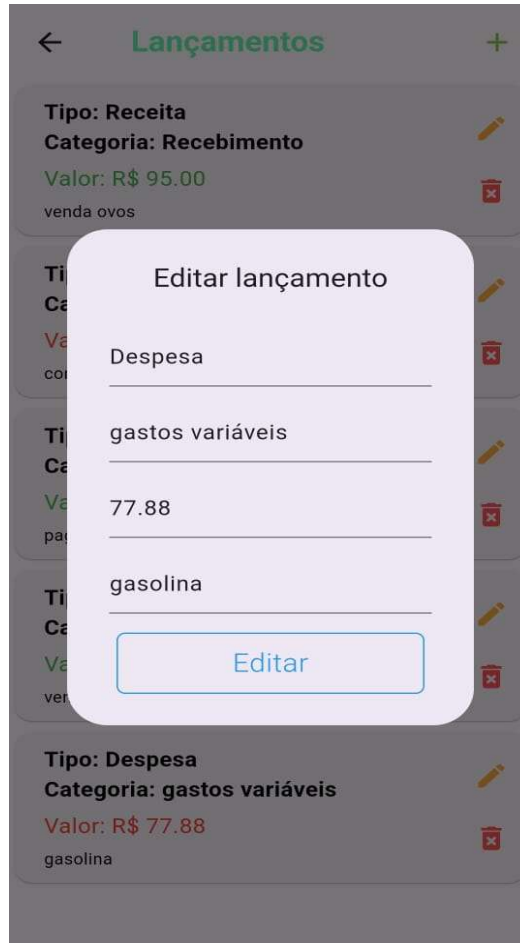
The image shows a mobile application interface for managing financial transactions. The background is a list of transactions under the heading 'Lançamentos'. The first entry is a receipt: 'Tipo: Receita', 'Categoria: Recebimento', 'Valor: R\$ 95.00', and 'venda ovos'. The second entry is a expense: 'Tipo: Despesa', 'Categoria: gastos variáveis', 'Valor: R\$ 77.88', and 'gasolina'. A modal form titled 'Novo lançamento' is overlaid on the screen. It contains four input fields: 'Insira o Tipo', 'Insira a categoria', 'Insira o valor', and 'Descreva o lançamento'. At the bottom of the modal is a blue button labeled 'Adicionar'.

Fonte: Autoria própria

4.8.4 Formulário de edição de lançamento

A Figura 13, semelhante à Figura 12, porém, com a funcionalidade de editar, traz os campos preenchidos com os valores referentes ao lançamento selecionado, podendo ser alterados e salvo ao clicar no botão 'editar' do formulário.

Figura 13 – Formulário para editar lançamento



Fonte: Autoria própria

5. CONCLUSÃO

De acordo com os conteúdos analisados, resumidos e aplicados é visto uma crescente demanda de aplicativos para suprir necessidades e automatizar tarefas, seja no agronegócio ou em quaisquer outros ramos a tecnologia tem um efeito positivo e veio para melhorar a qualidade de vida, otimizar o tempo e facilitar o trabalho do ser humano.

Com o uso de uma tecnologia como a que foi desenvolvida por meio desse trabalho, é possível integrar produtores ao mundo da tecnologia e ampliar seus pontos de vista perante algo que foi gerado algum tipo de preconceito, simplesmente por falta de contato e medo de experimentar algo novo.

É necessária paciência e muito trabalho duro para verem as coisas acontecerem, e com a tecnologia não é diferente, pelo menos não nesse caso, e o flutter com sua modularidade e facilidade de acesso representa muito isso, o flutter representa algo que é simples, mas ao mesmo tempo muito potente, de fácil aprendizagem, que se disseminou de maneira rápida e vem ganhando cada vez mais seu espaço no mercado.

Portanto, foi finalizado com êxito o que foi proposto neste trabalho, além da entrega de conteúdo, e informações foi demonstrado o funcionamento do software, juntamente com as tecnologias necessárias para execução do CRUD.

6. TRABALHOS FUTUROS

Diversas sugestões de produtores serão pontuadas ao executar testes e efetuar entrevistas com os mesmos, anotando em detalhes as carências e dores no que diz respeito à gestão e organização rural, também serão feitas de uma forma bem dinâmica, para que não se torne algo entediante.

Dando continuidade, a ideia é criar novos módulos, vinculados à cada processo dentro da propriedade rural, e aplicar todos os conceitos que foram anteriormente aplicados no módulo de lançamentos financeiro, desde a parte de requisitos aos passos tomados no desenvolvimento.

Outras melhorias que devem ser identificadas como validações minuciosas nos campos dos formulários, além de estabelecer os campos adequados e melhorar a interface, para que o usuário faça passos cada vez mais automáticos, induzindo-o e tornando o uso cada vez mais simples.

7. REFERÊNCIAS

A melhor ferramenta de diagramação UML. Disponível em: <https://astah.net/products/astah-uml/>. Acesso em: 25 nov. 2023.

Agricultura Digital: quais as dificuldades para a implementação no agronegócio. 2018. Disponível em: <https://www.vdibrasil.com/agricultura-digital-quais-as-dificuldades-para-a-implementacao-no-agronegocio/>. Acesso em: 15 ago. 2023.

ATAIDES, Guilherme Matos. **Análise e desenvolvimento de um sistema offline para gestão de pequenas propriedades rurais de Iporá e região com ênfase nos processos produtivos e reprodutivos.** 2021. 105 f. TCC (Graduação) - Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Instituto Federal Goiano – Campus Iporá, Iporá, 2021.

BORGES, Marcio Silva; GUEDES, Cezar Augusto Miranda; CASTRO, Maria Cristina Drumond e. **Agric Programa de assistência técnica para o desenvolvimento de pequenas propriedades leiteiras em Valença-RJ e região Sul Fluminense ultura digital: pesquisa, desenvolvimento e inovação nas cadeias produtivas.** Scielo, Rio de Janeiro, v. 24, n. 24, p. 1-24, nov. 2015.

DOE, John. **Flutter: A Comprehensive Platform for Mobile and Web App Development.** Mobile Development Review, v. 16, n. 3, p. 78-95, 2022.

DOE, John. **Flutter: A Comprehensive Framework for Cross-Platform Mobile App Development.** Mobile ApplicationDevelopment Review, v. 15, n. 2, p. 45-60, 2021.

HARAKAWA, Felipe Seidi; PEREIRA, Adriana Soares. **Desenvolvimento e Implementação do Aplicativo Mobile Blog de Todas.** 2020. 20 v. TCC (Graduação) - Curso de Bacharelado em Sistemas de Informação, Universidade Federal de Santa Maria (Ufsm), Frederico Westphalen, 2020.

IBGE. **População ocupada nos estabelecimentos agropecuários cai.** Disponível em: <https://censoagro2017.ibge.gov.br/2013-agencia-de-noticias/releases/25789-censo-agro-2017-populacao-ocupada-nos-estabelecimentos-agropecuarios-cai-8-8.html#:~:text=No%20Censo%20Agropecu%C3%A1rio%20de%202017,estabelecimentos%20agropecu%C3%A1rios%20levantados%20pelo%20censo..> Acesso em: 14 out. 2023.

KAY, Ronald D.; EDWARDS, William M.; DUFFY, Patrícia A.. **Gestão de Propriedades Rurais.** 7. ed. Porto Alegre: Bookman, 2014. 452 p.

Pesquisa descritiva, exploratória e explicativa. Disponível em: <https://www.diferenca.com/pesquisa-descritiva-exploratoria-e-explicativa/#:~:text=Na%20pesquisa%20explorat%C3%B3ria%20o%20objetivo,para%20compreender%20causas%20e%20efeitos..> Acesso em: 20 ago. 2023.

SANTIAGO, Marcos Rogerio. **Ensaio do SWEBOK – Software Engineering Body Of Knowledge**. 2011. 243 f. TCC (Graduação) - Curso de Gestão de Tecnologia da Informação, Universidade Gama Filho, Goiânia, 2011.

SILVA, Maria. Android Studio: A Comprehensive IDE for Android App Development. *Mobile Development Review*, v. 15, n. 4, p. 67-82, 2021.

SMITH, Jane. **Dart Programming Language: A Comprehensive Overview**. *Programming Languages Review*, v. 10, n. 4, p. 22-38, 2020.

SOBRE SQLite. Disponível em: <https://www.sqlite.org/about.html>. Acesso em: 15 nov. 2023.

SOCIEDADE NACIONAL DE AGRICULTURA. **Como a tecnologia está revolucionando o agronegócio**. 2019. Disponível em: <https://www.sna.agr.br/como-a-tecnologia-estarevolucionando-o-agronegocio/>. Acesso em: 22 de ago. 2023.

SOCIEDADE NACIONAL DE AGRICULTURA. **Como a tecnologia está revolucionando o agronegócio**, 2019. Disponível em: <https://www.sna.agr.br/como-a-tecnologia-estarevolucionando-o-agronegocio/>. Acesso em: 27 de ago. 2023.

SOMMERVILLE, Ian. **Engenharia de software**. 9. ed. São Paulo: Pearson, 2011. 544 p.

WAZLAWICK, Raul Sidnei. **Metodologia de Pesquisa para Ciência da Computação**. 2. ed. Rio de Janeiro: Elsevier, 2014. 153 p.

DIGITALOCEAN. **Padrão de design Android MVVM**. 2022. Disponível em: <https://www.digitalocean.com/community/tutorials/android-mvvm-design-pattern>. Acesso em: 16 dez. 2023.

MELO, Rubens de. **Flutter para iniciantes**. 2019. Disponível em: <https://speakerdeck.com/rubensdemelo/flutter-para-iniciantes>. Acesso em: 16 dez. 2023.

DIO. **Descubra Flutter: o poderoso framework da Google**. 2023. Disponível em: <https://www.dio.me/articles/descubra-flutter-o-poderoso-framework-da-google>. Acesso em: 16 dez. 2023.

SORATHIYA, Nidhi. **O potencial oculto do Flutter Embedder: revelando suas vantagens**. 2023. Disponível em: <https://www.dhiwise.com/post/the-hidden-potential-of-flutter-embedder-its-advantages>. Acesso em: 16 dez. 2023.

