

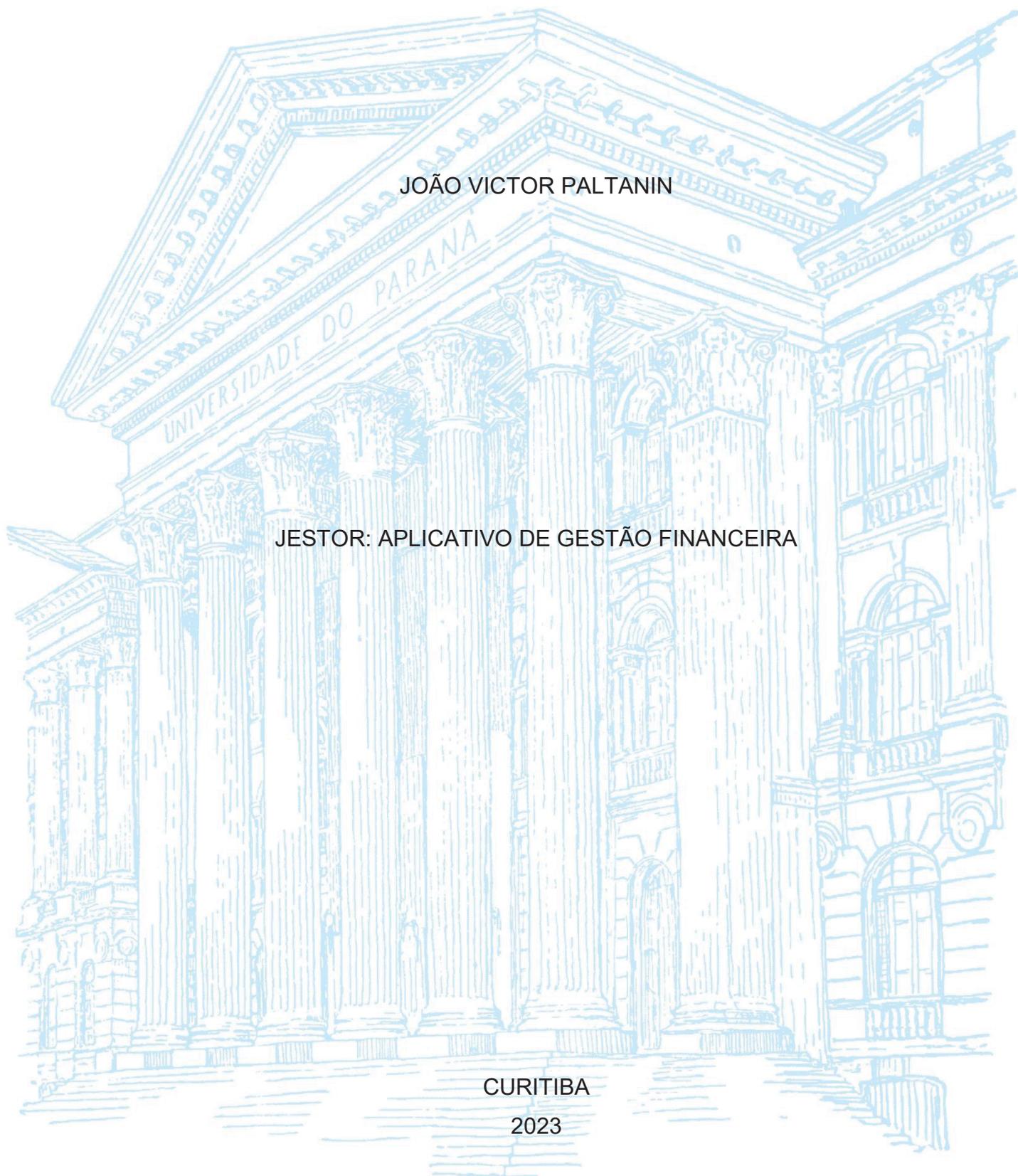
UNIVERSIDADE FEDERAL DO PARANÁ

JOÃO VICTOR PALTANIN

JESTOR: APLICATIVO DE GESTÃO FINANCEIRA

CURITIBA

2023



JOÃO VICTOR PALTANIN

JESTOR: APLICATIVO DE GESTÃO FINANCEIRA

Trabalho de Conclusão de Curso apresentado ao curso de Pós-Graduação em Desenvolvimento Ágil de Software, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, como requisito parcial à obtenção do título Especialista em Desenvolvimento Ágil de Software.

Orientadora: Profa. Dra. Rafaela Mantovani Fontana

CURITIBA

2023

## TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação DESENVOLVIMENTO ÁGIL DE SOFTWARE da Universidade Federal do Paraná foram convocados para realizar a arguição da Monografia de Especialização de **JOÃO VICTOR PALTANIN** intitulada: **JESTOR: APLICATIVO DE GESTÃO FINANCEIRA**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua aprovação no rito de defesa.

A outorga do título de especialista está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 27 de Outubro de 2023.



RAFAELA MANTOVANI FONTANA  
Presidente da Banca Examinadora



JAIME WOJCIECHOWSKI  
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Eu dedico esse trabalho à minha família e amigos, assim como a todos os professores que me ajudaram nessa jornada. Sobretudo, dedico à minha grande companheira Elisamary.

"A sabedoria oferece proteção, como o faz o dinheiro, mas a vantagem do conhecimento é esta: a sabedoria preserva a vida de quem a possui." (BÍBLIA, Eclesiastes, 7, 12).

## RESUMO

A vida financeira dos brasileiros é um tema delicado. Nos últimos anos grande parte da população está em situação de endividamento devido às desigualdades econômicas e sociais. Apesar dos avanços significativos no acesso à informação e à educação financeira, muitos brasileiros ainda enfrentam dificuldades para administrar o seu dinheiro. O desemprego, os altos impostos, a inflação e a baixa remuneração são alguns dos principais desafios que os brasileiros enfrentam. Além disso, a falta de educação financeira e de planejamento adequado contribuem para o endividamento e a dificuldade de poupar dinheiro. Neste cenário, é evidente a importância da tecnologia, principalmente com a facilidade do acesso à internet e popularização dos aparelhos celulares, para ampliação do conhecimento. Em vista disso, o propósito deste trabalho é a criação de um aplicativo de gestão financeira pessoal, para que os usuários possam lançar suas receitas e despesas, e também, ter maior controle sobre o próprio dinheiro. Para a modelagem da aplicação foram utilizados Histórias de Usuário, Diagrama de Classe, Diagrama de Casos de Uso e Diagrama de Sequência. Na criação do protótipo de telas foram utilizados conceitos de *UserExperience* e *UserInterface* juntamente com a ferramenta Figma. O processo de desenvolvimento do aplicativo se obteve com o planejamento de releases e a utilização do Trello como ferramenta de acompanhamento das atividades através de quadros. No *back-end*, foi desenvolvido uma API utilizando o *framework* SpringBoot que tem como base a linguagem de programação Java em conjunto com o *framework* OAuth2 que adiciona uma camada a mais de segurança. A API desenvolvida consulta e envia dados para um banco de dados. No desenvolvimento do *front-end* foi utilizado o *framework* Flutter pela facilidade de implementação em várias plataformas (Android, IOS, Windows, etc) e similaridade do Dart com o Java o que proporcionou otimização do processo.

Palavras-chave: Vida financeira; Endividamento; Educação financeira; Gestão financeira; Aplicativo *mobile*.

## **ABSTRACT**

Brazilians' financial life is a sensitive subject. Due to economic and social imbalances, a considerable portion of the population has been in debt in recent years. Despite major gains in information accessibility and financial education, many Brazilians continue to struggle with money management. Unemployment, high taxes, inflation, and low wages are just a few of the issues that Brazilians face. Furthermore, a lack of financial education and preparation contributes to debt and makes it difficult to save money. In this circumstance, the relevance of technology for knowledge development is clear, especially with the ease of access to the internet and the popularity of cell phones. In light of this, the goal of this project is to develop a personal financial management application (app) that allows users to enter their income and expenses while also giving them more control over their money. User Stories, Class Diagrams, Use Case Diagrams, and Sequence Diagrams were used to prepare the application. The Figma tool was used in conjunction with the UserExperience and UserInterface principles to create the prototype screens. The application development process was completed through Release Management and the use of Trello as a tool to track activity via boards. On the back end, an API was created by combining the SpringBoot framework, which is based on the Java programming language, with the OAuth2 framework, which offers another level of security. The API that was developed searches and transmits data to a database. The Flutter framework was adopted to build the front end given its simplicity to implement on different operating systems (Android, iOS, Windows, etc.), and the similarity of Dart with Java, which provided process optimization.

Keywords: Financial life; debt; financial education; financial management; mobile app.

## LISTA DE FIGURAS

FIGURA 1 - VERSÃO INICIAL DA ARQUITETURA DO SISTEMA.....	19
FIGURA 2 - SEQUÊNCIA DO FLUXO DE AUTENTICAÇÃO DO USUÁRIO .....	20
FIGURA 3 – PADRÃO DE CORES DO PROJETO .....	24
FIGURA 4 - PADRÃO DE ÍCONE E CORES DAS CATEGORIAS.....	25
FIGURA 5 - PROTÓTIPO INICIAL .....	26
FIGURA 6 - PROTÓTIPO FINAL .....	26
FIGURA 7 - QUADRO DE RELEASES .....	27
FIGURA 8 - DIAGRAMA DE ATIVIDADES DO PROCESSO MACRO DO APLICATIVO .....	29
FIGURA 9 - DIAGRAMA DE CASOS DE USO .....	30
FIGURA 10 - TELA DE CADASTRO DE USUÁRIO.....	32
FIGURA 11 - TELAS DE EDIÇÃO E EXCLUSÃO DE USUÁRIO.....	36
FIGURA 12 - TELA DE CADASTRO DE LANÇAMENTO .....	38
FIGURA 13 - MAPA MENTAL DAS CATEGORIAS DE ENTRADAS E SAÍDAS .....	40
FIGURA 14 - TELA DE LOGIN.....	44
FIGURA 15 - TELA DE LISTAGEM DE ENTRADAS .....	47
FIGURA 16 - TELA DE LISTAGEM DE SAÍDAS.....	49
FIGURA 17 - TELA DA MODAL DE DETALHE DE LANÇAMENTO .....	51
FIGURA 18 - TELA DE PEQUISA DE LANÇAMENTOS.....	53
FIGURA 19 - TELA DE SELEÇÃO DE MÊS E ANO .....	55
FIGURA 20 - DIAGRAMA DE CLASSES .....	57
FIGURA 21 - DIAGRAMA DE SEQUÊNCIA CADASTRAR LANÇAMENTO .....	58
FIGURA 22 - DIAGRAMA DE SEQUÊNCIA CADASTRAR USUÁRIO .....	59
FIGURA 23 - DIAGRAMA DE SEQUÊNCIA EDITAR LANÇAMENTO.....	60
FIGURA 24 - DIAGRAMA DE SEQUÊNCIA EDITAR SENHA DO USUÁRIO.....	61
FIGURA 25 - DIAGRAMA DE SEQUÊNCIA EXCLUIR LANÇAMENTO .....	62
FIGURA 26 - DIAGRAMA DE SEQUÊNCIA EXCLUIR USUÁRIO .....	63
FIGURA 27 - DIAGRAMA DE SEQUÊNCIA REALIZAR LOGIN .....	64
FIGURA 28 - DIAGRAMA FÍSICO DO BANCO DE DADOS .....	65

## LISTA DE TABELAS

TABELA 1 - INCONSISTÊNCIAS NO CADASTRO DE USUÁRIO .....	34
TABELA 2 - VALIDAÇÃO E-MAIL .....	35
TABELA 3 - INCONSISTÊNCIAS NO CADASTRO DE LANÇAMENTO .....	40
TABELA 4 - INCONSISTÊNCIAS NO LOGIN .....	46

## SUMÁRIO

<b>1 VISÃO GERAL DO PROJETO .....</b>	<b>16</b>
1.1 CONTEXTUALIZAÇÃO .....	16
1.2 DESCRIÇÃO DO PRODUTO DO PROJETO.....	18
1.3 FUNCIONALIDADES .....	18
1.4 TECNOLOGIAS UTILIZADAS .....	19
1.4.1 OAuth2 .....	21
1.4.2 Java.....	21
1.4.3 Spring Boot.....	22
1.4.4 Dart.....	22
1.4.5 Flutter .....	23
1.4.6 MySQL .....	23
1.4.7 Figma .....	24
1.4.8 Trello .....	27
1.4.9 Hardware.....	27
<b>2 DESENHO DO PROCESSO MACRO.....</b>	<b>29</b>
<b>3 DIAGRAMA DE CASOS DE USO .....</b>	<b>30</b>
<b>4 HISTÓRIAS DE USUÁRIO .....</b>	<b>31</b>
<b>5 DIAGRAMA DE CLASSES.....</b>	<b>57</b>
<b>6 DIAGRAMAS DE SEQUÊNCIA .....</b>	<b>58</b>
<b>7 DIAGRAMA FÍSICO DO BANCO DE DADOS .....</b>	<b>65</b>
<b>REFERÊNCIAS.....</b>	<b>66</b>

## 1 VISÃO GERAL DO PROJETO

### 1.1 CONTEXTUALIZAÇÃO

Com a estabilidade que o Plano Real firmou, desenvolveu-se uma cultura de consumo e de não acúmulo de recursos financeiros. A estabilidade econômica proporcionou a expansão do mercado de trabalho formal, o aumento dos salários, a ampliação de políticas de bem-estar e a expansão do crédito ao consumidor. Pela primeira vez tornou-se possível a realização de hipotecas imobiliárias até empréstimos para a compra de automóveis. Além disso, todas essas mudanças refletiram no declínio dos altos níveis de desigualdade de renda (LUNA e KLEIN, 2016).

De acordo com Jaguaribe (1997, p. 123), antes do Plano Real, a inflação no Brasil atingia taxas absurdas, tornando impossível para as pessoas planejarem seus gastos e investimentos a longo prazo. Muitas vezes, as pessoas tinham que gastar todo o dinheiro que tinham imediatamente, antes que o valor do dinheiro diminuísse ainda mais. Esta instabilidade financeira afetou negativamente a vida das pessoas, tornando difícil para elas estabelecerem um planejamento financeiro adequado.

Para alcançar objetivos financeiros, é fundamental ter uma boa organização financeira pessoal. Isso inclui saber exatamente quanto se ganha, quanto se gasta e quanto sobra no final do mês. De acordo com a Pesquisa de Endividamento e Inadimplência do Consumidor (Peic) realizada pela Confederação Nacional do Comércio de Bens, Serviços e Turismo - CNC em dezembro de 2019, o número de famílias endividadas aumentou para 65,6%, a maior taxa desde 2010 (CNC, 2019). Isso destaca a importância crescente de gerenciar cuidadosamente as finanças pessoais (CAMPOS, 2020).

No relatório nacional divulgado pela ANBIMA (Associação Brasileira das Entidades dos Mercados Financeiros e de Capitais) foram apresentados os resultados de uma pesquisa realizada com brasileiros de 152 municípios. Do total de 3374 brasileiros, 40% afirmaram não ter economizado por considerar que todo o valor ganho é destinado a pagar as contas do mês (ANBIMA, 2017). Ademais, constatou-se que há uma grande falta de conhecimento dos brasileiros em relação a finanças, já que somente 42% dos entrevistados afirmaram ter algum tipo de saldo aplicado em investimentos (ANBIMA, 2017).

O assunto das finanças pessoais é muito abordado na mídia nacional, no entanto, o tema ainda não é tratado com devida importância por grande parte dos brasileiros. Apesar disso, surge um fenômeno inaugural sobre o assunto por meio de mídias digitais. Alguns influenciadores digitais tentam desconstruir conceitos culturais de aversão ao investimento. Segundo uma pesquisa elaborada pelo portal de notícias Infomoney, o perfil de Thiago Nigro, um influenciador financeiro, teve um crescimento de 88,7% nos últimos 12 meses, alcançando 1,66 milhão de seguidores (INFOMONEY, 2019).

Com o crescimento do mundo digital e a popularidade dos aplicativos móveis, as plataformas digitais também se tornaram intermediárias na gestão financeira. Antes, esse papel era restrito aos grandes bancos. Os aplicativos para dispositivos móveis e a tecnologia têm desempenhado um papel fundamental na gestão financeira pessoal nos últimos anos. De acordo com um estudo da consultoria Accenture, em 2019, 62% dos brasileiros afirmaram utilizar algum tipo de aplicativo para gerenciar suas finanças (Accenture, 2019).

No entanto, é importante ressaltar que o uso dessas ferramentas deve ser acompanhado de uma educação financeira adequada. Um estudo da ANBIMA (Associação Brasileira das Entidades dos Mercados Financeiros e de Capitais) apontou que a falta de conhecimento financeiro ainda é um grande desafio para os brasileiros, com apenas 31% dos entrevistados afirmando ter algum conhecimento em finanças (ANBIMA, 2021).

É perceptível a importância de uma plataforma que auxilia pessoas a gerir, controlar e manter com mais facilidade o seu dinheiro. Uma pesquisa do Banco Interamericano de Desenvolvimento (BID) revelou que o uso de tecnologia para gestão financeira pessoal pode ter impactos positivos na vida financeira das pessoas. Segundo o estudo, os usuários de aplicativos financeiros tiveram um aumento médio de 13% em suas economias em relação aos não usuários (BID, 2019).

Diante desse contexto e os desafios que existem acerca do tema, o objetivo principal deste trabalho é desenvolver um aplicativo para *smartphones* que permita o lançamento de receitas e despesas, a consulta e a classificação em categorias. Dessa maneira, é evidente o potencial da informação e do mundo digital, na vida dos usuários que usufruem dos aplicativos móveis.

## 1.2 DESCRIÇÃO DO PRODUTO DO PROJETO

O produto deste projeto é um aplicativo *mobile* de gestão financeira pessoal que permite aos usuários controlarem seu orçamento, registrar suas receitas e despesas, e categorizá-las para melhor visualização e análise. O objetivo principal do aplicativo é auxiliar os usuários a gerenciar suas finanças de maneira mais eficaz e consciente.

A interface do aplicativo permite que os usuários adicionem rapidamente entradas ou saídas, bem como filtrar por descrição e visualizar os seus lançamentos separando por mês e ano. Além disso, o aplicativo fornece um gráfico de gastos por categoria e a visualização do saldo total do mês para ajudar os usuários a entenderem melhor suas finanças e tomar decisões informadas.

Uma das principais vantagens do produto final é a possibilidade de categorizar as entradas e saídas, permitindo que os usuários agrupem seus lançamentos de acordo com suas próprias necessidades e preferências. Por exemplo, é possível categorizar as saídas como alimentação, transporte, lazer, etc.

Em resumo, o aplicativo *mobile* Jestor permite que os usuários tenham um controle mais eficiente de suas finanças, ao registrar suas entradas e saídas de forma organizada, categorizando e fornecendo informações úteis sobre seus gastos e ganhos em diferentes períodos de tempo.

## 1.3 FUNCIONALIDADES

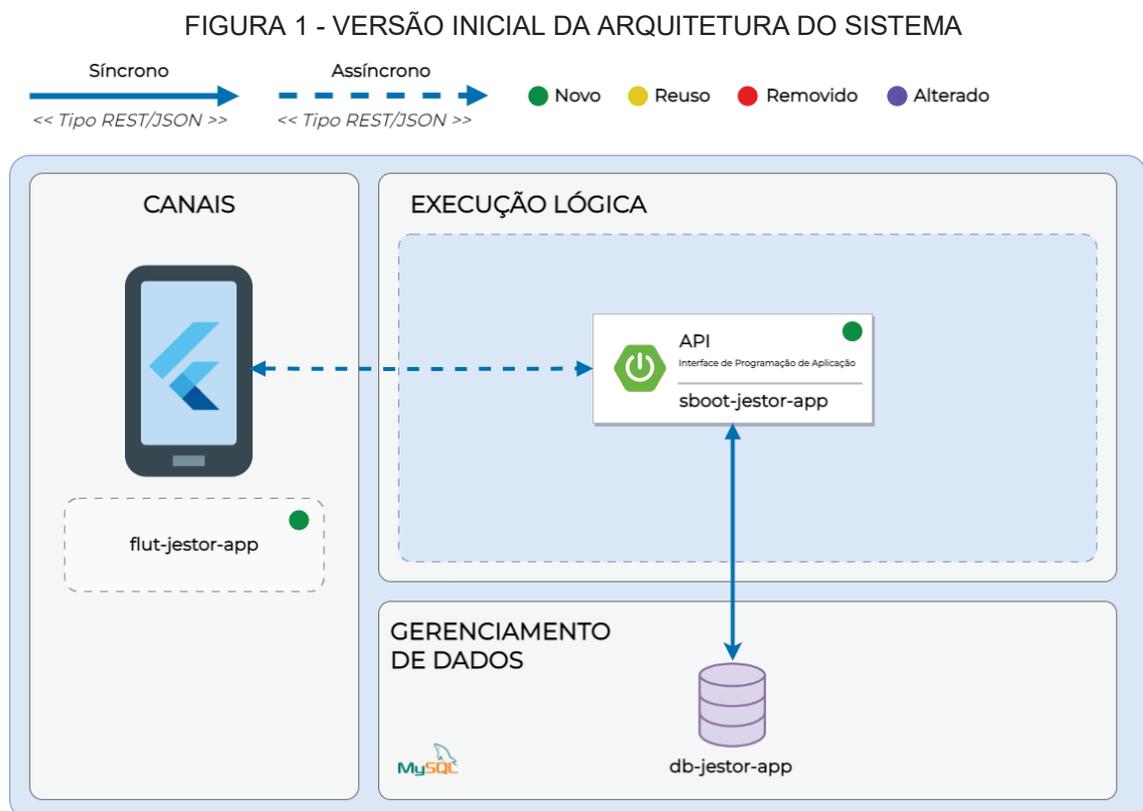
- O usuário cadastra suas entradas ou saídas informando o valor, a data, a descrição e a categoria;
- O usuário categoriza suas entradas como renda, bônus, vendas, rendimentos ou outras entradas;
- O usuário categoriza suas saídas como moradia, saúde, educação, lazer, transporte, alimentação, compras, pet ou outras saídas;
- O aplicativo permite que o usuário visualize suas entradas e saídas separando por mês e ano;
- O usuário do aplicativo pode filtrar suas entradas e saídas pela descrição do lançamento;

- O aplicativo apresenta um gráfico de gastos (saídas) por categoria para ajudar o usuário a entender melhor suas finanças e tomar decisões informadas.

#### 1.4 TECNOLOGIAS UTILIZADAS

A aplicação proposta é baseada em uma arquitetura (FIGURA 1) composta por *front-end* e *back-end*. O *front-end* consiste em um aplicativo móvel construído em Flutter. O *back-end*, composto por uma API REST em SpringBoot e um banco de dados relacional MySQL. O *front-end* se comunica com o *back-end* por meio dessa API.

A API REST em SpringBoot é responsável por gerenciar as solicitações do usuário, processar dados e retornar as respostas para o *front-end*. O banco de dados relacional MySQL é usado para armazenar informações do usuário, informações de login, lançamentos de entradas e saídas e as categorias padrões do aplicativo.

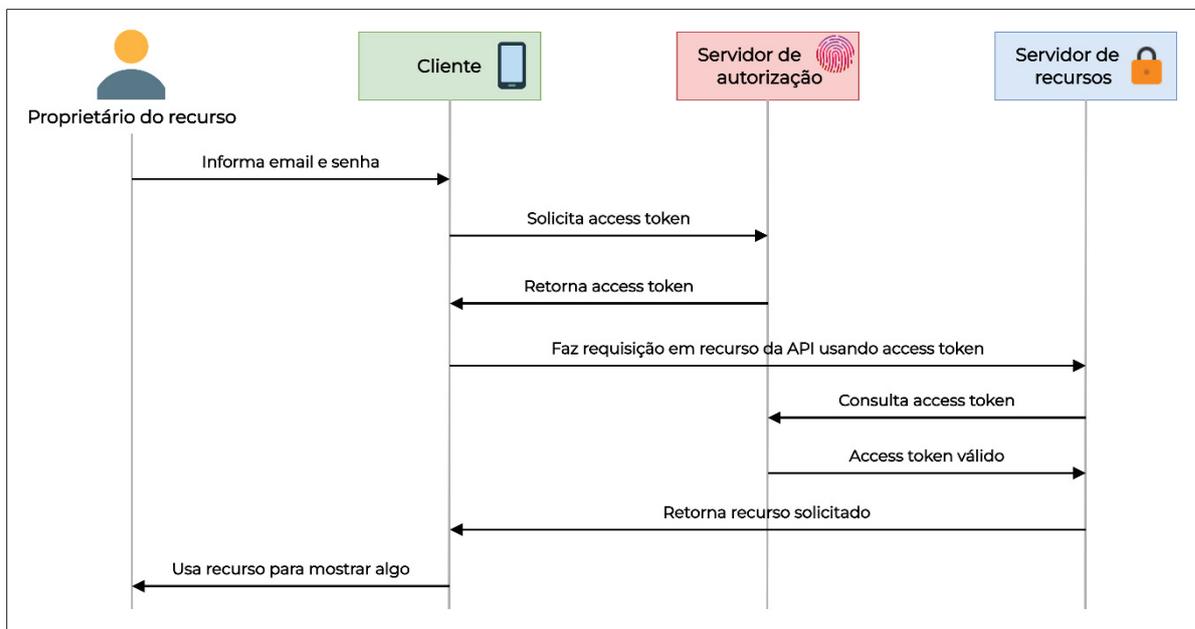


FONTE: O autor (2023)

Para garantir a segurança do usuário em um aplicativo, é essencial adotar medidas de criptografia de senha confiáveis. O bcrypt é um algoritmo de criptografia de senha amplamente utilizado, que transforma a senha em um hash unidirecional, o que impede que ela seja revertida para a senha original. Foi projetado para ser uma alternativa segura e resistente a ataques de força bruta a outros algoritmos de hashing de senha, como o MD5 e o SHA1, que foram posteriormente considerados inseguros (PROVOS; MAZIÈRES, 1999). O bcrypt é conhecido por sua segurança e é amplamente utilizado em aplicações web e bancos de dados para armazenar senhas de usuário com segurança.

Durante o processo de cadastro do usuário no aplicativo, a senha informada passa pelo algoritmo bcrypt, o hash gerado é armazenado no banco de dados, evitando que a senha seja salva como texto puro. Posteriormente, durante a autenticação do usuário (solicita o *access token*), a senha enviada passa pelo mesmo algoritmo, e o hash gerado é comparado com o hash salvo do usuário. Autenticado com sucesso, o usuário possuirá um Token que permite que ele tenha acesso aos recursos da API do *back-end*.

FIGURA 2 - SEQUÊNCIA DO FLUXO DE AUTENTICAÇÃO DO USUÁRIO



FONTE: O autor (2023)

### 1.4.1 OAuth2

O OAuth2 é um protocolo de autorização que permite que um aplicativo obtenha acesso seguro a recursos protegidos em nome de um usuário, sem precisar conhecer ou armazenar as credenciais de autenticação desse usuário. A documentação oficial do OAuth2, especificada na RFC 6749, descreve o framework em detalhes.

De acordo com a documentação do OAuth2, o protocolo é baseado em uma arquitetura cliente/servidor, em que o cliente representa o aplicativo solicitante, o servidor de autorização é responsável por autenticar o usuário e gerenciar as permissões de acesso, e o servidor de recursos hospeda os recursos protegidos que o cliente deseja acessar (AUTH0, 2023).

O OAuth2 também define outros fluxos de autorização, como o fluxo implícito, o fluxo de senha do proprietário e o fluxo de credenciais do cliente, que fornecem diferentes abordagens para obter um token de acesso, dependendo das necessidades e capacidades do cliente.

### 1.4.2 Java

Java é uma linguagem de programação orientada a objetos, que foi desenvolvida pela Sun Microsystems na década de 1990. Ela se tornou uma das linguagens de programação mais populares e é usada em uma ampla variedade de aplicativos, desde aplicativos web até aplicativos de desktop e dispositivos móveis.

De acordo com Deitel e Deitel (2011), a linguagem Java foi projetada para ser portátil, robusta, segura e fácil de usar. A portabilidade significa que os programas escritos em Java podem ser executados em qualquer sistema operacional que tenha uma máquina virtual Java (JVM) instalada. A robustez refere-se à capacidade da linguagem Java de detectar e lidar com erros de tempo de execução, como exceções, sem que o programa seja interrompido. A segurança é obtida através de mecanismos como o sandbox da JVM, que limita o acesso do programa a recursos do sistema operacional, tornando a linguagem Java uma escolha popular para aplicativos da web. A facilidade de uso é obtida através de uma sintaxe clara e concisa, bem como de recursos de biblioteca padrão abrangentes.

### 1.4.3 Spring Boot

Para o desenvolvimento do serviço que consulta a base de dados utilizou-se o *framework* Spring Boot, uma tecnologia popular no desenvolvimento de aplicações Java. De acordo com o site oficial do Spring Boot, alguns dos motivos para utilizar essa tecnologia incluem a facilidade de configuração, empacotamento em arquivos JAR executáveis, flexibilidade, compatibilidade com o Spring Framework e suporte à arquitetura de microsserviços.

Uma das principais vantagens do Spring Boot é a configuração automática, que permite que os desenvolvedores se concentrem na lógica de negócios em vez de perder tempo com configurações manuais. Além disso, o empacotamento em arquivos JAR executáveis torna a implantação do aplicativo muito mais fácil (SPRING IO, 2023).

### 1.4.4 Dart

Dart é uma linguagem de programação criada pela Google em 2011 com o objetivo de ser uma linguagem fácil de aprender, escalável e adequada para desenvolvimento web e de aplicativos móveis. É uma linguagem de tipagem estática, mas também suporta tipagem dinâmica, permitindo que os desenvolvedores escolham a opção mais adequada para seus projetos.

Segundo o site oficial do Dart, a linguagem foi criada para atender a demanda por uma linguagem moderna que abrangesse as necessidades dos desenvolvedores para construir aplicativos web e móveis cada vez mais complexos. O site também destaca a sintaxe clara e concisa do Dart, que torna a linguagem fácil de aprender e ler.

A vantagem do Dart é o seu ecossistema de ferramentas. A Google oferece várias ferramentas para desenvolvimento com Dart, como o Flutter, um framework para desenvolvimento de aplicativos móveis que utiliza o Dart como linguagem de programação.

#### 1.4.5 Flutter

O Flutter é um framework para o desenvolvimento de aplicativos móveis que foi criado pela Google em 2017. De acordo com informações divulgadas pela própria empresa, o Flutter foi desenvolvido para suprir algumas lacunas existentes em outras ferramentas de desenvolvimento mobile, como a dificuldade em criar interfaces de usuário personalizadas e a baixa produtividade no desenvolvimento de aplicativos multiplataforma.

Um aspecto relevante do Flutter é a sua capacidade de criar interfaces personalizadas e exclusivas. Isso se deve em grande parte à sua arquitetura baseada em widgets, que permite criar componentes únicos e personalizáveis para interface do usuário. Um estudo publicado no "International Journal of Advanced Research in Computer Science and Software Engineering" em 2021 destacou a flexibilidade dos widgets do Flutter para a criação de interfaces de usuário atraentes e dinâmicas. A possibilidade de criar interfaces de forma ágil e em várias plataformas foi o aspecto que motivou a escolha do Flutter para o desenvolvimento do produto.

#### 1.4.6 MySQL

Para o armazenamento e gerenciamento dos dados utilizou-se o MySQL, devido ao seu excelente desempenho e longa presença no mercado. Além disso, ele oferece diversas funcionalidades de forma gratuita.

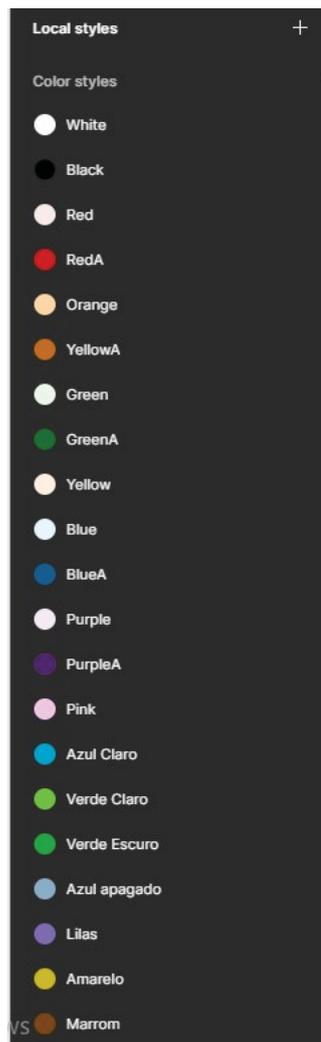
O MySQL é um sistema de gerenciamento de banco de dados relacional de código aberto, desenvolvido pela MySQL AB e atualmente pertencente à Oracle Corporation. É evidente a ampla utilização do MySQL por desenvolvedores de software em todo o mundo, devido à sua facilidade de uso, escalabilidade e performance, além de suportar uma ampla variedade de plataformas e linguagens de programação. O MySQL é uma opção acessível e eficiente para empresas de todos os tamanhos que desejam gerenciar grandes quantidades de dados de forma confiável e segura (MILANI, 2016).

### 1.4.7 Figma

O Figma é uma ferramenta de design e prototipagem bastante popular. Com essa ferramenta foi possível construir vários exemplos de telas e componentes. Em paralelo foram aplicados conceitos de usabilidade e design de interface. Definindo os padrões de cores (FIGURA 3), tamanho de botões, arredondamento de bordas, pesos e tamanhos de fontes, etc. A prototipação das telas foi muito mais ágil.

A utilização da ferramenta ocorreu em algumas etapas do projeto quando era necessário modificar, testar e padronizar diferentes estilos e componentes (FIGURA 4) sem a necessidade de programar e alterar o código. As alterações que ocorreram no design das telas são demonstradas na FIGURA 5 e na FIGURA 6. Uma vez definido o layout da tela na ferramenta implementava-se o código Flutter de fato.

FIGURA 3 – PADRÃO DE CORES DO PROJETO



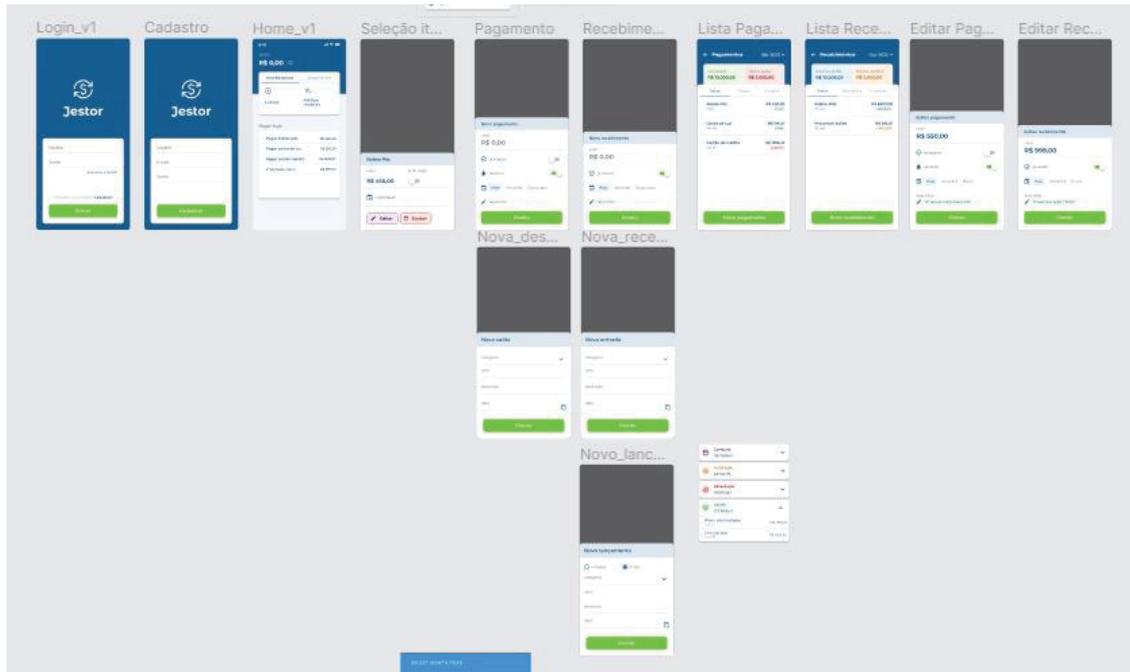
FONTE: O autor (2023)

FIGURA 4 - PADRÃO DE ÍCONE E CORES DAS CATEGORIAS

CATEGORIAS	
Ícones e suas cores	
Entradas	Saídas
 Renda	 Moradia
 Bônus	 Saúde
 Vendas	 Educação
 Rendimentos	 Lazer
 Outras entradas	 Transporte
	 Alimentação
	 Compras
	 Pet
	 Outras saídas

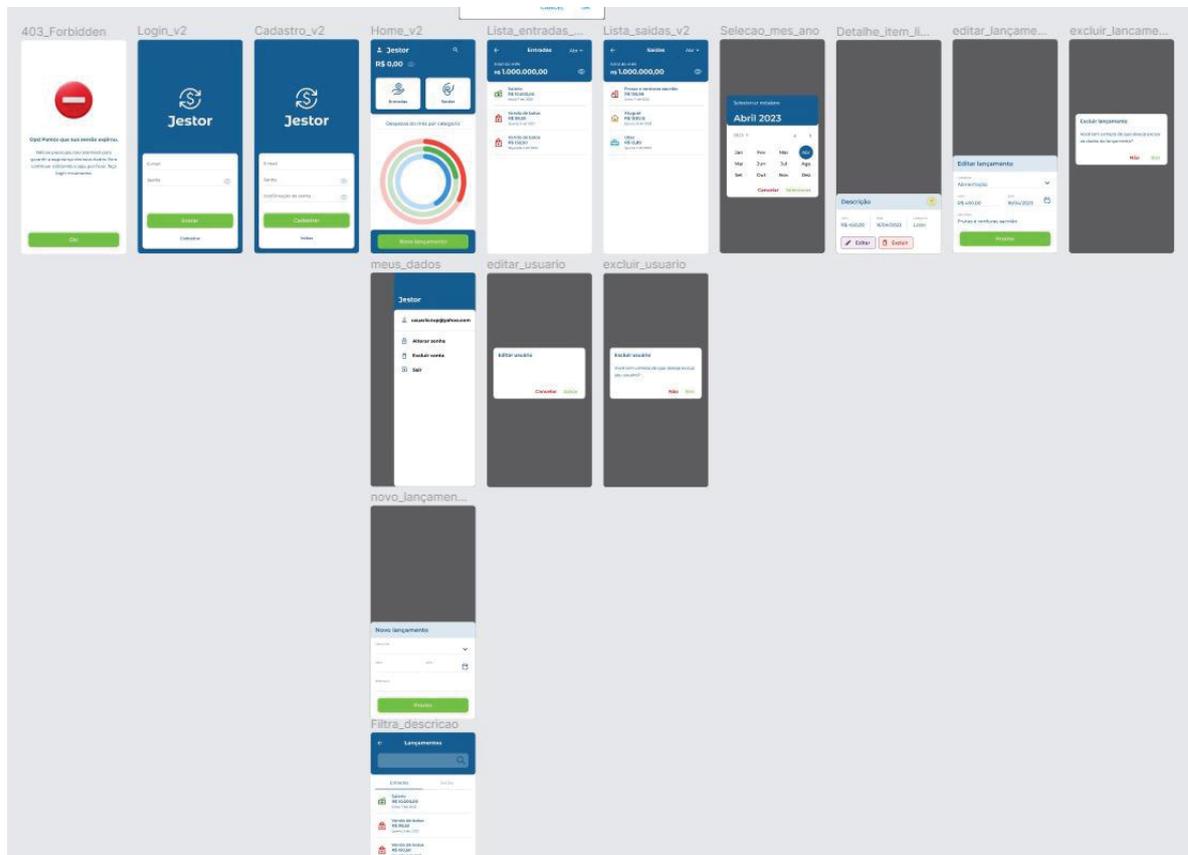
FONTE: O autor (2023)

FIGURA 5 - PROTÓTIPO INICIAL



FONTE: O autor (2023)

FIGURA 6 - PROTÓTIPO FINAL

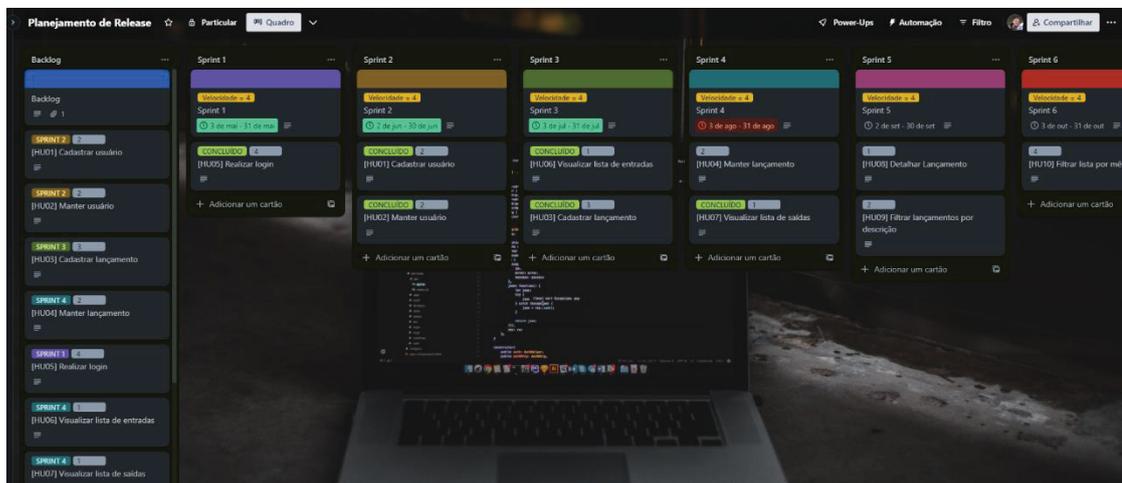


FONTE: O autor (2023)

### 1.4.8 Trello

O Trello é uma ferramenta de gerenciamento de projetos e tarefas. Utilizar essa ferramenta facilitou a criação do quadro de planejamento de release e a priorização do backlog, conforme mostra a FIGURA 7. Foi possível acompanhar o progresso do desenvolvimento com cards personalizados e colunas que identificavam a sprint a qual cada card pertencia. A ferramenta foi escolhida por ser popular e possuir muitos recursos gratuitos e úteis.

FIGURA 7 - QUADRO DE RELEASES



FONTE: O autor (2023)

### 1.4.9 Hardware

O hardware utilizado para o desenvolvimento do produto é um computador desktop com configurações bastante avançadas. O processador utilizado é o AMD Ryzen 5 3600, um modelo de 6 núcleos capaz de processar até 12 threads simultâneas. O clock base do processador é de 3,6 GHz, o que significa que ele pode executar instruções com uma taxa de 3,6 bilhões de ciclos por segundo.

Além disso, o computador possui 16GB de memória RAM, que é uma quantidade muito boa para a maioria das tarefas de desenvolvimento de software. Isso significa que o computador pode armazenar um grande número de dados na memória, o que ajuda a acelerar o desempenho geral do sistema.

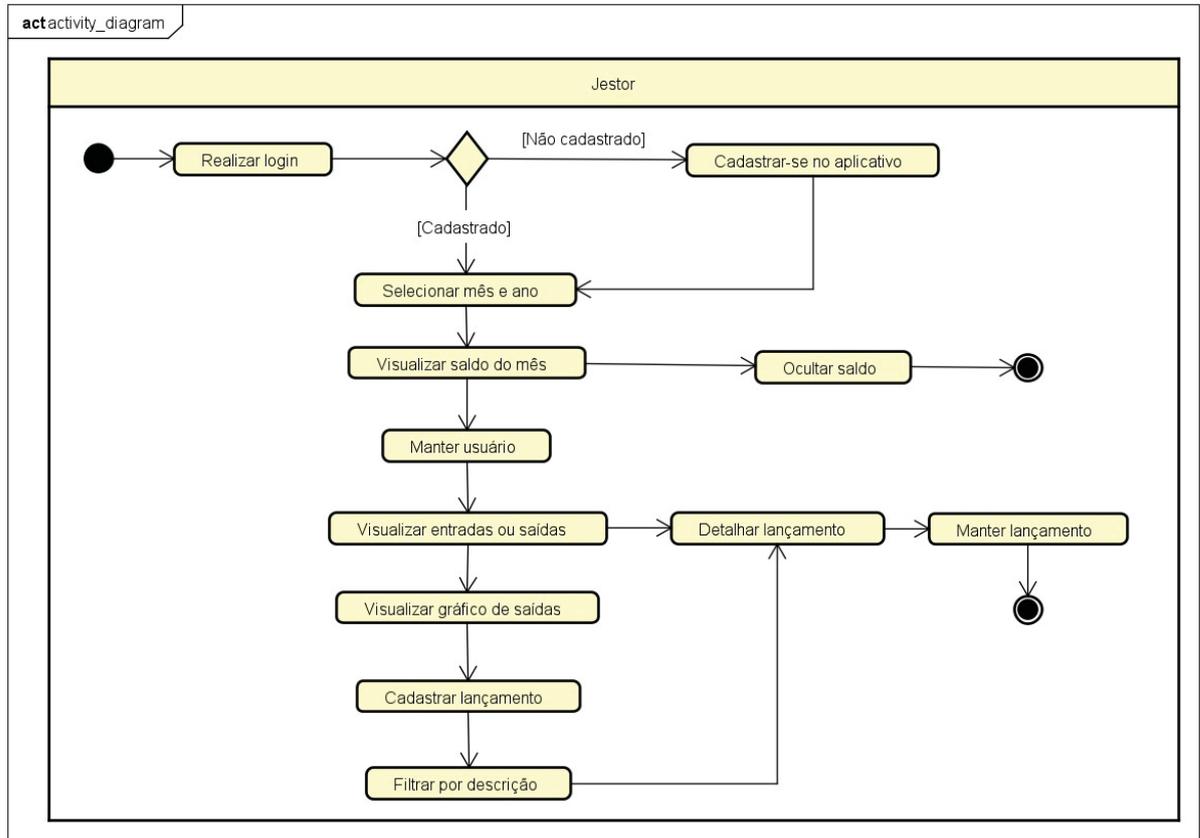
O HD do computador possui 1TB de capacidade, o que é uma quantidade bastante generosa de espaço de armazenamento. Isso permite armazenar uma grande quantidade de arquivos, como imagens, documentos, vídeos e outros dados, sem se preocupar com a falta de espaço. Além disso, também possui um SSD de 256GB, que é um tipo de armazenamento de estado sólido mais rápido que os discos rígidos tradicionais.

Por fim, o computador está equipado com uma placa de vídeo NVIDIA GeForce GTX 1650 ti de 12gb, o que significa que ele pode ser usado para tarefas de processamento gráfico pesado, como a renderização de imagens e vídeos.

Em resumo, o hardware utilizado para o desenvolvimento do produto é bastante poderoso e deve ser capaz de executar uma ampla variedade de tarefas relacionadas ao desenvolvimento de software.

## 2 DESENHO DO PROCESSO MACRO

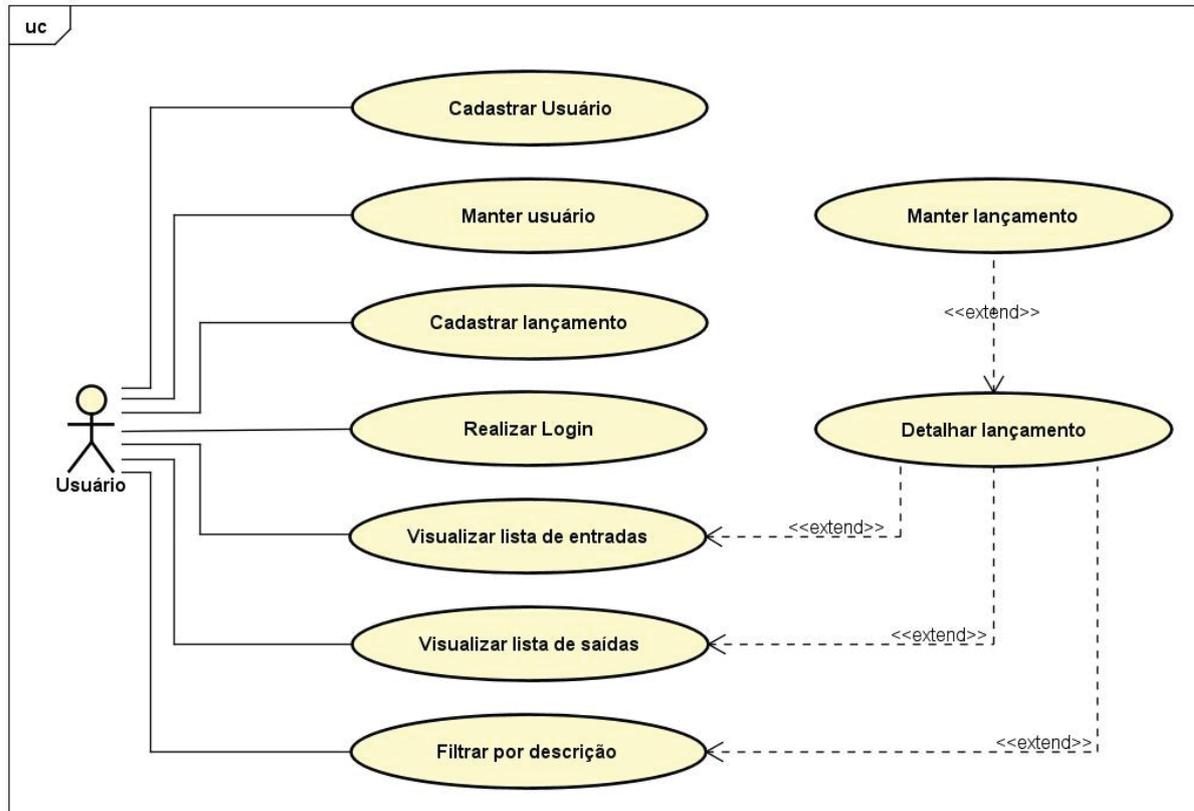
FIGURA 8 - DIAGRAMA DE ATIVIDADES DO PROCESSO MACRO DO APLICATIVO



FONTE: O autor (2023)

### 3 DIAGRAMA DE CASOS DE USO

FIGURA 9 - DIAGRAMA DE CASOS DE USO



FONTE: O autor (2023)

## 4 HISTÓRIAS DE USUÁRIO

HU01 – Cadastrar usuário

HU02 – Manter usuário

HU03 – Cadastrar lançamento

HU04 – Manter lançamento

HU05 – Realizar login

HU06 – Visualizar lista de entradas

HU07 – Visualizar lista de saídas

HU08 – Detalhar lançamento

HU09 – Filtrar lançamentos por descrição

HU10 – Selecionar mês e ano

### HU01 – Cadastrar usuário

**SENDO** uma pessoa não cadastrada  
**QUERO** realizar meu cadastro no aplicativo  
**PARA** ter a possibilidade de cadastrar lançamentos (entradas e saídas), visualizar o saldo do mês e o gráfico de despesas por categoria.

### DESENHO DA(S) TELA(S):

FIGURA 10 - TELA DE CADASTRO DE USUÁRIO

A imagem mostra a interface de usuário para o cadastro de um novo usuário no sistema Jestor. O cabeçalho é azul escuro com o logo 'Jestor' em branco, que consiste em um símbolo de dólar dentro de um círculo com setas. Abaixo do cabeçalho, há um formulário branco com três campos de entrada: 'E-mail', 'Senha' e 'Confirmação de senha'. O campo 'E-mail' tem uma barra de progresso à direita indicando '0/100'. Os campos 'Senha' e 'Confirmação de senha' possuem ícones de olho para alternar a visibilidade da senha. Abaixo dos campos, há um botão verde com o texto 'Cadastrar' e um link azul com o texto 'Voltar'.

FONTE: O autor (2023)

### CRITÉRIOS DE ACEITAÇÃO:

1. Deve permitir o cadastro de usuários
2. Não deve prosseguir com campos e-mail e senha vazios
3. Deve validar e-mail utilizando regex
4. Não deve permitir senha com menos de 8 caracteres
5. Deve permitir ver senha e ocultar senha
6. Deve retornar à tela de Login
7. Deve validar se e-mail informado já está cadastrado
8. Deve validar se as senhas coincidem

### CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO:

**Critério de contexto (Válido como premissa para todos os critérios):**

**Dado que** não tenho cadastro  
**E** cliquei no botão “Cadastrar”

### 1. Deve permitir o cadastro de usuários

**Dado que** preenchi o campo e-mail válido  
**E** a senha com mais de 8 caracteres  
**Quando** pressionar o botão “Cadastrar”  
**Então** o sistema deve realizar o registro dos meus dados

### 2. Não deve prosseguir com campos e-mail e senha vazios

**Dado que** deixei um ou mais campos vazios (R1)  
**Quando** pressionar o botão “Cadastrar”  
**Então** o sistema retorna mensagem (R1)

### 3. Deve validar e-mail

**Dado que** inseri um e-mail inválido (R1)  
**Quando** pressionar o botão “Cadastrar”  
**Então** o sistema retorna mensagem (R1)

### 4. Não deve permitir senha com menos de 8 caracteres

**Dado que** inseri uma senha com menos de 8 caracteres (R1)  
**Quando** pressionar o botão “Cadastrar”  
**Então** o sistema retorna mensagem (R1)

### 5. Deve permitir ver senha e ocultar senha

**Dado que** inseri uma senha com mais de 8 caracteres  
**Quando** pressionar o botão “ver ou ocultar senha”  
**Então** a senha se torna visível ou esconde a senha com asteriscos

## 6. Deve retornar à tela de Login

**Dado que** que não desejo mais me cadastrar

**Quando** pressionar o botão “Voltar”

**Então** o sistema retorna à tela de login

## 7. Deve validar se e-mail informado já está cadastrado

**Dado que** inseri um e-mail e uma senha

**Quando** pressionar o botão “Cadastrar”

**Então** o sistema retorna mensagem (R2)

## 8. Deve validar se as senhas coincidem

**Dado que** inseri uma senha

**Quando** informo a confirmação da senha

**Então** o sistema retorna mensagem (R1)

## REGRAS DE NEGÓCIO DA HISTÓRIA:

R1 - Inconsistência dos campos

TABELA 1 - INCONSISTÊNCIAS NO CADASTRO DE USUÁRIO

Inconsistência	Mensagem
Campo senha vazio	“Por favor, digite uma senha”
Campo e-mail vazio	“Por favor, digite seu e-mail”
Campo e-mail inválido	“Por favor, digite um e-mail válido”
Campo senha inválida	“A senha deve ter ao menos 8 caracteres”
Senhas diferentes	“Ops! As senhas digitadas não correspondem”

FONTE: O autor (2023)

## R2 - Validação de usuário

TABELA 2 - VALIDAÇÃO E-MAIL

E-mail existe?	Mensagem
Sim	“Ops! Este e-mail já está cadastrado”

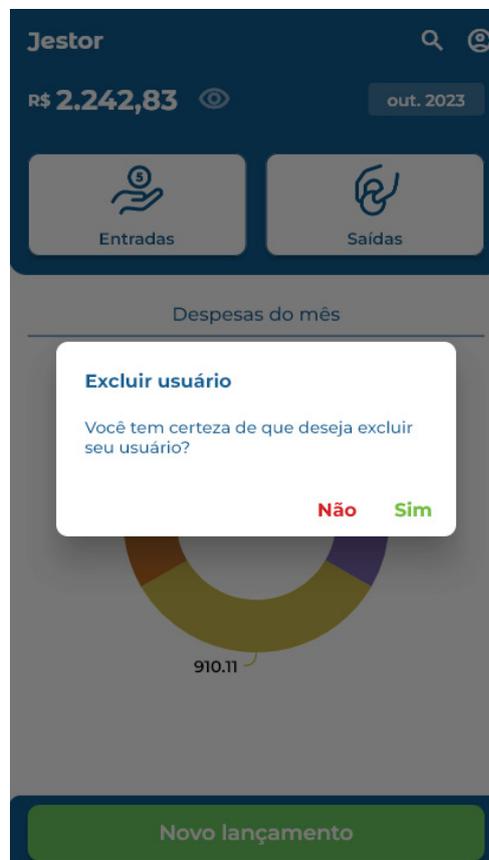
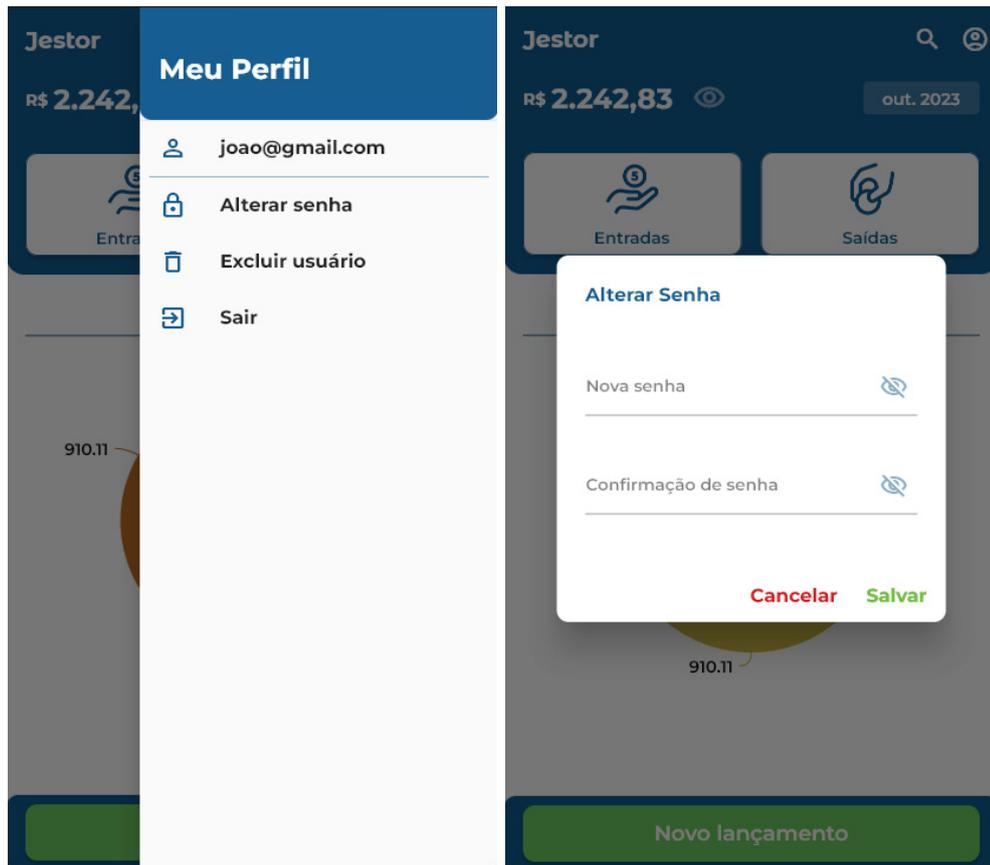
FONTE: O autor (2023)

## HU02 – Manter usuário

- SENDO** um usuário cadastrado
- QUERO** editar ou excluir meu cadastro no aplicativo
- PARA** ter a possibilidade de atualizar meus dados de acesso

## DESENHO DA(S) TELA(S):

FIGURA 11 - TELAS DE EDIÇÃO E EXCLUSÃO DE USUÁRIO



FONTE: O autor (2023)

**CRITÉRIOS DE ACEITAÇÃO:**

1. Deve permitir editar a senha do usuário
2. Deve permitir excluir os dados do usuário

**CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO:****Critério de contexto (Válido como premissa para todos os critérios):**

**Dado que** sou um usuário cadastrado

**1. Deve permitir editar a senha do usuário**

**Dado que** pressionar a opção “Alterar senha”

**Quando** abrir a modal de alteração de senha

**Então** informar a nova senha

**E** quando pressionar “Salvar”, minha senha é alterada

**2. Deve permitir excluir os dados do usuário**

**Dado que**

**Quando** pressionar o botão “Excluir”

**Então** é chamada a modal de confirmação (R1)

**E** quando pressionar “Sim”, meu usuário é excluído do sistema

**REGRAS DE NEGÓCIO DA HISTÓRIA:**

R1 - Modal de confirmação de exclusão de usuário

Ao clicar na opção “Excluir usuário”, será chamada a modal de confirmação com a frase “Você tem certeza de que deseja excluir seu usuário?” e dois botões “Não” e “Sim”, ao clicar em “Sim” o usuário é excluído do banco de dados e, ao clicar em “Não”, o modal é fechado.

## HU03 – Cadastrar lançamento

**SENDO** um usuário cadastrado no aplicativo  
**QUERO** cadastrar um lançamento  
**PARA** que eu possa visualizar as minhas entradas e saídas

### DESENHO DA(S) TELA(S):

FIGURA 12 - TELA DE CADASTRO DE LANÇAMENTO

A imagem mostra a interface de usuário do aplicativo 'Jestor' para o cadastro de um lançamento. No topo, há o nome do aplicativo, ícones de busca e perfil, o saldo atualizado em reais (R\$ 2.242,83) e o mês selecionado (out. 2023). Abaixo, dois botões permitem alternar entre 'Entradas' e 'Saídas'. O formulário principal, intitulado 'Novo lançamento', possui campos para: categoria (Alimentação), valor (R\$ 0,00), data (01/10/2023) e descrição (0/40). Um botão verde 'Pronto' está localizado na base do formulário.

FONTE: O autor (2023).

### CRITÉRIOS DE ACEITAÇÃO:

1. Deve permitir o cadastro de lançamentos
2. Não deve permitir campos vazios
3. Deve permitir selecionar uma categoria padrão

4. Deve valorizar automaticamente os campos ao abrir
5. Deve limitar a quantidade de caracteres do campo descrição

### **CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO:**

#### **Critério de contexto (Válido como premissa para todos os critérios):**

**Dado que** estou logado no aplicativo  
**Quando** cliquei no botão da tela inicial “Novo lançamento”  
**Então** apresenta a modal de cadastro de lançamento

#### **1. Deve permitir o cadastro de lançamentos**

**Dado que** os campos categoria, valor, descrição e data foram preenchidos corretamente  
**Quando** pressionar o botão “Pronto”  
**Então** o sistema deve salvar os dados do Lançamento e fechar a modal de cadastro de lançamentos

#### **2. Não deve permitir campos vazios**

**Dado que** o campo descrição está vazio  
**Ou** o campo valor está zerado ou vazio  
**Quando** pressionar o botão “Pronto”  
**Então** o sistema retorna a mensagem (R1)

#### **3. Deve permitir selecionar uma categoria padrão**

**Dado que** quero selecionar uma categoria  
**Quando** colocar o foco no campo de categoria  
**Então** deve listar as opções de categorias padrões  
**E** em ordem alfabética

#### **4. Deve valorizar automaticamente os campos ao abrir**

**Dado que** a modal abriu

**Quando** visualizo a tela

**Então** o campo valor deve vir zerado e no formato “R\$ 0,00”

**E** o campo data deve vir preenchido com o primeiro dia do mês selecionado no formato “dd/MM/yyyy”

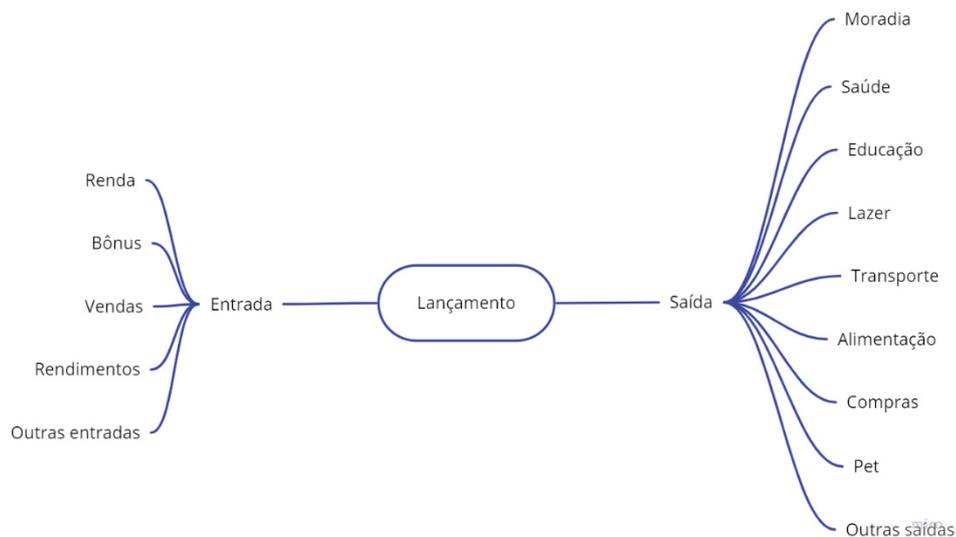
## 5. Deve limitar a quantidade de caracteres do campo descrição

**Dado que** preenchi o campo descrição

**Quando** atingir 40 caracteres

**Então** não consigo digitar mais caracteres

FIGURA 13 - MAPA MENTAL DAS CATEGORIAS DE ENTRADAS E SAÍDAS



FONTE: O autor (2023)

## REGRAS DE NEGÓCIO DA HISTÓRIA:

R1 – Inconsistência dos campos

TABELA 3 - INCONSISTÊNCIAS NO CADASTRO DE LANÇAMENTO

Inconsistência	Mensagem
Campo valor vazio ou zerado	“Por favor, digite um valor”

Campo descrição vazio	“Por favor, digite uma descrição”
-----------------------	-----------------------------------

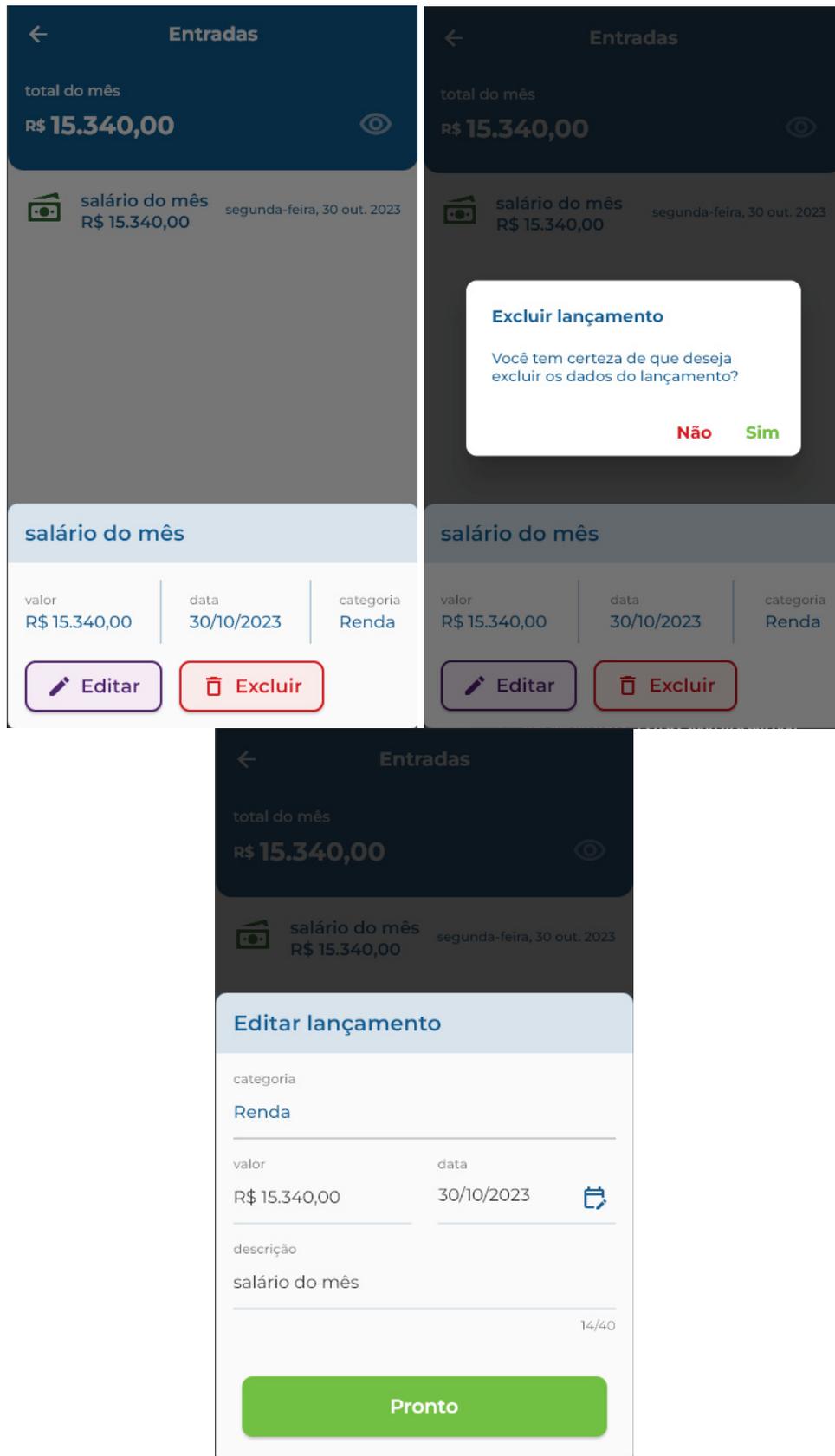
FONTE: O autor (2023)

#### **HU04 – Manter lançamento**

**SENDO** um usuário que já cadastrou um lançamento  
**QUERO** editar ou excluir um lançamento  
**PARA** que os dados sejam atualizados

**DESENHO DA(S) TELA(S):**

FIGURA 8 – TELAS DA EDIÇÃO E EXCLUSÃO DE LANÇAMENTO



FONTE: O autor (2023)

## CRITÉRIOS DE ACEITAÇÃO:

1. Deve permitir editar os dados do lançamento
2. Deve permitir excluir um lançamento

## CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO:

### Critério de contexto (Válido como premissa para todos os critérios):

**Dado que** estou logado no aplicativo  
**E** visualizei os detalhes do lançamento

#### 1. Deve permitir editar os dados do lançamento

**Dado que**  
**Quando** altero os dados do lançamento  
**E** pressionno o botão “Pronto”  
**Então** o sistema deve salvar os dados editados

#### 2. Deve permitir excluir um lançamento

**Dado que** existe a necessidade de excluir lançamento que cadastrei  
**Quando** pressionno o botão “Excluir”  
**Então** é chamada a modal de confirmação (R1)  
**E** quando pressionno “Sim”, o lançamento é excluído do sistema

## REGRAS DE NEGÓCIO DA HISTÓRIA:

R1 - Modal de confirmação de exclusão de lançamento

Ao clicar no botão “Excluir”, será chamada a modal de confirmação com a frase “Você tem certeza de que deseja excluir os dados do lançamento?” e dois botões “Não” e “Sim”, ao clicar em “Sim” o lançamento é excluído do banco de dados e, ao clicar em “Não”, o modal é fechado.

## HU05 – Realizar login

<b>SENDO</b>	um usuário cadastrado no aplicativo
<b>QUERO</b>	realizar o login no aplicativo
<b>PARA</b>	que eu possa acessar as funcionalidades disponíveis
<b>E PARA</b>	gerir e controlar minhas finanças

## DESENHO DA(S) TELA(S):

FIGURA 14 - TELA DE LOGIN



FONTE: O autor (2023)

## CRITÉRIOS DE ACEITAÇÃO:

1. Não deve permitir enviar 'E-mail' ou 'senha' em branco
2. Não deve permitir acesso caso os dados não coincidam com os cadastrados

3. Deve realizar o login
4. Deve redirecionar para o cadastro

### **CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO:**

#### **Critério de contexto (Válido como premissa para todos os critérios):**

**Dado que** estou cadastrado no sistema

#### **1. Não deve permitir enviar 'E-mail' ou 'senha' em branco**

**Dado que** um dos campos esteja vazio (R1)

**Quando** pressionar o botão "Entrar"

**Então** o sistema retorna mensagem de erro (R1)

#### **2. Não deve permitir acesso caso os dados não estejam cadastrados**

**Dado que** os dados são diferentes dos cadastrados (R1)

**Quando** pressionar o botão "Entrar"

**Então** o sistema retorna mensagem de erro (R1)

#### **3. Deve realizar o login**

**Dado que** ambos os campos foram preenchidos

**E** os dados informados coincidem com os cadastrados

**Quando** pressionar o botão "Entrar"

**Então** é possível acessar a minha conta

#### **4. Deve redirecionar para cadastro**

**Dado que** ambos os campos foram preenchidos

**Quando** pressionar o botão "Cadastrar"

**Então** o sistema deve redirecionar para a tela de cadastro

### **REGRAS DE NEGÓCIO DA HISTÓRIA:**

R1 - Inconsistência dos campos:

TABELA 4 - INCONSISTÊNCIAS NO LOGIN

<b>Inconsistência</b>	<b>Mensagem</b>
Campo senha vazio	“Por favor, digite sua senha”
Campo e-mail vazio	“Por favor, digite seu e-mail”
Campo e-mail inválido	“Por favor, digite um e-mail válido”
Campo senha inválida	“A senha deve ter ao menos 8 caracteres”
Campos com dados não cadastrados	“E-mail ou senha inválidos”

FONTE: O autor (2023)

#### **HU06 – Visualizar lista de entradas**

**SENDO** um usuário logado no aplicativo  
**QUERO** visualizar a lista de entradas  
**PARA** facilitar minha gestão e entendimento do que entra

**DESENHO DA(S) TELA(S):**

FIGURA 15 - TELA DE LISTAGEM DE ENTRADAS



FONTE: O autor (2023)

### CRITÉRIOS DE ACEITAÇÃO:

1. Deve permitir mostrar ou ocultar saldo do mês
2. Deve listar as entradas

### CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO:

**Critério de contexto (Válido como premissa para todos os critérios):**

**Dado que** estou logado na minha conta  
**E** acesso a lista de entradas

1. Deve permitir mostrar ou ocultar saldo do mês

**Dado que** a tela de listagem mostra o saldo do mês  
**Quando** pressionar o botão com o ícone de olho  
**Então** deve ocultar ou mostrar o saldo do mês

## 2. Deve listar as entradas

**Dado que**  
**Quando** acesso a lista de entradas  
**Então** visualizo as entradas conforme o padrão (R1) (R2)

### REGRAS DE NEGÓCIO DA HISTÓRIA:

R1 – A lista deve mostrar as entradas do mês selecionado

R2 – A ordenação padrão da lista deve ser por data (descrescente)

### HU07 – Visualizar lista de saídas

**SENDO** um usuário logado no aplicativo  
**QUERO** visualizar a lista de saídas  
**PARA** facilitar minha gestão e entendimento do que gasto

### DESENHO DA(S) TELA(S):

FIGURA 16 - TELA DE LISTAGEM DE SAÍDAS



FONTE: O autor (2023)

### CRITÉRIOS DE ACEITAÇÃO:

1. Deve permitir mostrar ou ocultar saldo do mês
2. Deve listar as saídas

### CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO:

**Critério de contexto (Válido como premissa para todos os critérios):**

**Dado que** estou logado na minha conta  
**E** acesso a lista de saídas

1. Deve permitir mostrar ou ocultar saldo do mês

**Dado que** a tela de listagem mostra o saldo do mês  
**Quando** pressionar o botão com o ícone de olho  
**Então** deve ocultar ou mostrar o saldo do mês

## 2. Deve listar as saídas

**Dado que**  
**Quando** acesso a lista de saídas  
**Então** visualizo as saídas conforme o padrão (R1) (R2)

### REGRAS DE NEGÓCIO DA HISTÓRIA:

R1 - A lista deve mostrar as saídas do mês selecionado

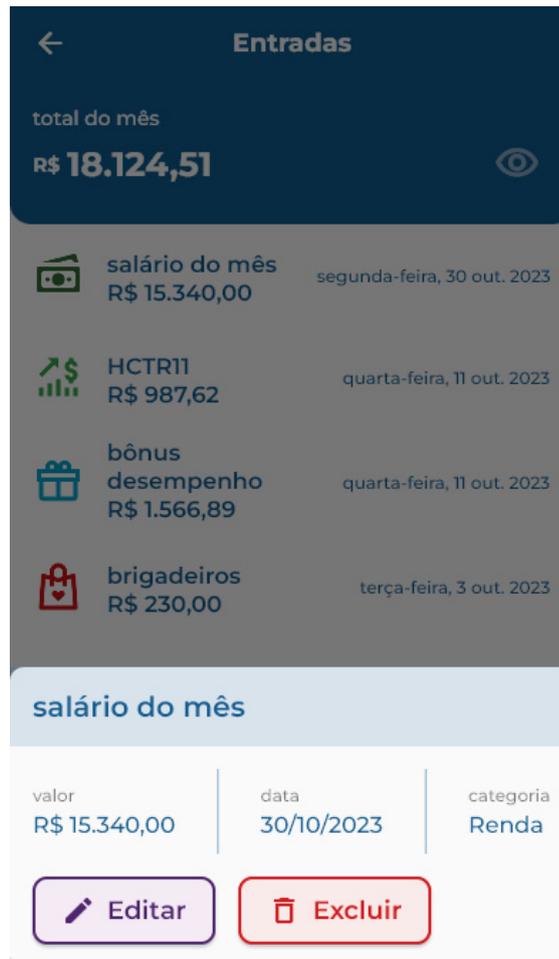
R2 – A ordenação padrão da lista deve ser por data (descrescente)

### HU08 – Detalhar lançamento

**SENDO** um usuário que possui lançamentos cadastrados  
**QUERO** visualizar os detalhes do lançamento  
**PARA** editar, excluir ou somente ver as informações

### DESENHO DA(S) TELA(S):

FIGURA 17 - TELA DA MODAL DE DETALHE DE LANÇAMENTO



FONTE: O autor (2023)

### CRITÉRIOS DE ACEITAÇÃO:

1. Deve detalhar o item da lista
2. Deve abrir a modal de edição de lançamento
3. Deve abrir a modal de confirmação de exclusão

### CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO:

**Critério de contexto (Válido como premissa para todos os critérios):**

**Dado que** estou logado na minha conta  
**E** acesso a lista de entradas ou saídas

1. Deve detalhar o item da lista

**Dado que**

**Quando** seleciono um item da lista

**Então** é mostrado a modal com os detalhes do item

## **2. Deve abrir a modal de edição de lançamento**

**Dado que** selecionei um item e a modal de detalhes está aberta

**Quando** pressionar o botão “Editar”

**Então** a modal de detalhes fecha e abre a modal de editar lançamento

## **3. Deve abrir a modal de confirmação de exclusão**

**Dado que** selecionei um item e a modal de detalhes está aberta

**Quando** pressionar o botão “Excluir”

**Então** a modal de detalhes fecha e abre a modal de confirmação de exclusão

## **HU09 – Filtrar lançamentos por descrição**

**SENDO** um usuário que possui lançamentos cadastrados

**QUERO** visualizar a lista de lançamentos e pesquisar pela descrição

**PARA** facilitar minha gestão e entendimento do que ganho e do que gasto

## **DESENHO DA(S) TELA(S):**

FIGURA 18 - TELA DE PEQUISA DE LANÇAMENTOS



FONTE: O autor (2023)

### CRITÉRIOS DE ACEITAÇÃO:

1. Deve listar todos os lançamentos
2. Deve filtrar lista por descrição
3. Deve permitir detalhar item de lista
4. Deve permitir excluir item de lista

### CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO:

**Critério de contexto (Válido como premissa para todos os critérios):**

- Dado que** estou logado na minha conta  
**E** na página inicial do aplicativo  
**E** possuo lançamentos cadastrados

**Quando** pressiono o botão de busca (ícone de lupa)  
**Então** sou direcionado para a tela de pesquisa de lançamentos

### 1. Deve listar todos os lançamentos

**Dado que** acessei a tela de pesquisa de lançamentos  
**Quando** visualizo a listagem  
**Então** deve mostrar todas as entradas e saídas

### 2. Deve filtrar lista por descrição

**Dado que**  
**Quando** informo um valor no campo de pesquisa  
**Então** deve mostrar todas as entradas ou saídas que iniciam com a descrição informada no campo de busca

### 3. Deve permitir detalhar item de lista

**Dado que** cliquei em um item da listagem  
**E** abriu a modal de detalhe de lançamento  
**Quando** clico no botão “Editar”  
**Então** abre a modal de edição de lançamento

### 4. Deve permitir excluir item de lista

**Dado que** cliquei em um item da listagem  
**E** abriu a modal de detalhe de lançamento  
**Quando** clico no botão “Excluir”  
**Então** abre a modal de exclusão de lançamento

## HU10 – Selecionar mês e ano

**SENDO** um usuário que está na tela inicial do app  
**QUERO** visualizar os lançamentos de outro mês e ano

**PARA** facilitar minha gestão e entendimento do que ganho e do que gasto

**DESENHO DA(S) TELA(S):**

FIGURA 19 - TELA DE SELEÇÃO DE MÊS E ANO



FONTE: O autor (2023)

**CRITÉRIOS DE ACEITAÇÃO:**

1. Deve permitir selecionar qualquer mês e ano entre 2001 e 2099
2. Deve consultar os lançamentos do mês e ano selecionados

**CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO:**

**Critério de contexto (Válido como premissa para todos os critérios):**

**Dado que** estou logado na minha conta  
**E** na página inicial do aplicativo  
**Quando** pressionar o botão de selecionar data  
**Então** abre a tela de seleção de mês e ano

### **1. Deve permitir selecionar qualquer mês e ano entre 2001 e 2099**

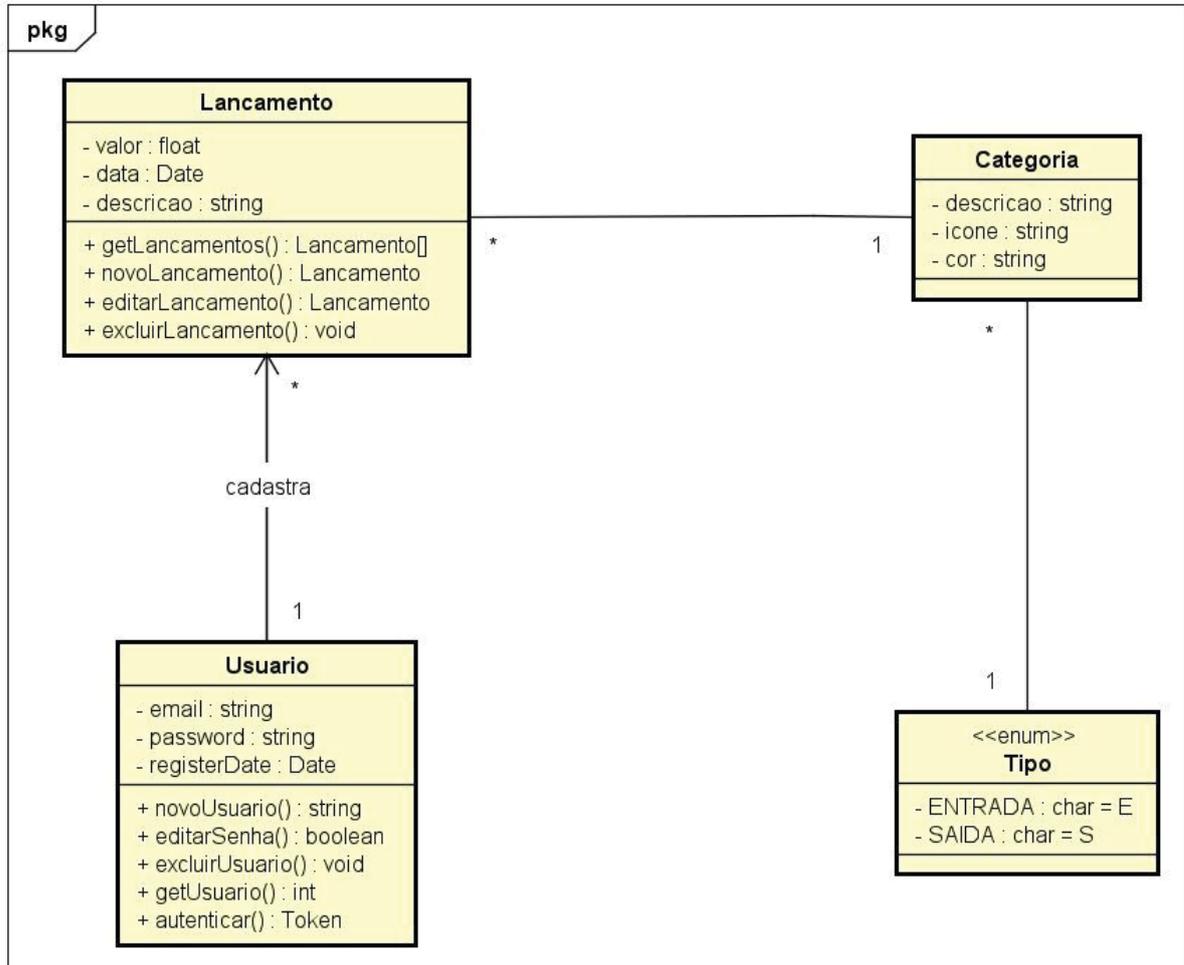
**Dado que**  
**Quando** visualizo o seletor de mês e ano  
**Então** posso selecionar qualquer mês e ano entre 2001 e 2099

### **2. Deve consultar os lançamentos do mês e ano selecionados**

**Dado que** selecionei um mês e ano  
**Quando** pressionar o botão "Selecionar"  
**Então** volta para a tela inicial do app com os dados da data selecionada

## 5 DIAGRAMA DE CLASSES

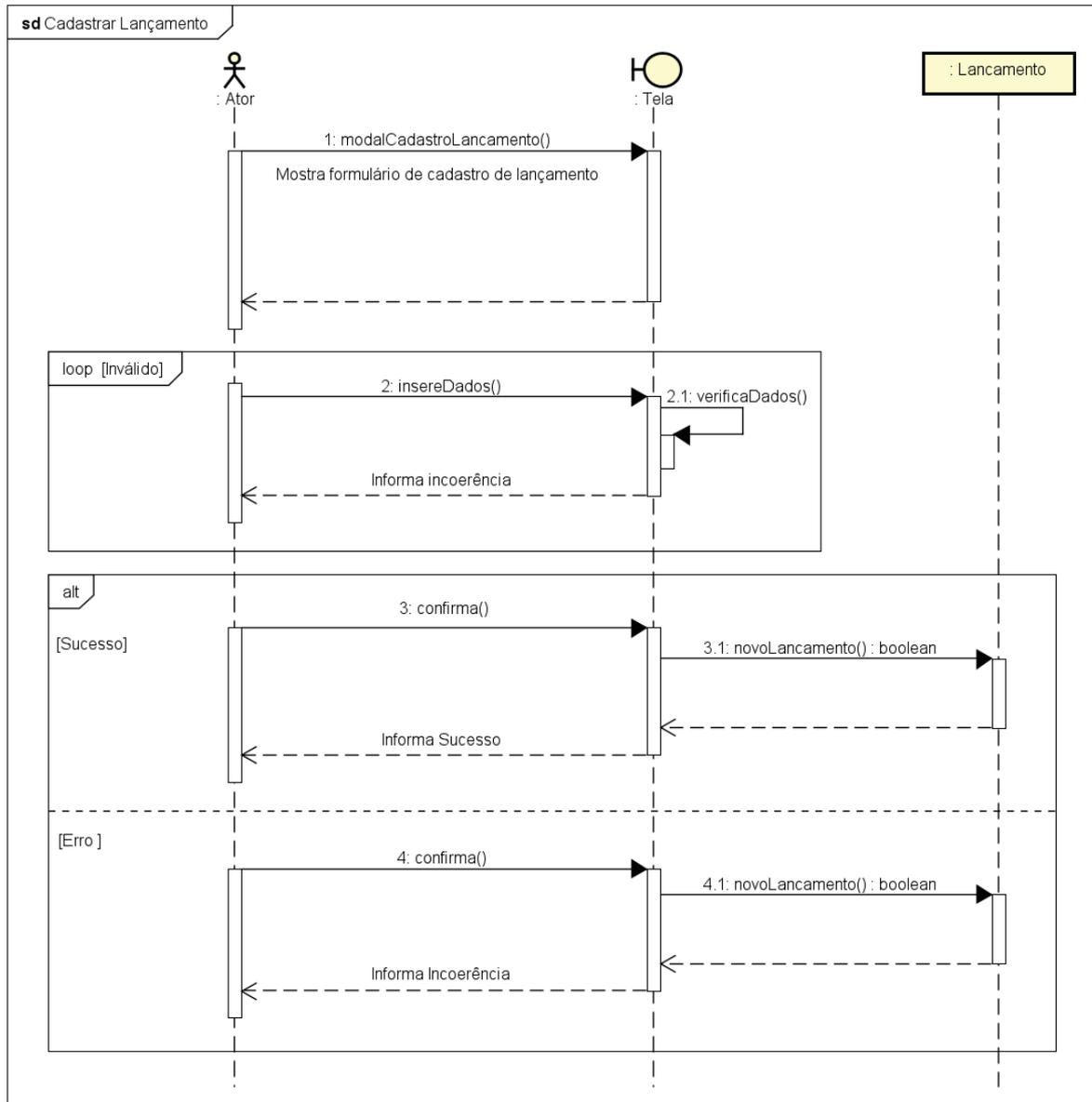
FIGURA 20 - DIAGRAMA DE CLASSES



FONTE: O autor (2023)

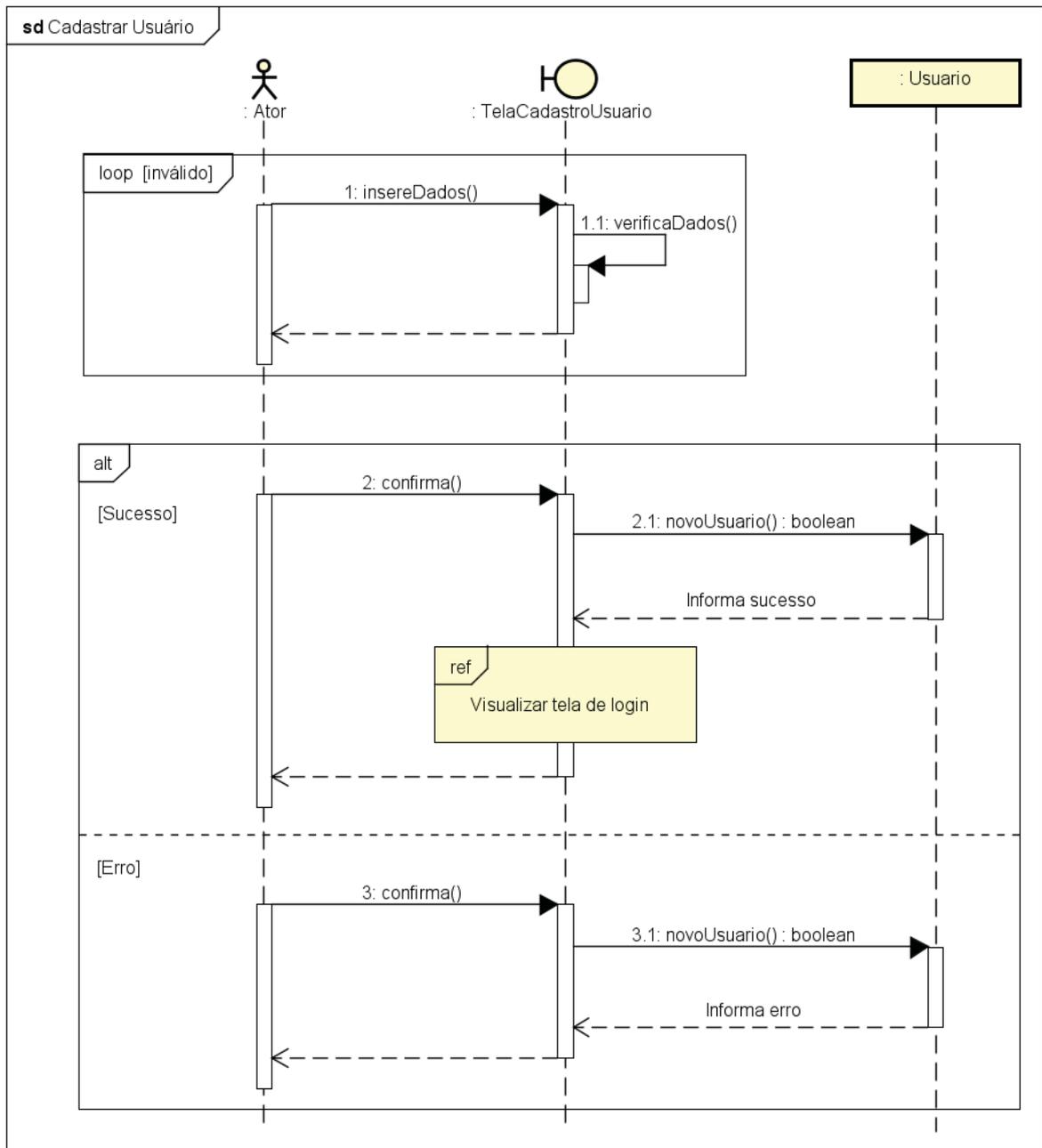
## 6 DIAGRAMAS DE SEQUÊNCIA

FIGURA 21 - DIAGRAMA DE SEQUÊNCIA CADASTRAR LANÇAMENTO



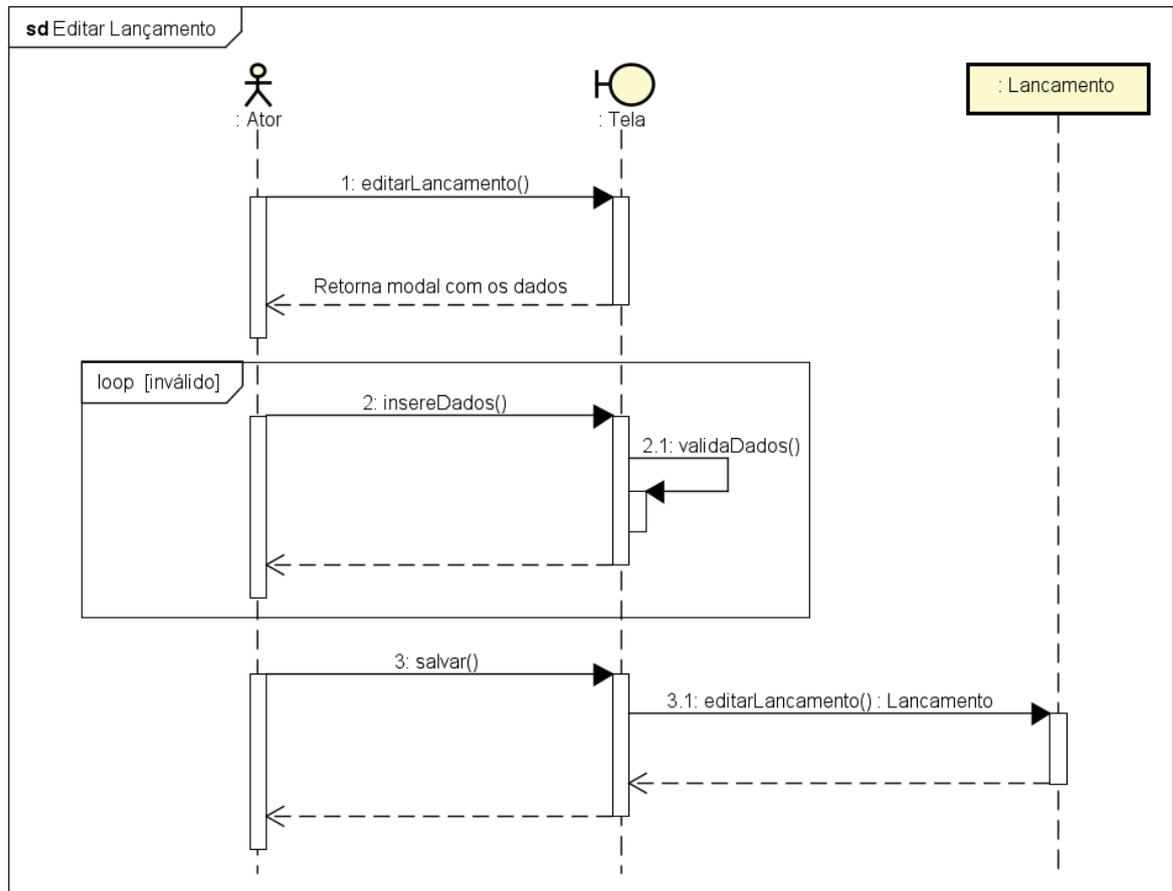
FONTE: O autor (2023)

FIGURA 22 - DIAGRAMA DE SEQUÊNCIA CADASTRAR USUÁRIO



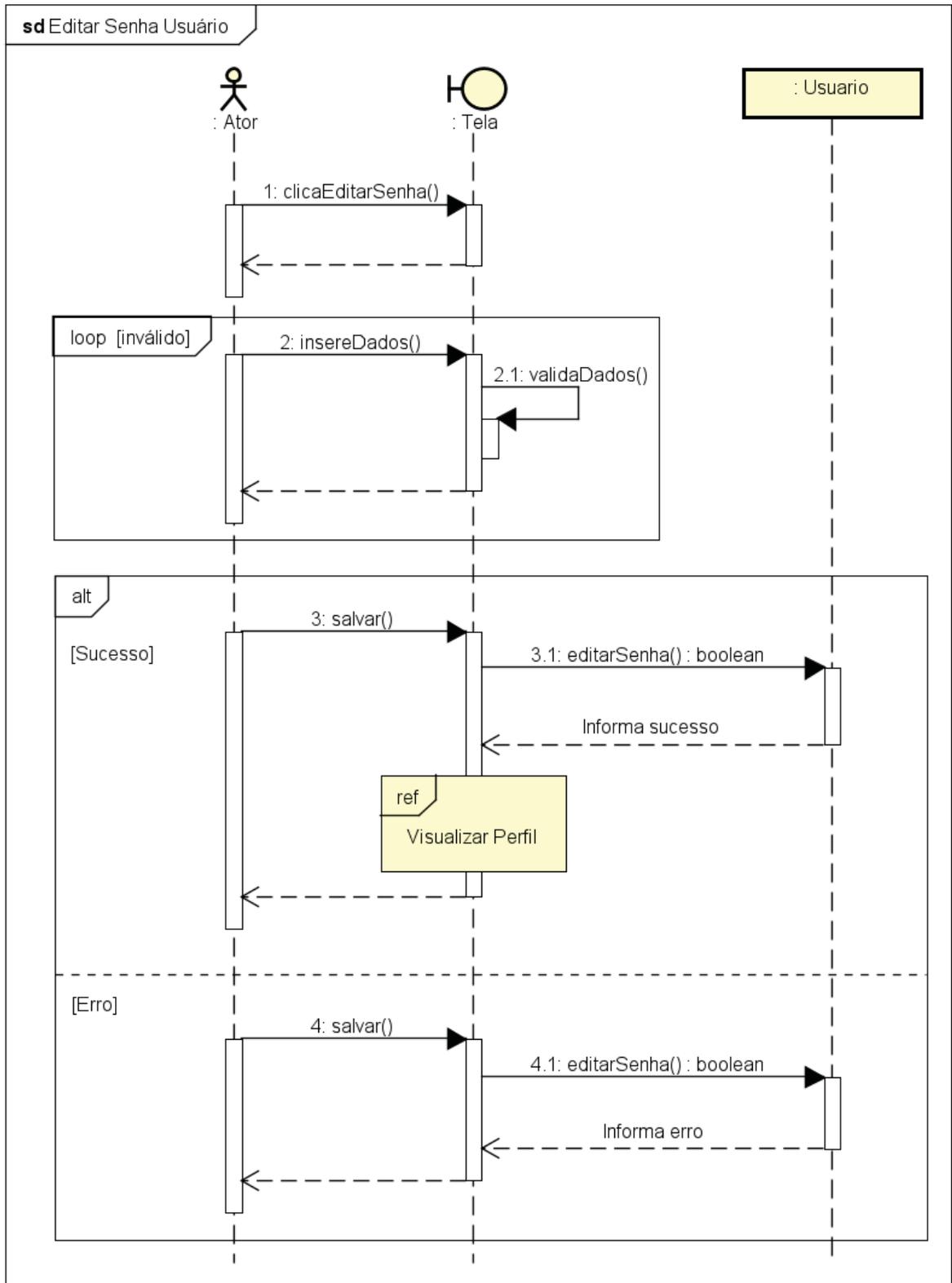
FONTE: O autor (2023)

FIGURA 23 - DIAGRAMA DE SEQUÊNCIA EDITAR LANÇAMENTO



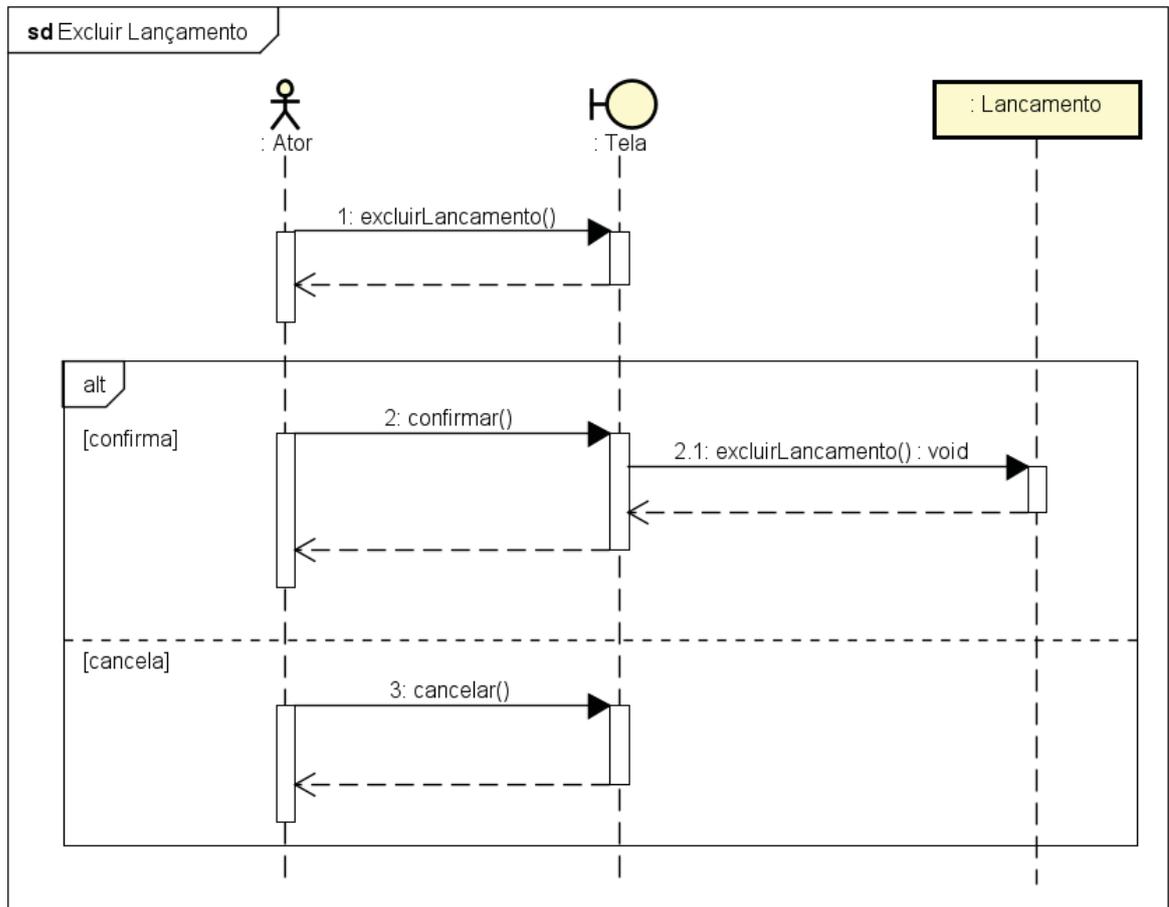
FONTE: O autor (2023)

FIGURA 24 - DIAGRAMA DE SEQUÊNCIA EDITAR SENHA DO USUÁRIO



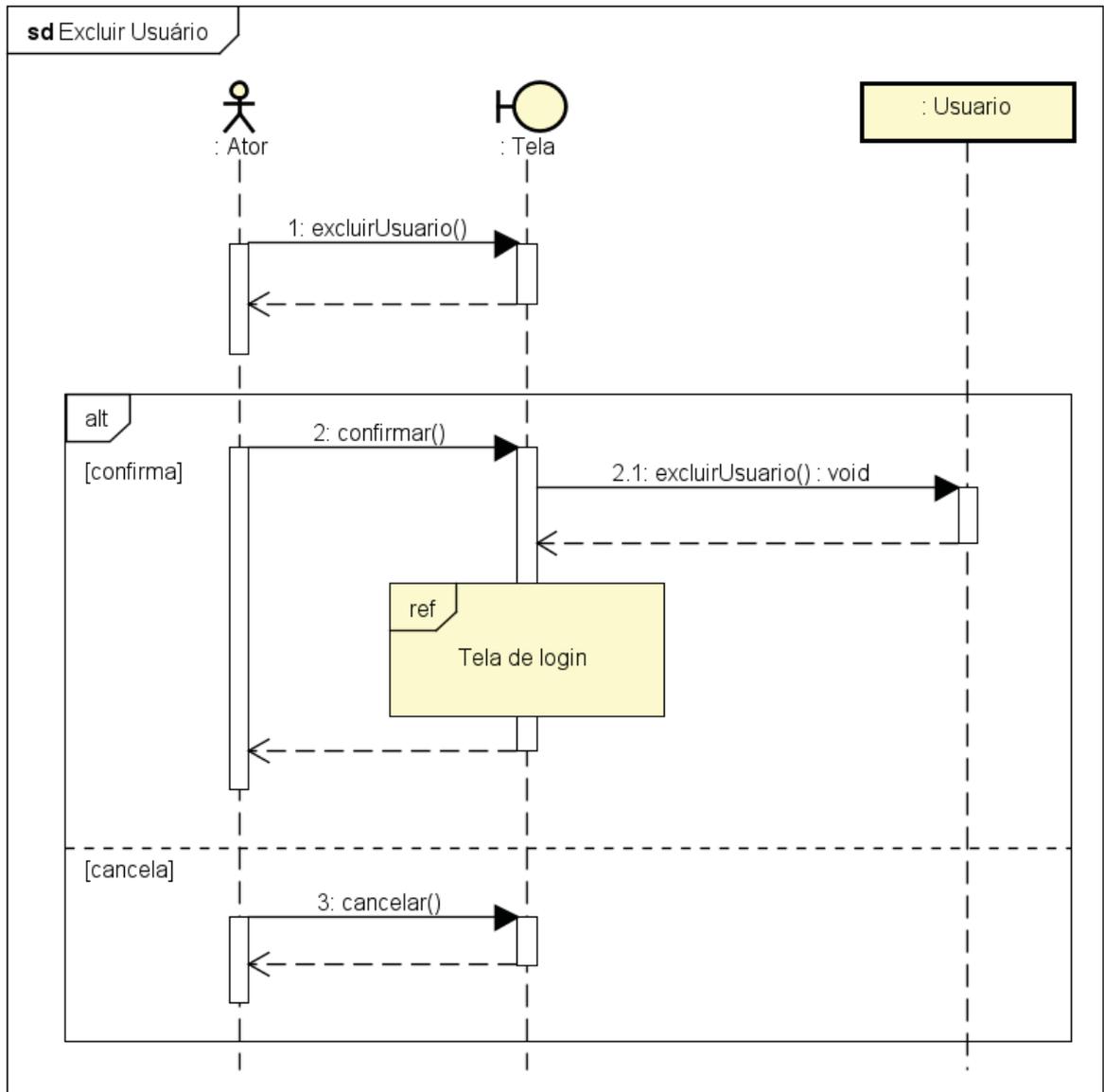
FONTE: O autor (2023)

FIGURA 25 - DIAGRAMA DE SEQUÊNCIA EXCLUIR LANÇAMENTO



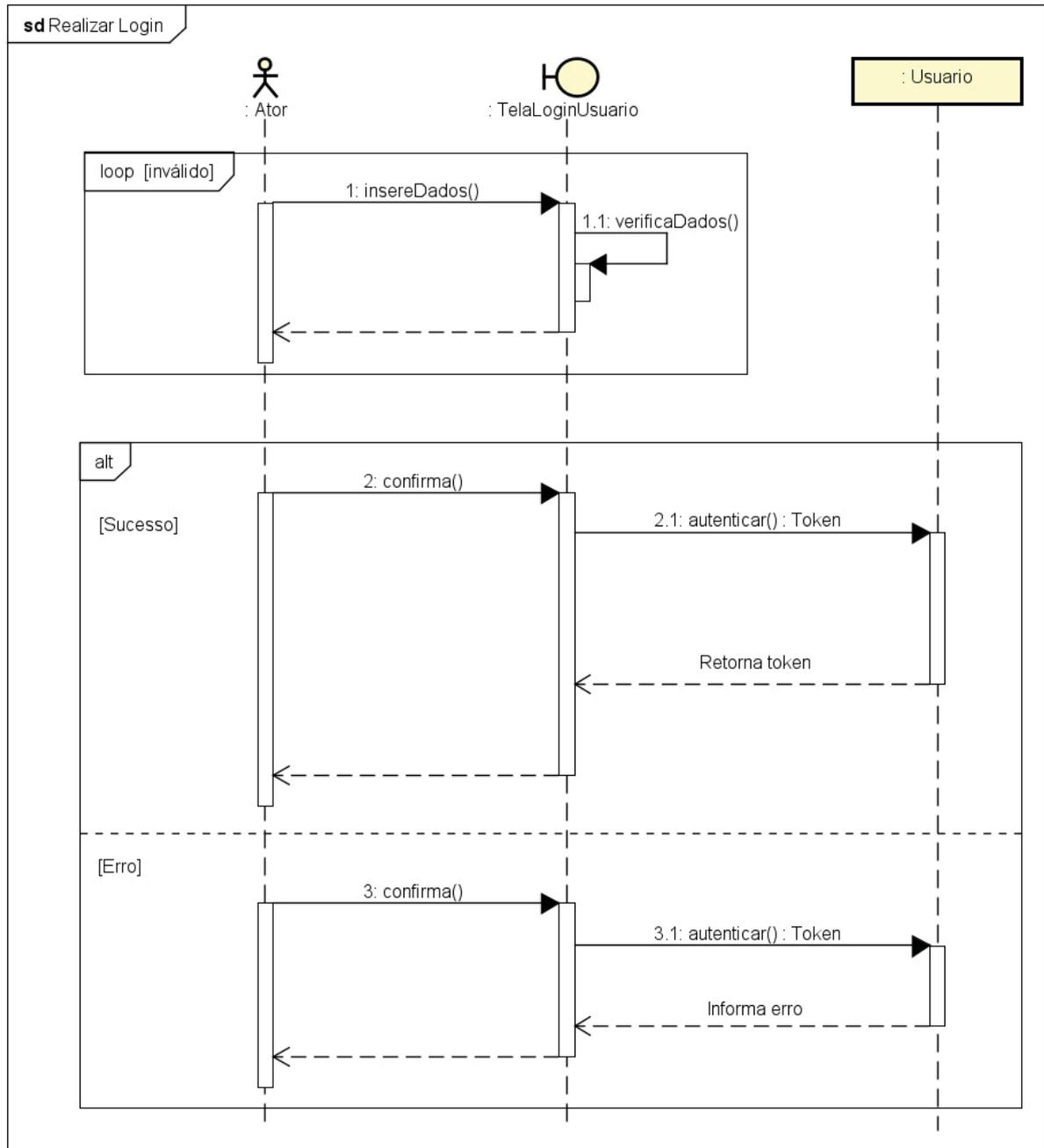
FONTE: O autor (2023)

FIGURA 26 - DIAGRAMA DE SEQUÊNCIA EXCLUIR USUÁRIO



FONTE: O autor (2023)

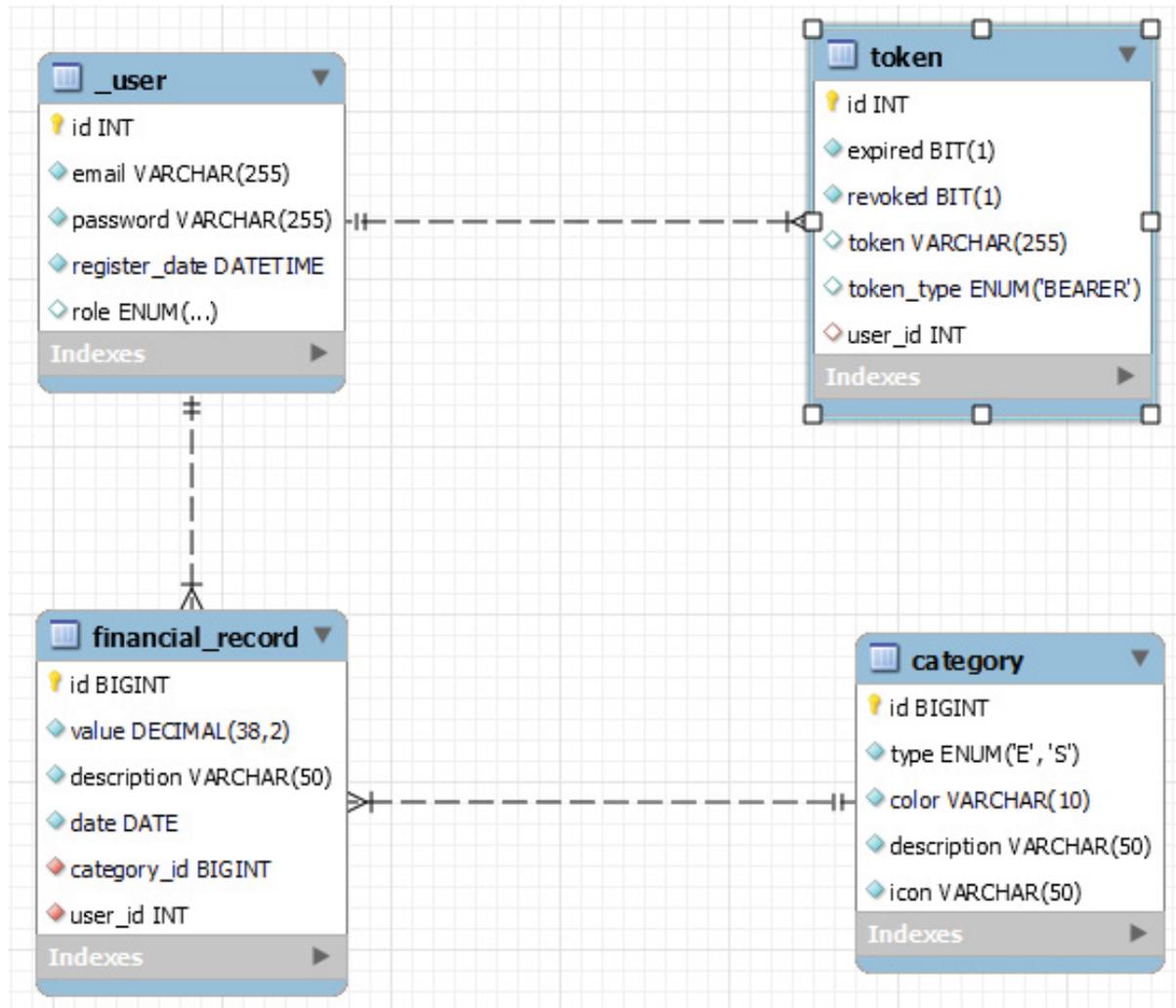
FIGURA 27 - DIAGRAMA DE SEQUÊNCIA REALIZAR LOGIN



FONTE: O autor (2023)

## 7 DIAGRAMA FÍSICO DO BANCO DE DADOS

FIGURA 28 - DIAGRAMA FÍSICO DO BANCO DE DADOS



FONTE: O autor (2023)

## REFERÊNCIAS

ACCENTURE. Banking in Brazil: Fintechs vs. Incumbents - A race to innovation. 2019. Disponível em: <https://www.accenture.com/acnmedia/PDF-96/Accenture-Banking-in-Brazil-Fintechs-vs-Incumbents-A-race-to-innovation.pdf>. Acesso em: 29 mar. 2023.

ANBIMA. Raio X do Investidor Brasileiro 2021. 2021. Disponível em: <https://www.anbima.com.br/raio-x/>. Acesso em: 29 mar. 2023.

ASSOCIAÇÃO BRASILEIRA DAS ENTIDADES DOS MERCADOS FINANCEIRO E DE CAPITAIS (ANBIMA). O raio X do investidor brasileiro. 2018. Disponível em: <https://cointimes.com.br/wp-content/uploads/2018/08/Relatorio-Raio-X-Investidor-PT.pdf>. Acesso em: 22 mar. 2023.

AUTH0 by Okta. O que é o OAuth 2.0, 2023. Disponível em: [O que é o OAuth 2.0 e o que ele faz por você? – Auth0](#). Acesso em: 07 abr. 2023.

BANCO INTERAMERICANO DE DESENVOLVIMENTO. Digital Finance and its Impact on Financial Inclusion and Stability: The Case of Brazil. 2019. Disponível em: <https://publications.iadb.org/publications/english/document/Digital-Finance-and-its-Impact-on-Financial-Inclusion-and-Stability-The-Case-of-Brazil.pdf>. Acesso em: 29 mar. 2023.

BASAK, S.; CHOWDHURY, S. K.; GHOSH, S. Flutter Framework for Cross-Platform Mobile Application Development: A Comparative Study. International Journal of Advanced Research in Computer Science and Software Engineering, v. 11, 2021.

CAMPOS, A. Percentual de famílias com dívidas chega a 65,6% em dezembro, diz CNC. 2020. Disponível em: <https://agenciabrasil.ebc.com.br/economia/noticia/2020-01/percentualde-familias-endividadas-chega-656-em-dezembro-diz-cnc>. Acesso em: 22 mar. 2023.

DEITEL, Paul; DEITEL, Harvey. Java: como programar. 6. ed. São Paulo: Pearson, 2011. Capítulo 1 - Introdução a computadores, programas e Java. Seção 1.5 - Java, páginas 17-19. Acesso em: 04 abr. 2023.

GOOGLE. Dart - Structured web programming. Disponível em: <https://dart.dev/>. Acesso em: 04 abr. 2023.

GOOGLE. Why Flutter uses Dart. Flutter.dev. Disponível em: <https://flutter.dev/docs/resources/inside-flutter>. Acesso em: 04 abr. 2023.

IETF, RFC 6749, 2012. Disponível em: <https://datatracker.ietf.org/doc/html/rfc6749>. Acesso em: 18 jun. 2023.

INFOMONEY. Thiago Nigro cresce 89% e vira principal perfil de finanças do Instagram. 2019. Disponível em: <https://www.infomoney.com.br/mercados/thiago->

[nigro-cresce-89-e-vira-principal-perfil-de-financas-do-instagram/](#). Acesso em: 22 mar. 2023.

JAGUARIBE, Helio. O Grande Salto para trás: A Decadência Econômica do Brasil. Editora Nova Fronteira, 1997. Acesso em: 22 mar. 2023.

LUNA, Francisco V.; KLEIN, Herbert S. História Econômica e Social do Brasil. Editora Saraiva, 2016. E-book. ISBN 9788547207786. Acesso em: 22 mar. 2023.

MILANI, André. MySQL guia do programador. São Paulo: Novatec, 2016. Acesso em: 04 abr. 2023.

PROVOS, Niels, MAZIÈRES, David. A Future-Adaptable Password Scheme. 1999. Disponível em: <http://publications.csail.mit.edu/lcs/pubs/pdf/20000111-1046.pdf>. Acesso em: 04 abr. 2023.

SPRING IO. Spring Boot - Simplify your Spring development. Disponível em: <https://spring.io/projects/spring-boot>. Acesso em: 04 abr. 2023.