



**ADELICIO DOS SANTOS BRITO  
IVAN SARDINHA DE BRITO JÚNIOR**

**FINANCONTROL - PLATAFORMA WEB PARA LANÇAMENTO E CONTROLE  
DE DESPESAS PESSOAIS**

Ji-Paraná

2024

**ADELICIO DOS SANTOS BRITO  
IVAN SARDINHA DE BRITO JÚNIOR**

**FINANCONTROL - PLATAFORMA WEB PARA LANÇAMENTO E CONTROLE  
DE DESPESAS PESSOAIS**

Trabalho de Conclusão de Curso  
apresentado à Banca Examinadora do  
Centro Universitário São Lucas, como  
requisito de aprovação para obtenção  
do título de Bacharel em Sistemas de  
Informação.

Orientador: Prof. Esp. Romário Vitorino  
Ferreira

Ji-Paraná

2024

**Dados Internacionais de Catalogação na Publicação - CIP**

B862f

Brito, Adelcio dos Santos.

FinanControl - Plataforma web para lançamento e controle de despesas pessoais. / Adelcio dos Santos Brito; Ivan Sardinha de Brito Júnior. – Ji-Paraná, 2024.  
43 p.; il.

Trabalho de Conclusão de Curso (Bacharel em Sistemas de Informação) – Centro Universitário São Lucas Ji-Paraná, 2024.

Orientador: Prof. Esp. Romário Vitorino Ferreira.

1. Despesas pessoais. 2. Controle financeiro. 3. Laravel 11. 4. Bootstrap. 5. Chart.js. 6. Plataforma web. 7. Gestão financeira. 8. Relatórios gráficos. 9. Planejamento financeiro. I. Brito Júnior, Ivan Sardinha de. II. Ferreira, Romário Vitorino. III. Título.

CDU 004.738.5:658.15

**Ficha Catalográfica Elaborada pelo Bibliotecário Giordani Nunes da Silva CRB 11/1125**

## **AGRADECIMENTOS**

Eu, Adelcio, gostaria de expressar minha mais profunda gratidão à minha esposa, que esteve ao meu lado em todos os momentos, oferecendo suporte emocional e sendo uma verdadeira parceira nesta jornada. Este trabalho é o resultado do esforço mútuo e do apoio que recebi ao longo de toda essa trajetória, incluindo o suporte dos professores, coordenador e demais pessoas que estiveram ao meu lado.

Eu, Ivan, sou grato a Deus por ter me fortalecido e trazido conforto nos momentos mais desafiadores ao longo desta jornada. Agradeço também aos meus pais, que sempre me incentivaram e apoiaram durante essa trajetória, e aos professores, que se dedicaram a transmitir os conhecimentos essenciais para minha carreira profissional.

Também queremos expressar nossa sincera gratidão ao professor especialista Romário Vitorino Ferreira, nosso orientador, pela paciência e disponibilidade em nos apoiar durante todo o processo de desenvolvimento deste projeto, assegurando sua conclusão bem-sucedida e nos guiando sempre na direção certa.

Por último, agradecemos a todos que, de alguma maneira, contribuíram para este momento único em nossas vidas.

## RESUMO

Este trabalho tem como objetivo o desenvolvimento de uma plataforma web destinada ao lançamento e controle de despesas pessoais, com o intuito de oferecer uma solução prática e eficiente para o gerenciamento financeiro. A aplicação permite que os usuários registrem suas despesas, organizando-as por categorias e visualizando relatórios gráficos interativos que ajudam a entender melhor seus hábitos de consumo e possibilitam a tomada de decisões mais informadas. A plataforma foi projetada para ser simples e de fácil uso, sem a opção de lançar receitas, focando exclusivamente no controle das saídas financeiras. O sistema foi desenvolvido utilizando o framework Laravel 11, que oferece uma estrutura segura e escalável para a criação do back-end, permitindo o fácil gerenciamento dos dados e interação com o banco de dados. Para o desenvolvimento da interface, foi adotado o Bootstrap, uma biblioteca de front-end que proporciona uma experiência amigável, garantindo acessibilidade e usabilidade. Para a geração dos relatórios gráficos, foi utilizada a biblioteca Chart.js, que permite a criação de gráficos interativos, de fácil interpretação, facilitando a análise das despesas por categoria e período. A proposta deste trabalho é criar uma ferramenta acessível e funcional que ajude os usuários a controlar seus gastos de forma intuitiva, sem a necessidade de conhecimentos avançados em finanças. Com o uso de relatórios gráficos dinâmicos, a plataforma visa proporcionar uma visão clara e detalhada das finanças pessoais, auxiliando na conscientização e no planejamento financeiro.

Palavras-Chave: Despesas pessoais. Controle financeiro. Laravel 11. Bootstrap. Chart.js. Plataforma web. Gestão financeira. Relatórios gráficos. Planejamento financeiro.

## **ABSTRACT**

This project aims to develop a web platform for tracking and managing personal expenses, with the goal of offering a practical and efficient solution for financial management. The application allows users to log their expenses, organizing them by categories and viewing interactive graphical reports that help them better understand their spending habits and make more informed decisions. The platform has been designed to be simple and easy to use, without the option to log income, focusing solely on tracking expenses. The system was developed using the Laravel 11 framework, which provides a secure and scalable structure for building the back-end, enabling easy data management and interaction with the database. For the development of the interface, Bootstrap was used, a front-end library that ensures a user-friendly experience, providing accessibility and usability. To generate the graphical reports, the Chart.js library was employed, which allows for the creation of interactive and easy-to-interpret charts, facilitating the analysis of expenses by category and time period. The purpose of this project is to create an accessible and functional tool that helps users manage their spending habits in an intuitive way, without the need for advanced financial knowledge. By using dynamic graphical reports, the platform aims to provide a clear and detailed overview of personal finances, assisting in financial awareness and planning.

**Keywords:** Personal expenses. Financial control. Laravel 11. Bootstrap. Chart.js. Web platform. Financial management. Graphical reports. Financial planning.

## LISTA DE FIGURAS

Figura 1 - Diagrama de Caso de Uso.....	25
Figura 2 - Diagrama de Classes.....	26
Figura 3 - Diagrama de Entidade e Relacionamento.....	27
Figura 4 - Diagrama de Sequência de Eventos Consultar Despesas.....	28
Figura 5 - Diagrama de Sequência de Eventos Lançar Nova Despesa.....	28
Figura 6 - Interface de Login.....	30
Figura 7 - Interface de Registro.....	30
Figura 8 - Interface da Página Inicial.....	31
Figura 9 - Barra de navegação da Página Inicial.....	32
Figura 10 - Interface de Grupos.....	32
Figura 11 - Interface de Subgrupos.....	33
Figura 12 - Interface de Formas de Pagamento.....	33
Figura 13 - Interface de Despesas.....	34
Figura 14 - Interface de Lançamento de Despesa.....	35
Figura 15 - Interface de Relatórios.....	35
Figura 16 - Interface de Perfil de Usuário.....	36

## SUMÁRIO

<b>INTRODUÇÃO</b>	9
<b>1. OBJETIVOS</b>	11
1.1. OBJETIVO GERAL	11
1.2. OBJETIVOS ESPECÍFICOS	11
<b>2. JUSTIFICATIVA</b>	12
<b>3. REFERENCIAL TEÓRICO</b>	13
3.1. PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE	13
3.2. FERRAMENTAS DE MODELAGEM	13
<b>3.2.1. Astah Community</b>	13
<b>3.2.2. MySQL Workbench</b>	14
3.3. LINGUAGENS DE DESENVOLVIMENTO WEB	14
<b>3.3.1. HTML</b>	14
<b>3.3.2. PHP</b>	15
<b>3.3.3. CSS</b>	15
<b>3.3.4. JavaScript</b>	16
3.4. BIBLIOTECAS E FRAMEWORKS	16
<b>3.4.1. Eloquent</b>	16
<b>3.4.2. Chart.js</b>	17
<b>3.4.3. bcrypt</b>	17
<b>3.4.4. Fortify</b>	17
<b>3.4.5. Bootstrap</b>	18
<b>3.4.6. Laravel</b>	18
3.5. LARAGON	19
3.6. COMPOSER	19
3.7. VISUAL STUDIO CODE	19
3.8. MYSQL	20
3.9. HEIDISQL	20
3.10. GITHUB	21
3.11. MAILHOG	21
3.12. TAIGA	21
<b>3.12.1. Scrum</b>	21
<b>4. MATERIAIS E MÉTODOS</b>	23



4.1. REQUISITOS FUNCIONAIS (RF).....	23
4.2. REQUISITOS NÃO FUNCIONAIS (RNF).....	24
4.3. DIAGRAMA DE CASO DE USO.....	25
4.4. DIAGRAMA DE CLASSES.....	26
4.5. DIAGRAMA DE ENTIDADE E RELACIONAMENTO.....	26
4.6. DIAGRAMA DE SEQUÊNCIA DE EVENTOS.....	27
<b>5. RESULTADOS.....</b>	<b>29</b>
5.1. SOBRE A PLATAFORMA WEB FINANCONTROL.....	29
5.2. INTERFACES DE LOGIN, REGISTRO E RECUPERAÇÃO DE SENHA.....	29
5.3. INTERFACE DA PÁGINA INICIAL.....	31
<b>5.3.1. Barra de Navegação da Página Inicial.....</b>	<b>31</b>
5.4. INTERFACE DE GRUPOS.....	32
5.5. INTERFACE DE SUBGRUPOS.....	33
5.6. INTERFACE DE FORMAS DE PAGAMENTO.....	33
5.7. INTERFACE DE DESPESAS.....	34
5.8. INTERFACE DE LANÇAMENTO DE DESPESA.....	34
5.9. INTERFACE DE RELATÓRIOS.....	35
5.10. INTERFACE DE PERFIL DE USUÁRIO.....	36
<b>CONSIDERAÇÕES FINAIS.....</b>	<b>37</b>
<b>REFERÊNCIAS.....</b>	<b>38</b>
<b>GLOSSÁRIO.....</b>	<b>41</b>

## INTRODUÇÃO

Nos dias de hoje, o controle financeiro pessoal se tornou uma necessidade fundamental para muitas pessoas, especialmente diante das dificuldades econômicas e da crescente complexidade das finanças. A capacidade de monitorar gastos, identificar padrões de consumo e tomar decisões informadas sobre o orçamento pessoal é crucial para a saúde financeira de qualquer indivíduo. Nesse contexto, a tecnologia tem se mostrado uma ferramenta poderosa para facilitar esse controle, proporcionando soluções práticas e acessíveis.

O objetivo deste trabalho é desenvolver uma plataforma web voltada para o lançamento e controle de despesas pessoais. A plataforma será focada exclusivamente no acompanhamento das saídas financeiras dos usuários, sem a inclusão de funcionalidades para registro de receitas. A proposta é fornecer uma ferramenta simples, intuitiva e acessível, que permita aos usuários registrar suas despesas, categorizá-las e visualizar relatórios gráficos interativos. Esses relatórios ajudarão os usuários a compreender melhor seus hábitos de consumo, oferecendo uma visão clara e detalhada das finanças pessoais e contribuindo para a tomada de decisões mais informadas.

A escolha do Laravel 11 para o desenvolvimento do backend do sistema visa garantir uma estrutura robusta, segura e escalável, permitindo o gerenciamento eficiente dos dados e a integração com o banco de dados. O front-end será desenvolvido utilizando o framework Bootstrap, que oferece uma interface responsiva, adaptando-se às diferentes resoluções de tela e garantindo uma experiência de uso otimizada. Para a geração de relatórios gráficos dinâmicos, será utilizada a biblioteca Chart.js, que possibilita a criação de gráficos interativos, facilitando a visualização das despesas por categorias e ao longo do tempo.

Este trabalho visa, assim, criar uma solução prática para o controle financeiro pessoal, utilizando tecnologias modernas e de fácil implementação. A ferramenta proposta busca ser uma aliada na conscientização financeira dos usuários, permitindo um acompanhamento mais preciso dos seus gastos e contribuindo para a realização de um planejamento financeiro mais eficaz.

O desenvolvimento deste projeto será realizado com base em uma abordagem de design centrado no usuário, com o objetivo de tornar a plataforma o mais intuitiva possível, atendendo às necessidades do público-alvo sem exigir

conhecimentos avançados em finanças. Com isso, a plataforma se posiciona como uma ferramenta útil para quem deseja melhorar sua organização financeira e obter maior controle sobre seus gastos.

## 1. OBJETIVOS

A seguir, serão apresentados o objetivo geral e os objetivos específicos propostos neste trabalho.

### 1.1. OBJETIVO GERAL

Este trabalho de conclusão de curso tem como objetivo desenvolver uma plataforma web acessível, intuitiva e eficiente, voltada para o lançamento e controle de despesas pessoais. A plataforma será capaz de fornecer aos usuários uma solução prática para gerenciar suas finanças, permitindo o registro de despesas, categorização dos gastos e visualização de relatórios gráficos dinâmicos, com o intuito de promover o controle financeiro e a conscientização sobre hábitos de consumo, mesmo para aqueles que não possuem conhecimentos avançados em finanças.

### 1.2. OBJETIVOS ESPECÍFICOS

- Facilitar a usabilidade da plataforma, garantindo uma interface simples e intuitiva para usuários sem experiência prévia em ferramentas financeiras;
- Oferecer uma solução acessível e de baixo custo para o controle de despesas pessoais;
- Proporcionar uma visualização clara e dinâmica das despesas por meio de gráficos interativos;
- Desenvolver um sistema de categorização de despesas que permita uma organização eficiente e prática dos gastos;
- Permitir a geração de relatórios financeiros mensais, proporcionando uma visão detalhada das finanças pessoais ao longo do tempo.

## 2. JUSTIFICATIVA

A demanda por soluções que auxiliem no controle das despesas pessoais tem crescido significativamente, especialmente em um cenário onde a gestão eficiente das despesas é fundamental para o bem-estar econômico dos indivíduos. Contudo, muitos ainda enfrentam desafios ao tentar gerenciar seus gastos de forma eficaz, seja pela falta de conhecimento em finanças ou pela complexidade das ferramentas disponíveis.

A proposta desta plataforma web é suprir essa necessidade, oferecendo aos usuários uma interface simples e acessível para o registro de despesas e a visualização de relatórios gráficos. Como destaca Eker (2006, p. 142), “Tudo se resume ao seguinte: ou você controla o seu dinheiro, ou ele o controlará. Para controlar o dinheiro, você tem que administrá-lo.”

O objetivo é criar uma ferramenta que atenda tanto iniciantes quanto aqueles com mais experiência em finanças pessoais, proporcionando uma visão clara e objetiva de seus gastos, sem a exigência de conhecimentos aprofundados ou o uso de recursos complicados. Esta abordagem busca facilitar a gestão das despesas, promover a conscientização sobre os hábitos de consumo e, assim, contribuir para a melhoria da saúde financeira dos usuários.

### **3. REFERENCIAL TEÓRICO**

Para a criação da plataforma web FinanControl, foi fundamental o uso de diversos recursos, abrangendo desde a definição dos requisitos até o projeto, implementação e lançamento da plataforma. Esta seção descreve os métodos, ferramentas, plugins, extensões, bibliotecas e frameworks que desempenharam um papel crucial na conclusão das etapas de desenvolvimento do software.

#### **3.1. PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE**

Todo processo envolve uma sequência de ações com o objetivo de atingir uma meta específica. Dessa maneira, ele pode ser aplicado para projetar, transformar, manter, criar, entre outras funções. Em resumo, qualquer atividade relacionada à execução de tarefas geralmente é considerada um procedimento (ENGHOLM, 2010).

De acordo com Silva, B. C. C. (2006), a norma IEEE 1219 define as seguintes fases no processo de desenvolvimento de software: Identificação, que envolve a coleta de requisitos e a definição das prioridades; Análise, onde é realizado um estudo de viabilidade das funcionalidades; Projeto, que se refere à modelagem do programa e do banco de dados; Implementação, que abrange o desenvolvimento do código-fonte com base no projeto definido; Testes, período dedicado à identificação de falhas e à aceitação por parte do cliente; e Entrega, que consiste na implantação final do sistema.

#### **3.2. FERRAMENTAS DE MODELAGEM**

Nesta seção, serão apresentadas as ferramentas utilizadas na fase de planejamento da plataforma web FinanControl. Essas ferramentas foram essenciais para a construção de modelos que representam a interação do usuário com o sistema, os relacionamentos entre objetos, a definição da estrutura do banco de dados e o comportamento das funcionalidades do software.

##### **3.2.1. Astah Community**

O Astah Community é uma ferramenta de modelagem UML (Unified Modeling Language) criada para atender à demanda por uma aplicação mais acessível, que auxilie os desenvolvedores a aumentarem sua produtividade, inovando na maneira como esses profissionais trabalham. Ele organiza a estrutura lógica das aplicações em formatos visuais de fácil compreensão. A ferramenta foi desenvolvida pela empresa japonesa Change Vision, liderada pelo CTO Kenji Hiranabe, engenheiro de software com mais de 20 anos de experiência (HIRANABE, [s.d.]).

### **3.2.2. MySQL Workbench**

O MySQL Workbench é uma plataforma integrada desenvolvida para facilitar o design, desenvolvimento e administração de bancos de dados MySQL. Ele oferece uma série de ferramentas para modelagem de dados, desenvolvimento de SQL, administração de servidores e manutenção de bancos de dados. Com o MySQL Workbench, os usuários podem criar diagramas ER (Entidade-Relacionamento), editar consultas SQL visualmente, realizar backups e restaurar dados, além de monitorar o desempenho do servidor. A ferramenta é compatível com diferentes sistemas operacionais e é amplamente utilizada por desenvolvedores e administradores de banco de dados para gerenciar e otimizar o uso do MySQL de maneira eficiente e acessível (MYSQL, 2019).

## **3.3. LINGUAGENS DE DESENVOLVIMENTO WEB**

Esta seção apresenta os conceitos relacionados às linguagens utilizadas na criação da plataforma web, as quais fazem parte da fase de implementação e desempenham diferentes funções dentro do código fonte. Segundo Fernandes, M. C. (2021, p. 11): "Qualquer linguagem que o leitor possa vir a imaginar provavelmente já foi usada comercialmente para o desenvolvimento de sites em algum projeto".

### **3.3.1. HTML**

O HTML tornou-se o padrão para a criação de páginas na internet, sendo

utilizado juntamente com as ferramentas CSS e JavaScript. Através de suas tags, é possível informar aos navegadores como cada elemento do site deve ser exibido. Ele possibilita a criação de documentos estruturados com títulos, parágrafos, listas, links, tabelas, formulários, entre outros elementos, nos quais podem ser incorporadas imagens e objetos, como animações ou vídeos (FLATSCHART, 2011).

### **3.3.2. PHP**

O PHP é uma linguagem de código aberto, criada a partir da linguagem C, com foco específico no desenvolvimento web. Sua origem remonta à necessidade de Rasmus Lerdorf de monitorar as visitas em sua página pessoal. Inicialmente, o PHP foi utilizado em uma aplicação cliente-servidor, sendo integrado ao HTML das páginas web. Rasmus disponibilizou o código-fonte do PHP, permitindo que programadores pudessem modificá-lo e melhorar o pré-processador de hipertexto continuamente (CONVERSE e PARK, 2003).

### **3.3.3. CSS**

De acordo com Tableless (2006), o CSS (Cascading Style Sheets), ou Folhas de Estilo em Cascata em português, é uma ferramenta que possibilita aos desenvolvedores web personalizar o visual das páginas, aprimorando a apresentação para os usuários. A ideia do CSS foi proposta em 1994 por Hakon Lie, com o intuito de simplificar a programação de sites, pois, antes de sua criação, era necessário escrever uma grande quantidade de código HTML para obter resultados semelhantes aos do CSS. Desenvolvido pela W3C em 1995, o CSS foi ganhando popularidade, especialmente entre 1997 e 1999, quando passou a ser amplamente adotado pelos desenvolvedores.

A principal razão de seu sucesso está na capacidade de separar o conteúdo de uma página de seu estilo visual, o que permite alterar a aparência sem afetar o conteúdo. Essa separação é feita por meio de arquivos distintos, o que aumenta a flexibilidade e reduz os riscos de comprometer o conteúdo ao fazer ajustes visuais. Além disso, o CSS permite escrever uma quantidade menor de código, pois é possível agrupar várias tags ou classes em um único arquivo para aplicar o estilo de forma mais eficiente (TABLELESS, 2006).



### **3.3.4. JavaScript**

O JavaScript é uma linguagem de programação de alto nível, com tipagem dinâmica e fraca, conhecida por sua grande potência e presença em praticamente todos os sites na internet. Originalmente desenvolvida pela Netscape e, posteriormente, adquirida pela Sun Microsystems, foi criada com o objetivo de atender às necessidades da web (SILVA, M. S., 2010).

Sua popularidade se deve ao fato de estar incorporada em todos os navegadores modernos, que incluem um interpretador JavaScript, o que permitiu que a linguagem evoluísse para uma ferramenta de programação versátil, robusta e eficiente. Hoje, o JavaScript faz parte das três tecnologias essenciais para o desenvolvimento web, ao lado do HTML e do CSS (FLANAGAN, 2012).

## **3.4. BIBLIOTECAS E FRAMEWORKS**

Além das linguagens utilizadas no desenvolvimento web, foi crucial o emprego de recursos que facilitam o processo de implementação, melhorando a qualidade da plataforma, reduzindo custos e acelerando o tempo de produção.

Nesse contexto, foram adotados os seguintes recursos para auxiliar na programação: Bibliotecas, que são conjuntos de códigos criados para resolver problemas específicos; e Frameworks, que são estruturas formadas por um conjunto de códigos genéricos que facilitam o desenvolvimento de sistemas e aplicações. (BORGES, 2023).

### **3.4.1. Eloquent**

A documentação oficial do Laravel descreve o Eloquent como um ORM (Object-Relational Mapping), ou Mapeamento Objeto-Relacional em português, uma camada de abstração que simplifica a interação com o banco de dados, utilizando modelos (models) para manipular as tabelas de forma mais intuitiva. Esse recurso é amplamente utilizado no Laravel, pois possibilita a execução de consultas completas ao banco de dados de forma prática e com um código reduzido (LARAVEL, [s.d.]).

### 3.4.2. Chart.js

O Chart.js é uma biblioteca JavaScript popular e de código aberto, ideal para criar gráficos interativos de forma simples e eficiente. Oferece uma variedade de tipos de gráficos, é altamente personalizável com plugins, e utiliza renderização em canvas para garantir alta performance, especialmente com grandes volumes de dados. Essa biblioteca é compatível com frameworks como React, Vue, Angular e Laravel, tem uma documentação detalhada e é amplamente apoiada pela comunidade. Além disso, sua arquitetura permite otimização de desempenho e redução do tamanho do arquivo gerado que contém todo o código necessário para a aplicação, através de uma técnica chamada *tree-shaking*, que elimina código não utilizado, diminuindo o tamanho do arquivo final e melhorando o tempo de carregamento (CHART.JS, [s.d.]).

### 3.4.3. bcrypt

O bcrypt é uma biblioteca de *hashing* de senhas, ou seja, ela transforma senhas em uma sequência de caracteres fixa (chamada *hash*) usando *Blowfish*, que é um algoritmo de criptografia que, no caso do bcrypt, é usado com múltiplas iterações para tornar as senhas mais difíceis de serem quebradas por ataques de força bruta, conforme documentação oficial. O bcrypt também gera um *salt*, que é um valor aleatório único para cada senha, garantindo que senhas iguais resultem em *hashes* diferentes. Além disso, o bcrypt permite configurar o número de iterações, equilibrando segurança e desempenho. A biblioteca oferece funções assíncronas para evitar bloquear o servidor em sistemas com alta carga e é compatível com várias versões do Node.js. A versão mínima recomendada para garantir segurança é a 5.0.0, que resolve problemas de vulnerabilidades em versões anteriores (BCRYPT, 2023).

### 3.4.4. Fortify

Segundo a documentação oficial do Laravel, o Laravel Fortify é uma implementação de back-end para autenticação em Laravel, sem depender de interface de usuário. Ele fornece as rotas e controladores necessários para

funcionalidades como login, registro de usuário, redefinição de senha, verificação de e-mail e muito mais. Porém, ao contrário de outros pacotes, o Fortify não inclui views ou interfaces prontas, permitindo que você use sua própria interface front-end. Com ele, é possível configurar quais funcionalidades de autenticação deseja ativar. O Fortify é altamente personalizável, permitindo ajustes no fluxo de autenticação, tratamento de erros e redirecionamentos. Ele também inclui recursos como autenticação de dois fatores e limitadores de tentativas de login, aumentando a segurança da aplicação (LARAVEL, [s.d.]b).

### 3.4.5. Bootstrap

De acordo com a página oficial do Bootstrap (s.d.), o Bootstrap é o framework CSS gratuito e de código aberto mais conhecido para o desenvolvimento de websites. Quanto ao seu lançamento, é informado que:

O Bootstrap foi criado no Twitter em meados de 2010 por @mdo e @fat. Antes de ser uma estrutura de código aberto, o Bootstrap era conhecido como Twitter Blueprint. Após alguns meses de desenvolvimento, o Twitter realizou sua primeira Hack Week e o projeto explodiu à medida que os desenvolvedores de todos os níveis evoluíram sem nenhuma orientação externa. Ele serviu como guia de estilo para o desenvolvimento de ferramentas internas na empresa, por mais de um ano até ter o seu lançamento público, e continua a fazê-lo até hoje. (BOOTSTRAP, [s.d.]).

Este framework passou por duas grandes atualizações, sendo reestruturado nas versões 2 e 3. Na versão 2, a responsividade foi adicionada de forma opcional, permitindo que o site se adaptasse à resolução da janela. Já na versão 3, a responsividade foi implementada por padrão, adotando a abordagem *mobile first* (BOOTSTRAP, [s.d.]).

### 3.4.6. Laravel

O Laravel é um framework PHP voltado para simplificar o desenvolvimento de aplicações web, proporcionando uma codificação mais ágil e eficiente. Definido como um "framework com sintaxe expressiva e elegante", ele permite que os desenvolvedores foquem na criação do projeto sem se preocupar com detalhes menores, conforme descrito em seu site oficial (OTWELL, 2015).

Adotando a arquitetura MVC para melhorar a organização do código e facilitar sua manutenção, o Laravel oferece diversos recursos prontos para uso, como telas funcionais que podem ser personalizadas de acordo com as necessidades do projeto. Com uma excelente documentação e uma comunidade ativa, o Laravel se destaca como uma ferramenta poderosa para o desenvolvimento web rápido e de alta qualidade (GABARDO, 2017).

### 3.5. LARAGON

Laragon é um ambiente de desenvolvimento portátil, rápido e leve, ideal para PHP, Node.js, Python, Java, Go, Ruby, entre outras tecnologias. Ele oferece uma configuração simples e automática, com a possibilidade de adicionar facilmente novas versões de servidores e bancos de dados. Com uma interface fácil de usar e recursos como URLs personalizadas, isolamento do sistema operacional e a capacidade de ser movido entre diferentes dispositivos, Laragon é uma ferramenta poderosa e flexível para desenvolvimento web moderno. Além disso, ele utiliza poucos recursos de RAM e facilita tarefas como ativar extensões PHP ou instalar os Sistemas de Gerenciamento de Conteúdo, conhecidos como CMS (Content Management Systems) com um clique (LARAGON, [s.d.]).

### 3.6. COMPOSER

O Composer é uma ferramenta voltada para o gerenciamento de dependências na linguagem PHP. Ao contrário de gerenciadores de pacotes como o pip ou npm, o Composer, segundo sua documentação oficial, administra apenas as dependências dentro de um projeto específico, realizando a instalação ou atualização delas, mas sem afetar o ambiente global. Ou seja, por padrão, as dependências são instaladas exclusivamente no projeto em que estão sendo utilizadas (COMPOSER, [s.d.]).

### 3.7. VISUAL STUDIO CODE

De acordo com a documentação oficial, o Visual Studio Code (ou VS Code) é um editor de código poderoso e desenvolvido pela Microsoft. Ele é multi

plataforma, open-source e gratuito. Classificado como uma IDE (Ambiente de Desenvolvimento Integrado), o VS Code é projetado para facilitar o trabalho dos desenvolvedores, oferecendo uma plataforma robusta para escrever e editar código. O editor pode ser personalizado por meio de plugins, permitindo suporte para diversas linguagens de programação, além de recursos adicionais para aumentar a produtividade. Seu design é moderno, com uma interface intuitiva e limpa, proporcionando uma experiência agradável para o usuário (VISUAL STUDIO CODE, 2023).

### 3.8. MYSQL

O MySQL desempenhou um papel fundamental no armazenamento de informações relacionadas aos dados em geral. De acordo com Milani (2006):

MySQL é um servidor e gerenciador de banco de dados (SGBD) relacional, de licença dupla (sendo uma delas de software livre), projetado inicialmente para trabalhar com aplicações de pequeno e médio portes, mas hoje atendendo a aplicações de grande porte e com mais vantagens do que seus concorrentes. Possui todas as características que um banco de dados de grande porte precisa, sendo reconhecido por algumas entidades como o banco de dados open source com maior capacidade para concorrer com programas similares de código fechado, tais como SQL Server (da Microsoft) e Oracle. (MILANI, 2006, p. 22).

O MySQL foi criado por David Axmark, Allan Larsson e Michael "Monty" Widenius em 1994. Em 2008, a empresa responsável pelo desenvolvimento do MySQL, a MySQL AB, foi comprada pela Sun Microsystems (MILANI, 2006).

### 3.9. HEIDISQL

O HeidiSQL é um software gratuito e intuitivo voltado para quem trabalha com bancos de dados. Ele permite a conexão a diversas plataformas, como MariaDB, MySQL, Microsoft SQL, PostgreSQL, SQLite, Interbase e Firebird. Com o HeidiSQL, é possível editar dados e estruturas desses bancos de dados de forma simples. Criado em 2002 por Ansgar, continua sendo uma ferramenta popular, especialmente para MariaDB e MySQL, e é ativamente mantido (HEIDISQL, [s.d.]).

### 3.10. GITHUB

De acordo com a documentação oficial do GitHub, a plataforma é destinada à hospedagem de código, controle de versões e colaboração entre desenvolvedores. Com mais de 94 milhões de usuários, 4 milhões de organizações e 330 milhões de repositórios ativos, o GitHub permite gerenciar projetos e integrar contribuições de outros colaboradores no desenvolvimento principal. A plataforma é amplamente adotada por grandes empresas globais, como Pinterest, Stripe, KPMG, Mercedes-Benz, P&G e Telus (GITHUB, [s.d.]).

### 3.11. MAILHOG

O MailHog é uma ferramenta de desenvolvimento que atua como um servidor de e-mail local, capturando e-mails enviados durante o desenvolvimento de aplicações e evitando que eles cheguem a destinatários reais. Ele fornece uma interface web simples onde os desenvolvedores podem visualizar os e-mails de teste, sendo especialmente útil em frameworks como o Laravel, onde o envio de e-mails é comum. Com o MailHog, é possível testar e-mails de forma segura e eficiente em ambientes de desenvolvimento, sem o risco de afetar usuários reais (BOLSANELLO, 2024).

### 3.12. TAIGA

O Taiga é uma plataforma web voltada para o gerenciamento de projetos, integrando funcionalidades das metodologias ágeis Scrum e Kanban. Reconhecida como a melhor ferramenta *agile* de 2015, ela oferece diversos recursos, como múltiplos fluxos de trabalho, a capacidade de listar problemas do projeto, criar definições para a equipe no dicionário, controlar o andamento das tarefas e importar projetos de outras ferramentas. Essas e outras funcionalidades tornam o Taiga uma opção robusta para gerenciamento de projetos ágeis (TAIGA, 2015).

#### 3.12.1. Scrum

O gerenciamento do projeto foi realizado utilizando a metodologia Scrum,

que foi implementada por meio da ferramenta Taiga. De acordo com DRUMOND (s.d.), o Scrum é uma abordagem que otimiza o fluxo de trabalho, resultando em uma maior eficiência ao longo do processo de desenvolvimento.

Composta por papéis (como proprietário do produto, Scrum Master e equipe de desenvolvimento), artefatos (como backlogs e incrementos) e cerimônias (como planejamento de *sprint*, reuniões diárias, revisão de *sprint* e retrospectivas), a metodologia enfatiza transparência, feedback constante e melhoria contínua. Fundamentada no empirismo e no pensamento enxuto, o Scrum se adapta às necessidades do projeto e da equipe, sendo ideal para ambientes de desenvolvimento complexos (DRUMOND, [s.d.]).

## 4. MATERIAIS E MÉTODOS

Esta seção apresenta os conceitos e objetivos da plataforma, além de fornecer detalhes sobre os diagramas que ilustram a interação entre o usuário e a aplicação, a definição das entidades e o comportamento esperado das funcionalidades. Esses diagramas oferecem uma visão clara e visual do funcionamento do software, facilitando o processo de desenvolvimento e implementação.

### 4.1. REQUISITOS FUNCIONAIS (RF)

Os requisitos funcionais descrevem o que o software deve fazer para atender às necessidades dos usuários e partes interessadas. Eles definem as funcionalidades e operações que o sistema deve executar, incluindo as interações do usuário e as respostas do sistema a determinadas ações (COOPERSYSTEM, 2024).

Abaixo estão listados os requisitos funcionais que a aplicação deve cumprir para atender tanto aos usuários convidados quanto aos administradores:

- RF1: A aplicação deve permitir que o usuário se registre e faça login com um sistema de autenticação para garantir a segurança e a privacidade dos dados financeiros pessoais;
- RF2: O usuário deve poder cadastrar, editar e excluir grupos, que será a classificação responsável por categorizar as despesas;
- RF3: O usuário deve poder cadastrar, editar e excluir subgrupos, que será a subclassificação das despesas;
- RF4: O usuário deve poder cadastrar, editar e excluir formas de pagamentos, que serão utilizadas nos lançamentos das despesas;
- RF5: O usuário deve poder lançar, editar e excluir suas despesas, incluindo informações como data, categoria e tipo do lançamento, valor, forma de pagamento e descrição;
- RF6: A aplicação deve permitir ao usuário visualizar suas despesas agrupadas por categorias, com opções de filtrar por grupo, subgrupo ou descrição;



- RF7: A aplicação deve fornecer uma tela que exibe relatórios gráficos com a divisão das despesas por grupo, subgrupo e comparações entre os totais mensais de despesas ao longo do período desejado pelo usuário.

#### 4.2. REQUISITOS NÃO FUNCIONAIS (RNF)

Os requisitos não funcionais definem características do sistema que não estão diretamente relacionadas às funcionalidades específicas. Eles se concentram em aspectos como desempenho, segurança, usabilidade, confiabilidade e outros atributos de qualidade. Esses requisitos influenciam a forma como o sistema executa suas funções e a experiência do usuário, mas não descrevem as operações exatas que o software deve realizar (COOPERSYSTEM, 2024).

Abaixo estão os requisitos não funcionais que a aplicação precisa atender ao longo do seu desenvolvimento:

- RNF1: A aplicação deve verificar se o usuário está autenticado antes de liberar o acesso às funcionalidades de gerenciamento de despesas;
- RNF2: As senhas dos usuários devem ser armazenadas com criptografia para garantir a segurança dos dados pessoais;
- RNF3: A aplicação deve validar as entradas de dados ao lançar uma nova despesa, exibindo um aviso caso algum campo obrigatório esteja vazio ou mal preenchido;
- RNF4: A aplicação deve tratar erros adequadamente, exibindo mensagens claras e amigáveis em caso de falhas nas requisições ou operações realizadas pelos usuários;
- RNF5: Quando o usuário gerar um relatório de despesas, o sistema deve processar os dados informados e exibir as informações de maneira clara e organizada na tela;
- RNF6: Os campos de inserção de valores e descrições de despesas devem ser destacados quando estiverem em foco, para melhorar a usabilidade e compreensão do usuário;
- RNF7: O usuário deve ter a opção de enviar seus relatórios de despesas por e-mail, garantindo que ele tenha um histórico acessível de seus registros

financeiros;

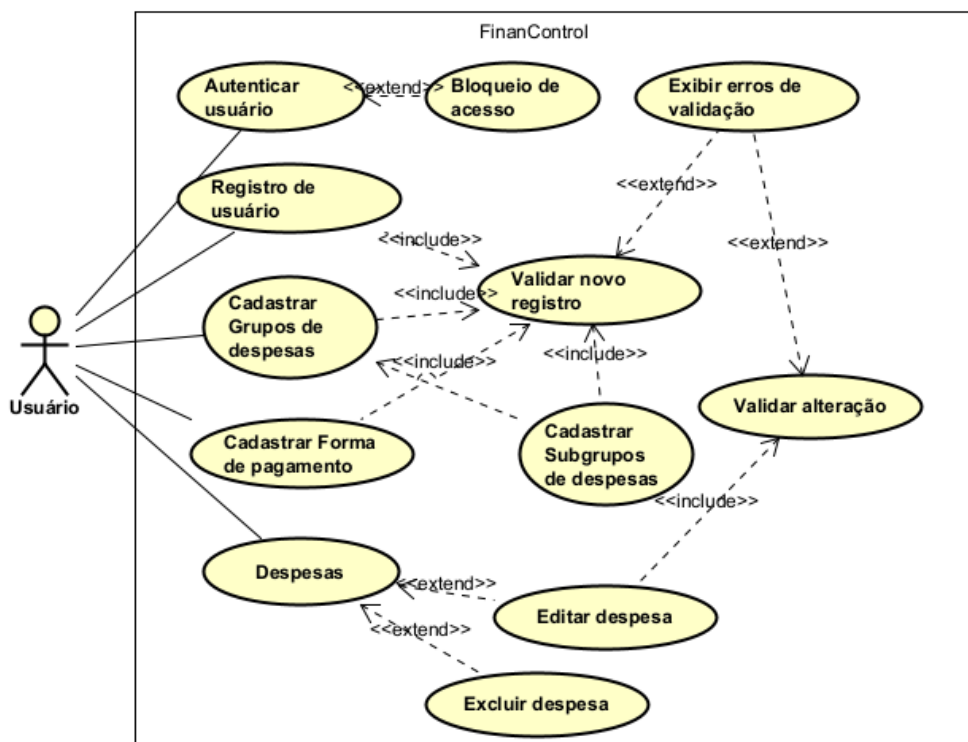
- RNF8: Todas as interfaces da plataforma precisam ser adaptáveis, ajustando o layout automaticamente de acordo com o tamanho da tela do dispositivo, seja ele um desktop, tablet ou celular.

#### 4.3. DIAGRAMA DE CASO DE USO

Os diagramas de caso de uso facilitam a visualização das interações que um usuário ou cliente pode ter com um sistema. Inicialmente aplicados apenas na programação de computadores, esses diagramas ganharam popularidade nos setores de varejo e atendimento ao cliente, ajudando a ilustrar como os clientes se relacionam com uma empresa (GASKIN, 2024).

O caso de uso do FinanControl atende conforme a Figura 1.

Figura 1 - Diagrama de Caso de Uso



Fonte: Próprio autor, 2024.

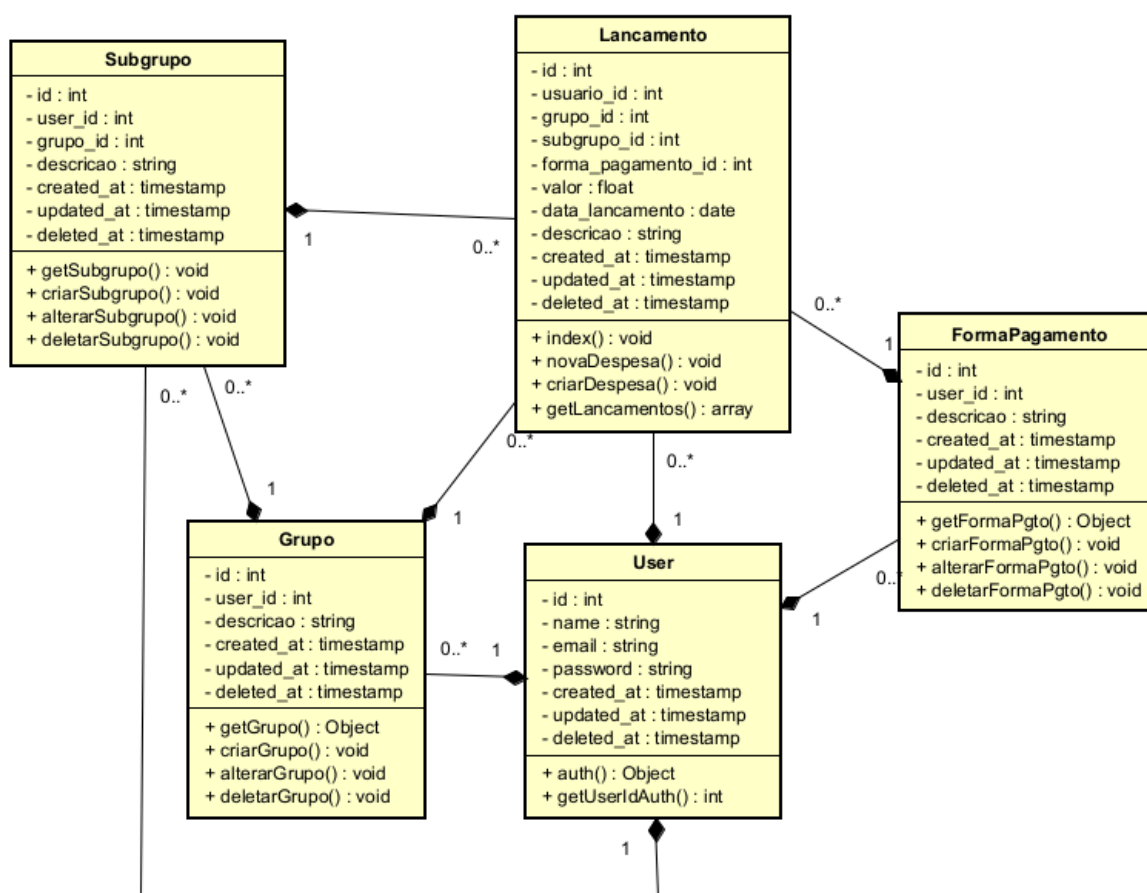
A ilustração anterior mostra os componentes principais de um diagrama de caso de uso, que incluem: Atores, que representam os usuários ou entidades externas que interagem com o sistema; e Associações, que são classificadas como:

Inclusão, que indica uma execução obrigatória; Extensão, que demonstra uma relação de uso opcional; e Herança, usada quando um ator herda o papel de outro.

#### 4.4. DIAGRAMA DE CLASSES

Um diagrama de classes é uma representação visual das estruturas e relacionamentos entre as classes de um sistema em um projeto de software. Ele mostra atributos e métodos das classes, além das relações entre elas, como associações, herança e dependência. Esse tipo de diagrama ajuda a organizar e entender a arquitetura e a lógica do sistema antes da implementação. A Figura 2 retrata o diagrama de classes da aplicação FinanControl.

Figura 2 - Diagrama de Classes



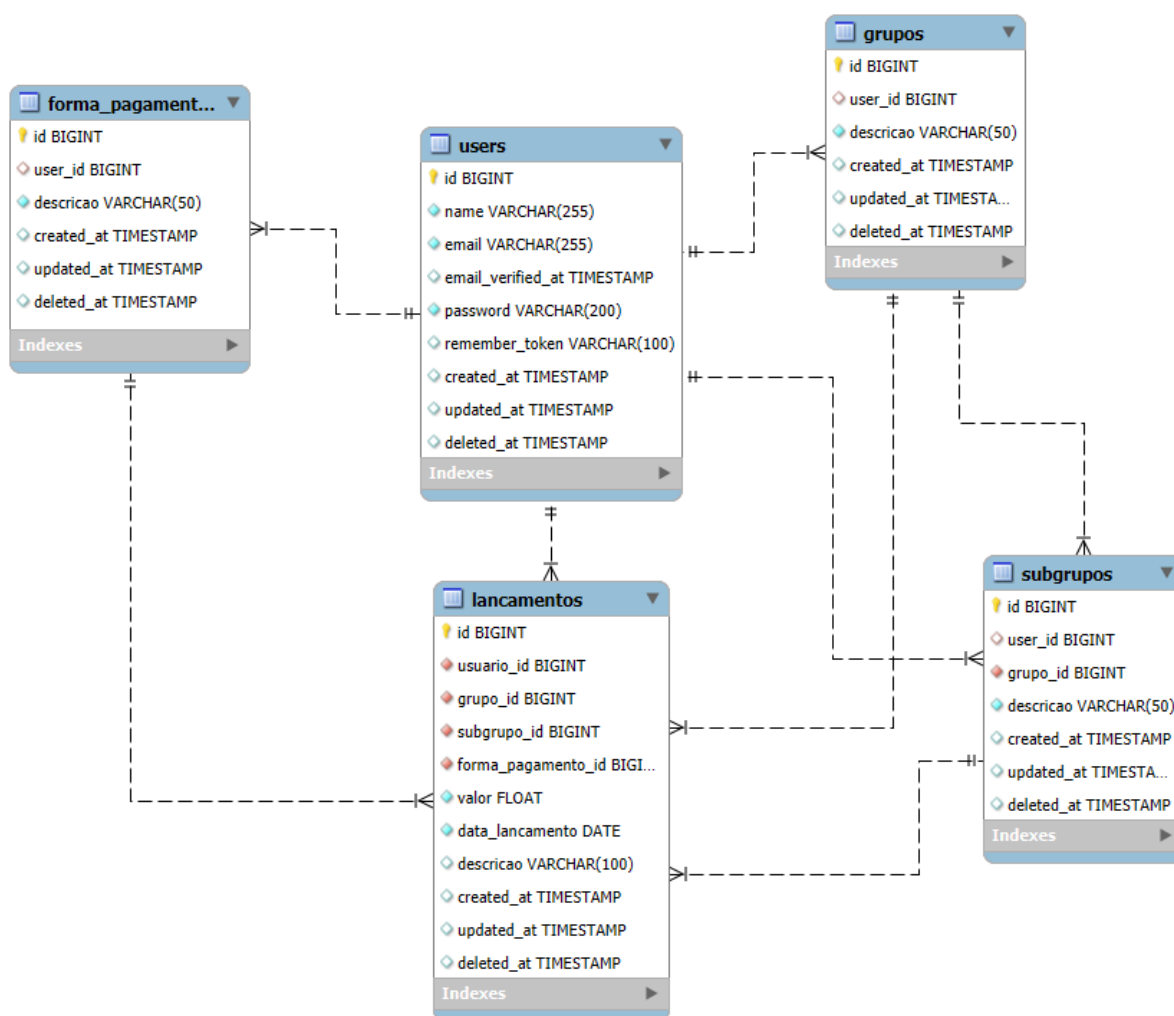
Fonte: Próprio autor, 2024.

#### 4.5. DIAGRAMA DE ENTIDADE E RELACIONAMENTO

Um diagrama de entidade e relacionamento é um fluxograma que mostra como as entidades, como Usuários, Produtos e Postagens, se relacionam em um banco de dados relacional. Ele é usado para modelar e criar bancos de dados, definindo regras lógicas e de negócios no modelo de dados (RAFAELQ80, 2023).

A criação do diagrama de entidade e relacionamento é representada a seguir na Figura 3.

Figura 3 - Diagrama de Entidade e Relacionamento



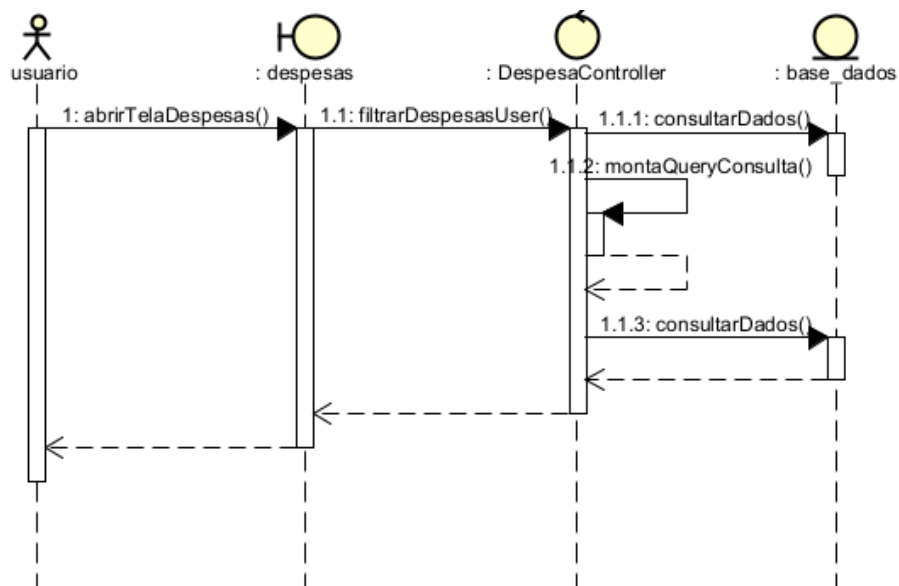
Fonte: Próprio autor, 2024.

#### 4.6. DIAGRAMA DE SEQUÊNCIA DE EVENTOS

Os diagramas de sequência de eventos, amplamente utilizados por desenvolvedores, representam as interações entre objetos dentro de um único caso de uso. Eles mostram como as várias partes de um sistema se conectam e se

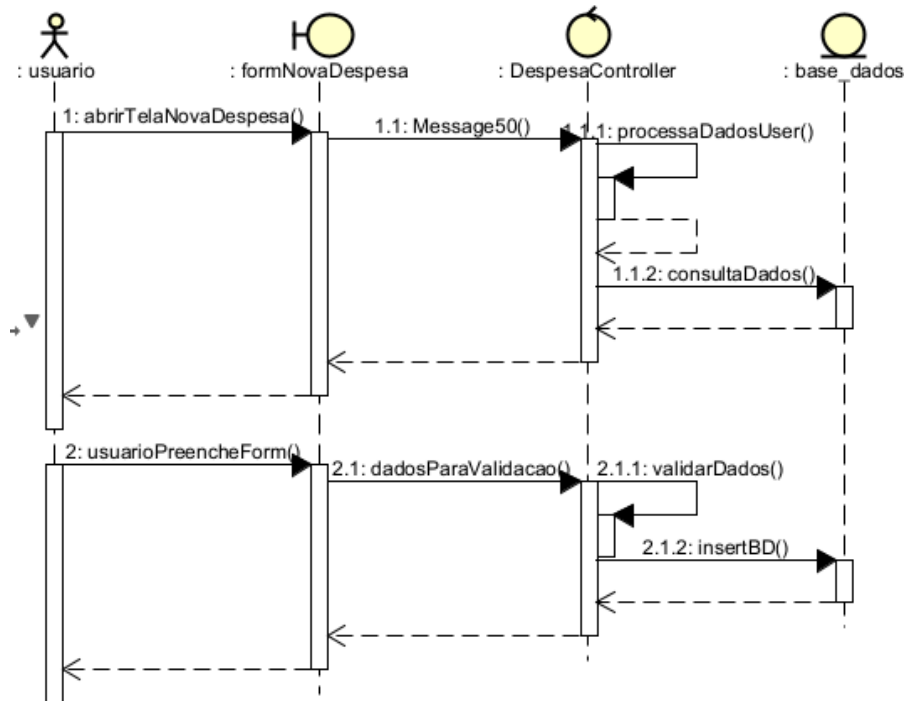
comunicam para realizar uma função específica, além de mostrar a ordem em que essas interações acontecem ao executar o caso de uso. Simplificando, um diagrama de sequência ilustra as partes de um sistema colaborando em uma sequência de passos para alcançar um objetivo (CREATELY, 2021).

Figura 4 - Diagrama de Sequência de Eventos Consultar Despesas



Fonte: Próprio autor, 2024.

Figura 5 - Diagrama de Sequência de Eventos Lançar Nova Despesa



Fonte: Próprio autor, 2024.

## 5. RESULTADOS

### 5.1. SOBRE A PLATAFORMA WEB FINANCONTROL

O FinanControl é uma plataforma online desenvolvida para auxiliar no gerenciamento de despesas pessoais, permitindo que os usuários registrem e acompanhem seus gastos de forma simples e organizada. O FinanControl foi criado com o objetivo de oferecer uma ferramenta prática e intuitiva para quem deseja manter suas despesas pessoais sob controle, sem a necessidade de planilhas complexas ou aplicativos complicados. Construído com a tecnologia Laravel, o FinanControl garante uma experiência de uso estável, rápida e segura. A escolha pelo Laravel como framework de desenvolvimento permite maior flexibilidade e facilita a manutenção e evolução do sistema, assegurando que a plataforma continue funcional e atualizada, atendendo continuamente às necessidades dos usuários.

A plataforma possui dois tipos de usuários: Administrador, que define e organiza as categorias de despesas e cuida da estrutura dos relatórios e funcionalidades; e Usuário Convidado, que registra suas despesas em categorias específicas e visualiza gráficos detalhados sobre seus gastos. Essa estrutura permite que o FinanControl seja uma ferramenta acessível e eficiente para acompanhar o dia a dia financeiro. Entre as funcionalidades principais do FinanControl estão os relatórios gráficos interativos, que permitem ao usuário visualizar suas despesas de forma clara e detalhada. Com esses recursos visuais, o usuário pode identificar padrões de gastos, analisar suas principais despesas e ajustar suas finanças pessoais para alinhar com suas metas.

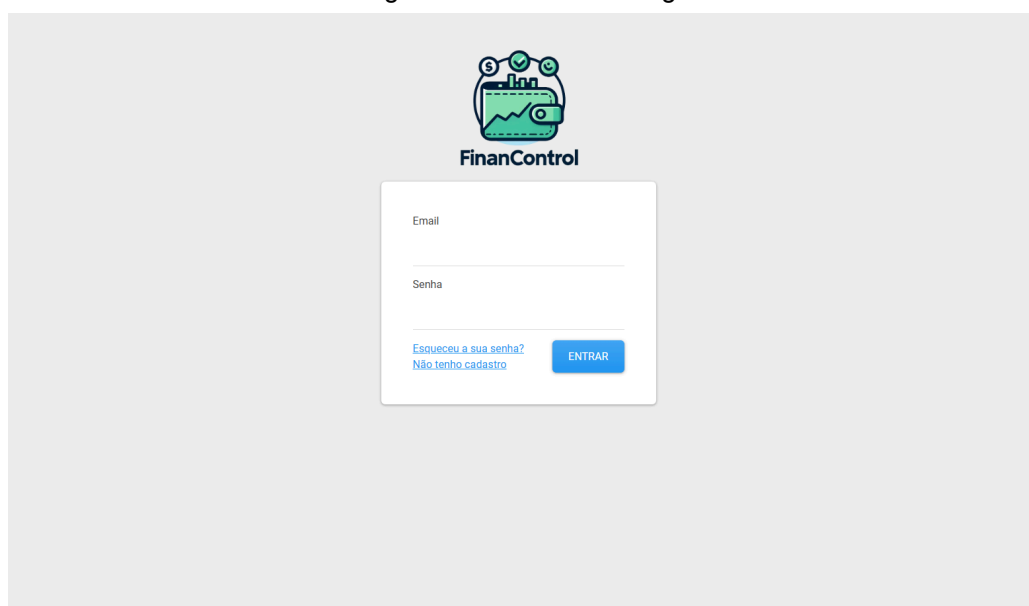
Por ser uma aplicação web, o FinanControl oferece acesso fácil de qualquer dispositivo com navegador, dispensando a necessidade de instalação. A plataforma foi projetada para ser responsiva e intuitiva, garantindo que o usuário consiga registrar e consultar suas despesas de maneira prática e eficiente, onde quer que esteja.

### 5.2. INTERFACES DE LOGIN, REGISTRO E RECUPERAÇÃO DE SENHA

Ao tentar acessar a plataforma FinanControl através de um navegador, o

usuário será automaticamente redirecionado para a tela de login. Essa tela foi desenvolvida utilizando o Fortify, garantindo segurança no processo de autenticação. Caso o usuário precise recuperar a senha, ele poderá acessar a tela de recuperação, onde o sistema, também integrado ao Fortify, enviará um e-mail de recuperação para o endereço cadastrado. Além disso, os novos usuários da plataforma têm a opção de criar uma conta, permitindo o cadastro e acesso à aplicação.

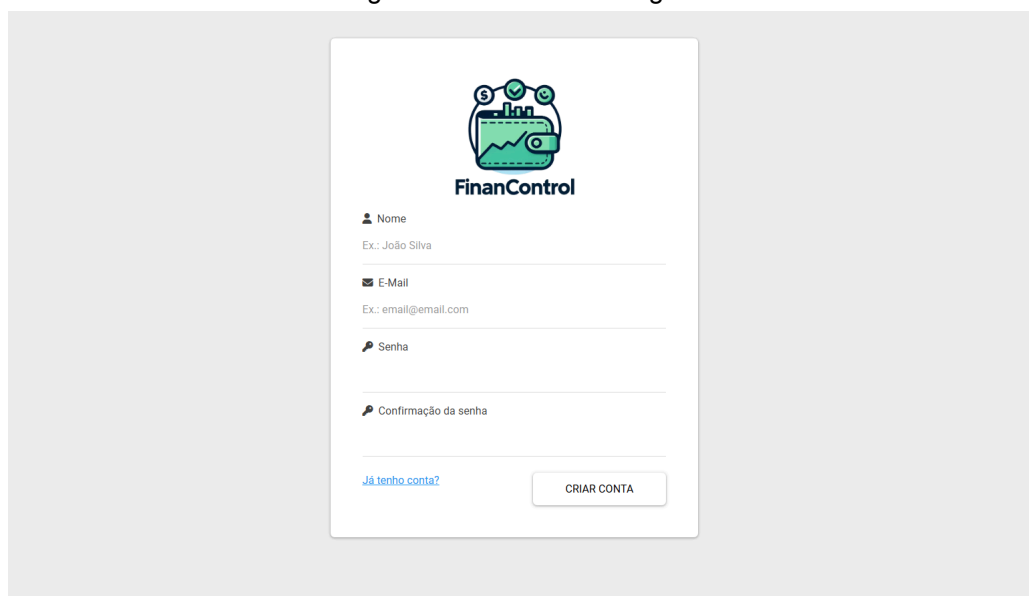
Figura 6 - Interface de Login



A interface de login do FinanControl apresenta o logo da empresa no topo, seguido por campos de entrada para 'Email' e 'Senha'. Abaixo dos campos, há links para 'Esqueceu a sua senha?' e 'Não tenho cadastro', e um botão azul 'ENTRAR'.

Fonte: Próprio autor, 2024.

Figura 7 - Interface de Registro



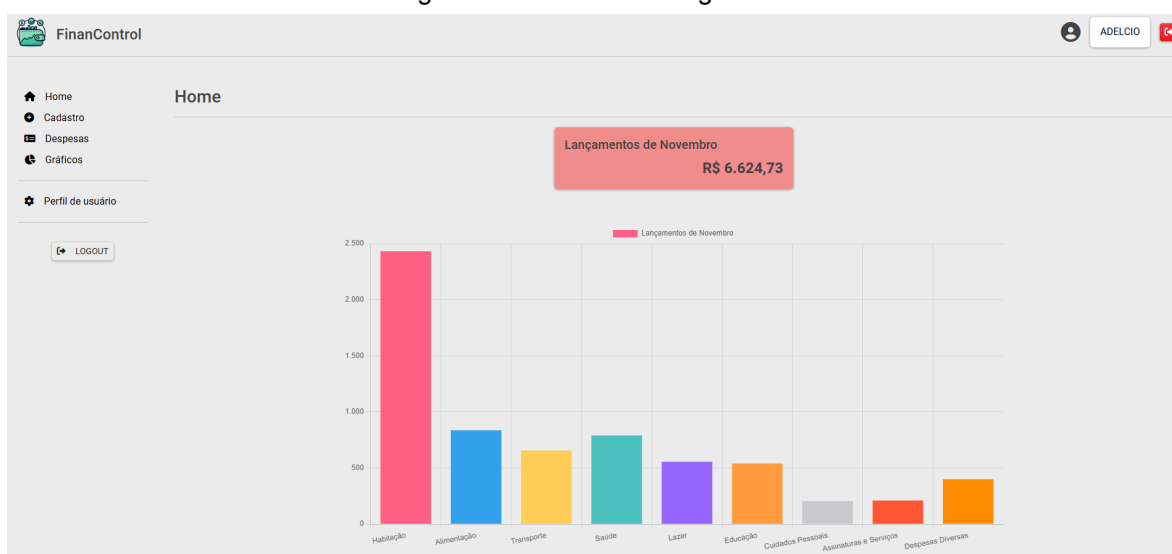
A interface de registro do FinanControl apresenta o logo da empresa no topo, seguido por campos de entrada para 'Nome', 'E-Mail' e 'Senha'. Abaixo dos campos, há um campo para 'Confirmação da senha' e um botão 'CRIAR CONTA'. Há também um link 'Já tenho conta?' para redirecionar o usuário para a tela de login.

Fonte: Próprio autor, 2024.

### 5.3. INTERFACE DA PÁGINA INICIAL

Após a autenticação bem-sucedida, o usuário será redirecionado para a página inicial, onde encontrará um painel simples com informações, como o total de despesas do mês atual e um gráfico que exibe as despesas classificadas por grupos.

Figura 8 - Interface da Página Inicial



Fonte: Próprio autor, 2024.

#### 5.3.1. Barra de Navegação da Página Inicial

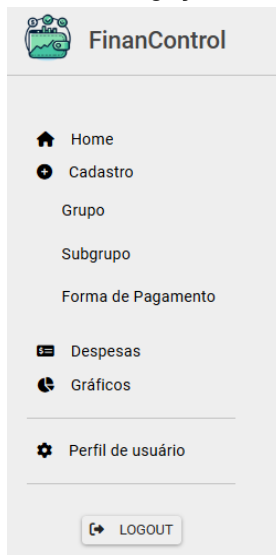
Na parte superior esquerda da plataforma, encontra-se uma barra de navegação que oferece diversas opções para facilitar a experiência do usuário. Ao clicar na opção "Home", o usuário é redirecionado para a tela inicial do sistema. A opção "Cadastro" utiliza um recurso do Bootstrap chamado dropdown, que, ao ser clicado, revela três opções adicionais: "Grupo", que possibilita ao usuário cadastrar, editar ou remover grupos; "Subgrupo", que permite ao usuário cadastrar, editar ou remover subgrupos; e "Forma de Pagamento", onde é possível realizar o cadastro e gestão de diferentes formas de pagamento.

Além disso, a barra de navegação inclui outras opções essenciais, como "Despesas", que exibe uma lista de todas as despesas registradas pelo usuário; "Gráficos", que apresenta uma visualização gráfica das despesas; e "Perfil de Usuário", que permite ao usuário acessar e editar suas informações pessoais.



Dessa forma, as opções da barra de navegação organizam a aplicação de maneira clara e intuitiva, proporcionando uma experiência de navegação fluida e eficiente.

Figura 9 - Barra de navegação da Página Inicial

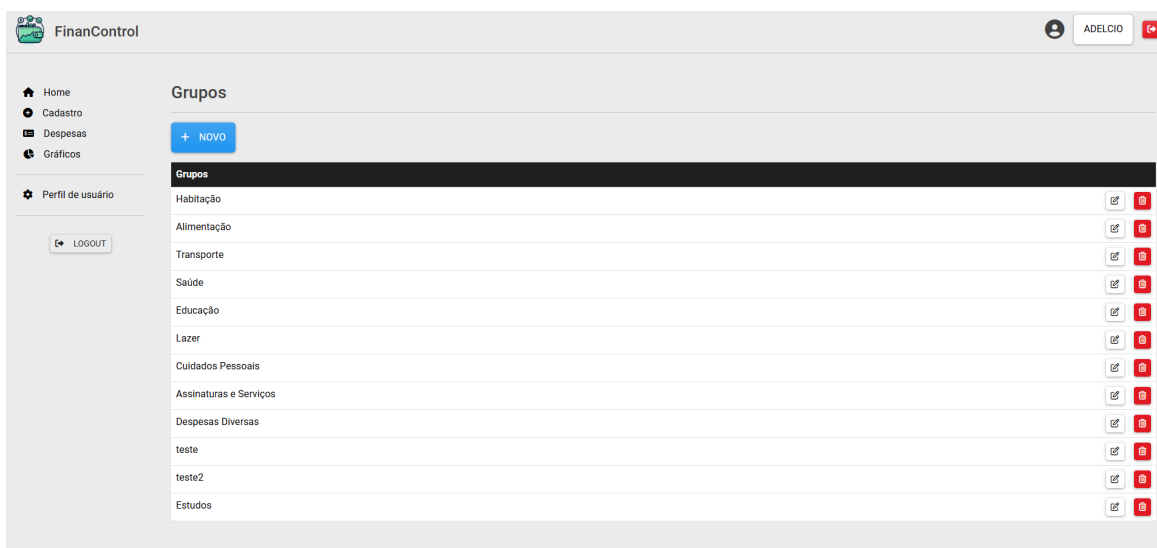


Fonte: Próprio autor, 2024

#### 5.4. INTERFACE DE GRUPOS

Tela onde são listados os grupos cadastrados pelo usuário autenticado, dando opção de editá-los, excluí-los ou realizar o cadastro de um novo grupo.

Figura 10 - Interface de Grupos

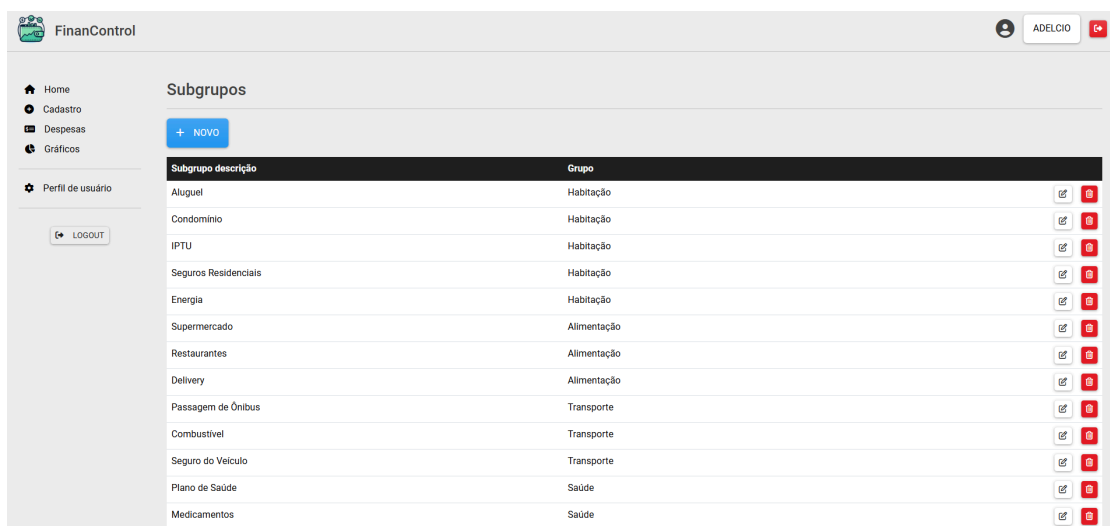


Fonte: Próprio autor, 2024.

## 5.5. INTERFACE DE SUBGRUPOS

Tela onde são listados os subgrupos cadastrados pelo usuário autenticado, dando opção de editá-los, excluí-los ou realizar o cadastro de um novo subgrupo.

Figura 11 - Interface de Subgrupos



Subgrupo descrição	Grupo
Aluguel	Habitação
Condomínio	Habitação
IPTU	Habitação
Seguros Residenciais	Habitação
Energia	Habitação
Supermercado	Alimentação
Restaurantes	Alimentação
Delivery	Alimentação
Passagem de Ônibus	Transporte
Combustível	Transporte
Seguro do Veículo	Transporte
Plano de Saúde	Saúde
Medicamentos	Saúde

Fonte: Próprio autor, 2024.

## 5.6. INTERFACE DE FORMAS DE PAGAMENTO

Tela onde são listadas as formas de pagamento cadastradas pelo usuário autenticado, dando opção de editá-las, excluí-las ou realizar o cadastro de uma nova forma de pagamento.

Figura 12 - Interface de Formas de Pagamento



Formas de Pagamento
CARTAO DE CREDITO
CARTAO DE DEBITO
DINHEIRO
PIX

Fonte: Próprio autor, 2024.

## 5.7. INTERFACE DE DESPESAS

A Tela de Despesas permite ao usuário visualizar todas as despesas registradas, com filtros avançados aplicados em tempo real por meio de JavaScript. É possível realizar filtragens por período, grupo, subgrupo e até mesmo pela descrição do lançamento, gerando um totalizador com base nos critérios selecionados. Esses filtros aprimoram a navegação, proporcionando um controle detalhado e personalizado dos gastos, conforme ilustrado na Figura 13.

Figura 13 - Interface de Despesas

The screenshot shows a web interface titled 'Despesas'. At the top left is a red button labeled '\$ NOVA DESPESA'. Below it is a search bar 'Pesquisar por descrição...' and two dropdown menus for 'Grupo (Todos)' and 'Subgrupo (Todos)'. To the right are two date pickers labeled 'dd/mm/aaaa'. The main part of the interface is a table with the following columns: 'Descrição', 'Grupo', 'Tipo (Subgrupo)', 'Data', and 'Valor R\$'. The table contains 10 rows of expense data. At the bottom right of the table, a 'Total:' row shows 'R\$ 74.467,60'. Below the table is a pagination bar showing 'Showing 1 to 10 of 51 results' and a set of numbered links (1, 2, 3, 4, 5, 6).

Descrição	Grupo	Tipo (Subgrupo)	Data	Valor R\$
Serviços de consultoria	Transporte	Seguros Residenciais	30/11/2024	R\$ 150,00
Serviços de design	Alimentação	Viagens	29/11/2024	R\$ 110,00
Compra de equipamentos	Habitação	Delivery	28/11/2024	R\$ 120,00
Serviços de marketing	Despesas Diversas	Plano de Saúde	27/11/2024	R\$ 90,00
Despesas com viagens	Assinaturas e Serviços	Presentes	26/11/2024	R\$ 60,00
Compra de alimentos	Cuidados Pessoais	Combustível	25/11/2024	R\$ 50,00
Serviços de consultoria	Lazer	Consultas Médicas	24/11/2024	R\$ 300,00
Compra de materiais	Educação	Roupas e Calçados	23/11/2024	R\$ 220,00
Pagamento de taxas	Saúde	Assinatura de Apps	22/11/2024	R\$ 250,00
Despesas com publicidade	Transporte	Cinema e Shows	21/11/2024	R\$ 75,00
Total:				R\$ 74.467,60

Fonte: Próprio autor, 2024.

## 5.8. INTERFACE DE LANÇAMENTO DE DESPESA

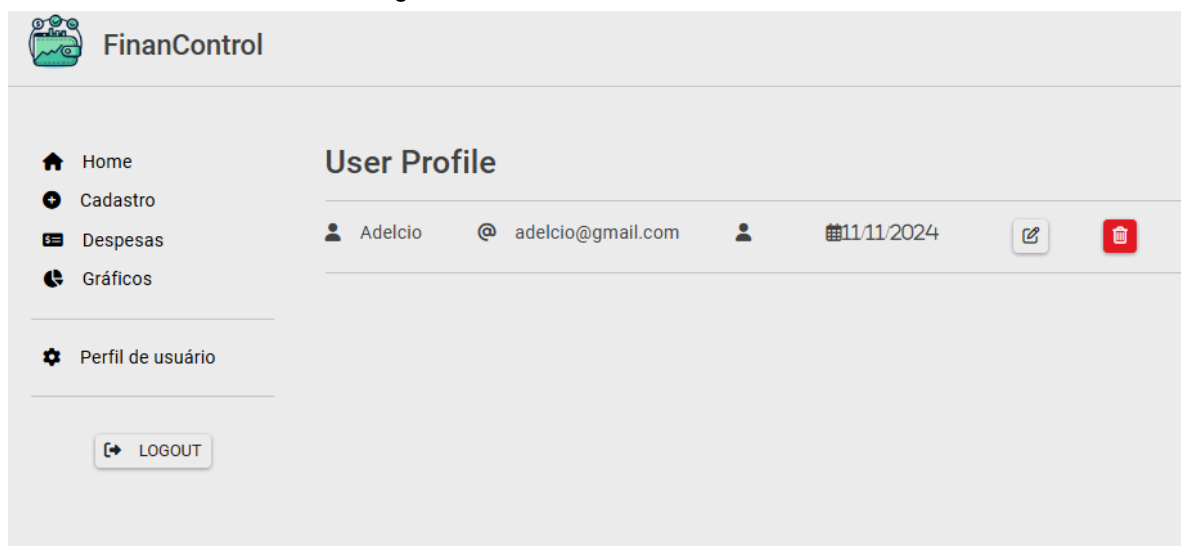
Ao clicar no botão “NOVA DESPESA” na tela de Despesas, conforme ilustrado na Figura 13, o usuário é direcionado para a tela de Lançamento de Despesa. Nessa tela, é necessário preencher um formulário com as seguintes informações: data do lançamento, tipo de lançamento (subgrupo), descrição detalhada da despesa, valor e forma de pagamento. O formulário realiza a validação dos dados utilizando o Eloquent ORM, que processa cada campo para garantir que as informações sejam enviadas corretamente, como mostrado na Figura 14.



## 5.10. INTERFACE DE PERFIL DE USUÁRIO

Tela onde são exibidas as informações do usuário, dando-lhe opção de editá-las, ou até mesmo excluir o usuário.

Figura 16 - Interface de Perfil de Usuário



Fonte: Próprio autor, 2024.

## CONSIDERAÇÕES FINAIS

Desde sua origem, o controle financeiro pessoal tem sido uma necessidade crescente, sendo fundamental para garantir a saúde financeira dos indivíduos. No contexto digital, a utilização de plataformas que facilitam a gestão de despesas tem se mostrado uma ferramenta poderosa para alcançar esse objetivo. O FinanControl, desenvolvido como parte deste Trabalho de Conclusão de Curso, surge como uma solução prática e eficiente para o controle de despesas pessoais. A plataforma tem como propósito possibilitar que os usuários lancem suas despesas de forma simples, as categorize de acordo com suas necessidades e, por fim, produzam relatórios que facilitam a visualização e o acompanhamento dos gastos.

A fase de desenvolvimento da plataforma envolveu diversas etapas que foram essenciais para garantir que os atributos de qualidade de software fossem atendidos, proporcionando uma experiência de uso satisfatória e eficiente. A escolha das tecnologias utilizadas e o planejamento meticuloso durante o desenvolvimento tiveram como propósito otimizar a produção, reduzir custos e garantir a segurança dos dados.

A plataforma FinanControl, ao permitir o lançamento de despesas e a geração de relatórios de forma intuitiva, oferece aos usuários uma forma prática e automatizada de controlar seus gastos, contribuindo para um gerenciamento financeiro mais organizado e consciente. Além disso, o sistema foi desenvolvido com foco na simplicidade e eficiência, proporcionando uma interface amigável e funcionalidades que atendem diretamente às necessidades de quem busca controlar suas despesas pessoais.

Com a conclusão deste trabalho, foi possível aplicar conceitos técnicos de desenvolvimento de sistemas web, abrangendo desde o planejamento e a implementação até o funcionamento completo da plataforma. A solução desenvolvida está funcional e pronta para ser implantada, podendo ser acessada por qualquer usuário através de um navegador, oferecendo uma plataforma eficiente e segura para o controle de despesas pessoais.

## REFERÊNCIAS

BCRYPT. **bcrypt**, 2023. Disponível em: <<https://www.npmjs.com/package/bcrypt>>. Acesso em: 20 set. 2024.

BOOTSTRAP. **Sobre: Bootstrap v4.1**, [s.d.]. Disponível em: <<https://getbootstrap.com.br/docs/4.1/about/overview/>>. Acesso em: 18 set. 2024.

BOLSANELLO, P. R. **Como Instalar e Usar o MailHog com Laravel para Testes de Email**, 2024. Disponível em: <<https://paulorb.dev/como-instalar-e-usar-o-mailhog-com-laravel-para-testes-de-email/#:~:text=O%20MailHog%20%C3%A9%20uma%20ferramenta%20essencial%20para%20o%20desenvolvimento%20de,envi%C3%A1%20Dios%20para%20destinat%C3%A1rios%20reais.>>. Acesso em: 15 out. 2024.

BORGES, J. **O Que São Frameworks e Quais os Mais Utilizados?**, 2023. Disponível em: <<https://www.dio.me/articles/o-que-sao-frameworks-e-quais-os-mais-utilizados>>. Acesso em: 16 set. 2024.

CHART.JS. **Chart.js documentation**, [s.d.]. Disponível em: <<https://www.chartjs.org/docs/latest/>>. Acesso em: 17 set. 2024.

COMPOSER. **Getting Started**, [s.d.]. Disponível em: <<https://getcomposer.org/doc/00-intro.md>>. Acesso em: 22 set. 2024.

CONVERSE, T.; PARK, J. **PHP a Bíblia**. Rio de Janeiro: Elsevier, 2003.

COOPERSYSTEM. **Requisitos funcionais e não funcionais: o que são e qual é a diferença?**, 2024. Disponível em: <<https://www.coopersystem.com.br/requisitos-funcionais-e-nao-funcionais-o-que-sao-e-qual-e-a-diferenca/>>. Acesso em: 20 out. 2024.

CREATELY. **Tutorial do Diagrama de Sequência**: Guia completo com exemplos, 2020. Disponível em: <<https://create.ly.com/blog/pt/diagrama/tutorial-do-diagrama-de-sequencia/>>. Acesso em: 23 out. 2024.

DRUMOND, C. **O que é Scrum e como começar**, [s.d.]. Disponível em: <<https://www.atlassian.com/br/agile/scrum>>. Acesso em: 22 ago. 2024.

EKER, T. Harv. **Os segredos da mente milionária**. Tradução: Pedro Jorgensen Junior. 1. ed. Rio de Janeiro: Sextante, 2006.

ENGHOLM, H. **Engenharia de Software na Prática**. 1. ed. São Paulo: Novatec Editora, 2010.

FERNANDES, M. C. **Linguagens de servidor: uma abordagem prática com PHP**. São Paulo: Editora Senac São Paulo, 2021.

FLANAGAN, D. **JavaScript: o guia Definitivo**. 6. ed. Porto Alegre: Bookman, 2012.

FLATSCHART, F. **HTML 5: embarque imediato**. 1. ed. [s.l.] Brasport, 2011.

GABARDO, A. C. **Laravel para ninjas**. 1. ed. São Paulo: Novatec Editora, 2017.

GASKIN, J. **Tudo o que você precisa saber sobre diagramas de caso de uso**, 2022. Disponível em: <<https://pt.venngage.com/blog/diagrama-de-caso-de-uso/>>. Acesso em: 22 out. 2024.

GITHUB. **Sobre o Git: Documentação do GitHub**, [s.d.]. Disponível em: <<https://docs.github.com/pt/get-started/using-git/about-git>>. Acesso em 02 nov. 2024.

HEIDISQL. **What is HeidiSQL?**, [s.d.]. Disponível em: <<https://www.heidisql.com/>>. Acesso em 9 out. 2024.

HIRANABE, K. **Learn About Diagramming, Modeling Software Company**, [s.d.]. Disponível em: <<https://astah.net/about/>>. Acesso em: 26 set. 2024.

LARAGON. **Documentation**, [s.d.]. Disponível em: <<https://laragon.org/docs/>>. Acesso em: 18 set. 2024.

LARAVEL. **Eloquent: Getting Started**, [s.d.]a. Disponível em: <<https://laravel.com/docs/11.x/eloquent#introduction>>. Acesso em: 27 out. 2024.

LARAVEL. **Laravel Fortify**, [s.d.]b. Disponível em: <<https://laravel.com/docs/11.x/fortify>>. Acesso em: 27 out. 2024.

MILANI, A. **MySQL: Guia do Programador**. 1. ed. São Paulo: Novatec Editora, 2007.

MYSQL. **MySQL Workbench**, 2019. Disponível em: <<https://www.mysql.com/products/workbench/>>. Acesso em: 28 set. 2024.

OTWELL, T. **Laravel: The PHP Framework For Web Artisans**, 2015. Disponível em: <<https://laravel.com/>>. Acesso em: 25 set. 2024.

RAFAELQ80. **Diagrama Entidade Relacionamento (DER) no MySQL Workbench**, 2023. Disponível em: <[https://github.com/conteudoGeneration/cookbook\\_java\\_fullstack/blob/main/03\\_mysql/04.md](https://github.com/conteudoGeneration/cookbook_java_fullstack/blob/main/03_mysql/04.md)>. Acesso em: 23 out. 2024.

SILVA, B. C. C. **Processos e ferramentas para o desenvolvimento de software livre: um estudo de caso**. Dissertação (Mestrado em Informática) - Universidade Federal do Espírito Santo, Vitória, 2006.

SILVA, M. S. **JavaScript - Guia do Programador: Guia Completo das Funcionalidades de Linguagem JavaScript**. 1. ed. São Paulo: Novatec Editora, 2010.

TABLELESS. **Uma breve história do CSS**, 2006. Disponível em: <<https://tableless.com.br/uma-breve-historia-do-css/>>. Acesso em: 25 out. 2024.



TAIGA. **About us**, 2015. Disponível em: <<https://taiga.io/about-us/>>. Acesso em: 18 out. 2024.

VISUAL STUDIO CODE. **Documentation for Visual Studio Code**, 2023. Disponível em: <<https://code.visualstudio.com/docs>>. Acesso em: 21 out. 2024.

## GLOSSÁRIO

**API:** *Application Programming Interface*, conjunto de regras que permite a comunicação e integração entre softwares.

**Back-end:** Parte do desenvolvimento responsável pelo processamento de dados, lógica de negócios e interação com o banco de dados, geralmente no servidor.

**CSS:** Linguagem usada para estilizar a aparência de páginas web, controlando elementos como cores, fontes e layouts.

**Dashboard:** Interface gráfica que exibe informações e dados importantes de forma resumida e visual.

**Framework:** Conjunto de ferramentas e bibliotecas que oferece uma estrutura para facilitar o desenvolvimento de aplicações de forma rápida e organizada.

**Front-end:** Parte do desenvolvimento que lida com a interface do usuário e a interação visual.

**HTML:** *HyperText Markup Language*, linguagem de marcação usada para estruturar conteúdo na web.

**JSON:** *JavaScript Object Notation*, formato leve e legível para armazenar e transmitir dados entre servidores e aplicações web, frequentemente usado em APIs.

**Layout:** Organização visual e disposição dos elementos em uma página web ou aplicativo.

**PHP:** *Hypertext Preprocessor*, linguagem de programação de código aberto usada no desenvolvimento de aplicações web dinâmicas e interativas.

**Plugins:** Extensões de software que adicionam funcionalidades a um sistema ou aplicativo sem alterar sua estrutura principal.

**Script:** Conjunto de instruções que executa comandos em uma aplicação.

**SMTP:** *Simple Mail Transfer Protocol*, protocolo usado para enviar e-mails entre servidores na internet.