**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**PUC Minas Virtual**

**Pós-graduação *Lato Sensu* em Engenharia de *Software***

Projeto Integrado

Relatório Técnico

FINANCE

Thiago Beck Ribeiro Guimarães

Belo Horizonte

Maio/2024

# Projeto Integrado

**Sumário**

[Projeto Integrado 3](#_Toc167395074)

[1. Cronograma de Trabalho 4](#_Toc167395075)

[2. Introdução 5](#_Toc167395076)

[O problema 5](#_Toc167395077)

[Objetivo 5](#_Toc167395078)

[Justificativa e motivações 5](#_Toc167395079)

[3. Definição Conceitual da Solução 6](#_Toc167395080)

[Lista de Requisitos Funcionais 6](#_Toc167395081)

[Lista de requisitos não funcionais 7](#_Toc167395082)

[Regras de Negócio 7](#_Toc167395083)

[4. Protótipo Navegável do Sistema 8](#_Toc167395084)

[Tela de entrada 8](#_Toc167395085)

[Tela principal 9](#_Toc167395086)

[Tela de cadastro de categoria 9](#_Toc167395087)

[Tela de cadastro de movimentação de entrada/saida 10](#_Toc167395088)

[Tela de opções 10](#_Toc167395089)

[5. Arquitetura da Solução 11](#_Toc167395090)

[Diagrama de Casos de Uso 11](#_Toc167395091)

[Diagrama de Classes 12](#_Toc167395092)

[Modelagem Objeto Relacional 15](#_Toc167395093)

[6. Plano de Testes 16](#_Toc167395094)

[7. Relatório de Execução de Testes de Software 17](#_Toc167395095)

[8. Código da Aplicação 19](#_Toc167395096)

[9. Referências 20](#_Toc167395098)

## Cronograma de Trabalho

|  |  |  |  |
| --- | --- | --- | --- |
| **Datas** | | **Atividade / Tarefa** | **Produto / Resultado** |
| **De** | **Até** |
| 01/05/2024 | 03/05/2024 | 1. Busca de ideias e documentação | Entendimento da elaboção do projeto |
| 03/05/2024 | 03/05/2024 | 1. Mapeamento da documentação. | Planejamento do desenvolvimento |
| 03/05/2024 | 05/05/2024 | 1. Elaboração dos casos de uso | Diagrama de Casos de uso |
| 05/05/2024 | 07/05/2024 | 1. Revisão geral da documentação. | Diagrama de Classes e Entidade-Relacionamento |
| 03/05/2024 | 12/05/2024 | 1. Criação do Backend | Implementação em Java. |
| 07/05/2024 | 12/05/2024 | 1. Criação do Frontend | Implementação em React. |
| 10/05/2024 | 12/05/2024 | 1. Integração de Backend com Frontend | Conexões via Rest API |
| 11/05/2024 | 14/05/2024 | 1. Ajuste da documentação após desenvolvimento. | Correções no diagrama de Entidade-Relacionamento |
| 14/05/2024 | 17/05/2024 | 1. Ajustes de código e correções de bug. | Finalização da parte Web. |
| 17/05/2024 | 22/05/2024 | 1. Elaboração do relatório técnico | Geração de toda a documentação do Projeto em formato de Relatório. |
| 23/05/2024 | 23/05/2024 | 1. Adição de todo o projeto no GitHub para acesso do professor. | Publicação do Projeto. |

## Introdução

Segundo (CONFEDERAÇÃO NACIONAL DO COMÉRCIO DE BENS, 2020) o percentual de famílias com dívidas alcançou 66,5% em maio de 2020 no Brasil conforme a Pesquisa de Endividamento e Inadimplência do Consumidor (PEIC), divulgada em 20 de maio de 2020.

Um dos motivos para o endividamento seria a falta de organização e gestão dos gastos, além da falta de conhecimento sobre como controlar suas finanças e melhores maneiras para que consigam gastar e economizar dinheiro da maneira mais viável, gerando um grande número de pessoas endividadas.

O objetivo desse trabalho tem como função primordial ajudar a população em massa a organizar suas finanças de maneira consciente e rápida, através da criação de um aplicativo para celular com o intuito de organizar os gastos diários de uma pessoa.

### O problema

O planejamento financeiro pessoal é algo que todos deveriam fazer para ter o melhor controle de suas finanças, tendo em vista que devemos saber exatamente com o que estamos gastando e registrar futuros gastos para que não ocorram imprevistos.

Contudo, o cotidiano moderno é muito corrido e a maioria das pessoas não tem tempo para realizar este controle em um papel ou planilha, tornando-se mais fácil a utilização de um aplicativo.

### Objetivo

Elaborar um software capaz de armazenar os fluxos financeiros pessoais de cada usuário, mostrando com auxílios de dashboards e gráficos onde está sendo os maiores gastos e em quais categorias (alimentação, lazer, saúde, despesas fixas) eles ocorrem.

### Justificativa e motivações

O trabalho tem como justificativa fazer com que o usuário tenha um melhor controle e visão de seus gastos e assim possa administrar melhor seu capital e saber exatamente onde está gastando seu dinheiro.

A consequência da má administração de suas receitas pode levar a pessoa esquecer-se de pagar certas contas e utilizar aquele dinheiro para outros fins, levando futuramente a dívidas.

As motivações são em função da aplicação dos conhecimentos técnicos adquiridos durante o curso de engenharia da computação, e ainda, por uma paixão pessoal sobre finanças e aplicação de capital.

## Definição Conceitual da Solução

Este trabalho teve sua origem a partir da vontade de aprender sobre a criação de aplicativos para web, além da aplicação dos conceitos aprendidos durante o curso de Engenharia de Software. Com um gosto pessoal pelo ramo de finanças foi desenvolvido um aplicativo para gerenciamento de gastos diários, para que qualquer pessoa possa ter um melhor controle sobre suas movimentações.

A primeira etapa do desenvolvimento, foram levantados os casos de uso possíveis utilizando como base os aplicativos citados como trabalhos relacionados, Mobills, Minhas Economias. Sendo assim foram desenvolvidos os requisitos funcionais e não funcionais para a aplicação.

Na segunda etapa, os requisitos funcionais e não funcionais foram validados e melhorados para então ser criado as regras de negócio do software. Nesta etapa foi reduzido o número inicial de casos de uso para ter um escopo mais limitado sendo possível a implementação no tempo viável.

Na terceira etapa todas os requisitos foram implementados no back-end e testados conforme o desenvolvimento de cada funcionalidade e logo após foram desenvolvidos as telas e recursos visuais que seriam disponibilizados para o usuário.

### Lista de Requisitos Funcionais

Na Tabela 1 é apresentado os requisitos funcionais para a modelagem do sistema, referente as funções mínimas que devem ser atendidas pelo software.

Tabela - Lista de requisitos funcionais

|  |  |
| --- | --- |
| **Requisitos Funcionais** | **Caso de Uso** |
| RF01 - O sistema deve permitir o seu uso mediante usuário e senha (login). | UC01 |
| RF02 - O sistema deve permitir o cadastro de um novo usuário. | UC02 |
| RF03 - O sistema deve permitir a um usuário logado visualizar seu saldo atual. | UC03 |
| RF04 - O sistema deve permitir a um usuário logado visualizar seus gastos do mês atual. | UC04 |
| RF05 - O sistema deve permitir a um usuário logado visualizar suas receitas do mês atual. | UC05 |
| RF06 - O sistema deve permitir a um usuário logado visualizar seus gastos do mês por categoria. | UC06 |
| RF07 - O sistema deve permitir a um usuário logado visualizar suas transações. | UC07 |
| RF08 - O sistema deve permitir a um usuário logado deletar uma transação. | UC08 |
| RF09 - O sistema deve permitir a um usuário logado visualizar uma transação específica e seus detalhes. | UC09 |
| RF10 - O sistema deve permitir a um usuário logado visualizar suas transações no mês de sua escolha. | UC10 |
| RF11 - O sistema deve permitir a um usuário logado adicionar uma transação de entrada. | UC11 |
| RF12 - O sistema deve permitir a um usuário logado adicionar uma transação de saída. | UC12 |
| RF13 - O sistema deve permitir a um usuário logado adicionar novas categorias para transações de saída. | UC13 |
| RF14 - O sistema deve permitir a um usuário logado deslogar do sistema. | UC14 |

Fonte: Autor, 2024

### Lista de requisitos não funcionais

Na Tabela 2 é apresentando os requisitos não funcionais que irão demostrar como o sistema deve desempenhar suas funções em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenibilidade e tecnologias envolvidas.

Tabela - Lista de requisitos não funcionais

|  |
| --- |
| **Requisitos não funcionais** |
| RNF01 - O sistema deverá ter controle de acesso. |
| RNF02 - O sistema terá seu acesso liberado através do aplicativo; |
| RNF03 - O sistema deverá executar em plataformas Android e IOS. |
| RNF04 - O sistema deverá possuir uma interface intuitiva. |
| RNF05 - O sistema deverá utilizar o padrão de comunicação http rest. |
| RNF06 - O sistema deverá utilizar o JSON como padrão para a comunicação http. |
| RNF07 - O sistema deverá se comunicar com o banco de dados PostgreSQL. |
| RNF08 - O sistema deverá ter seu back-end desenvolvido na linguagem Java. |
| RNF09 - O sistema deverá ter seu front-end desenvolvido na linguagem JavaScript. |
| RNF10 - O sistema deverá em seu front-end utilizar da biblioteca react native. |

Fonte: Autor, 2024

### Regras de Negócio

Na tabela 3 é possível observar as regras de negócio que definem como ele funciona. Essas abrangem uma gama de assuntos, como suas políticas, objetivos, compromissos éticos e sociais, decisões estratégicas, leis e regulamentações.

Tabela - Lista de regras de negócio

|  |
| --- |
| **Regras de negócio** |
| RN01 - Para registrar um novo usuário é necessário um e-mail válido. |
| RN02 - Para registrar um novo usuário é necessária uma senha com mínimo de 8 caracteres. |
| RN03 - Para fazer login é necessário ser um usuário cadastrado. |
| RN04 - Após o login o token de validação do usuário será valido por 20 dias. |
| RN05 - Nenhum usuário poderá ser excluído. |

Fonte: Autor, 2024

## Protótipo Navegável do Sistema

Neste capítulo serão apresentadas as telas do aplicativo proposto neste trabalho e um breve comentário sobre suas respectivas funções.

### Tela de entrada

A Figura representa a tela de entrada onde é possível executar 2 funções: entrar no aplicativo com seus dados de e-mail e senha - caso o usuário já tenha cadastro -, ser redirecionado para a tela de registro. Validações de e-mail e usuário cadastrado também estão presentes - caso o e-mail for inválido ou não cadastrado uma mensagem será exibida para o usuário.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Tela de acesso ao App. Fonte: Autor, 2024

### Tela principal

A Figura é observado a tela principal do sistema, nesta tela o usuário poderá observar de forma generalizada o saldo atual de suas movimentações cadastradas no sistema, o saldo para o mês corrente, assim como suas entradas e saídas.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Tela principal. Fonte: Autor, 2024

### Tela de cadastro de categoria

Na Figura é apresentado a tela de cadastro de categorias, onde o usuário poderá cadastrar novas nomeações para agrupar suas movimentações de saída. Essas categorias serão utilizadas pela tela principal exibir o gráfico de gastos por categoria e também será exibido no histórico de movimentações.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Tela de cadastro de categoria. Fonte: Autor, 2024

### Tela de cadastro de movimentação de entrada/saída

Na Figura é apresentada a tela de cadastro de movimentações de entrada e saída. Nesta tela o usuário irá cadastrar suas movimentações, tal como o recebimento de algum valor, onde é possível escolher a data que ele foi creditado assim como adicionar uma descrição para essa movimentação e categoria.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Tela de cadastro de movimentação de entrada e saída. Fonte: Autor, 2024.

### Tela de opções

Na Figura é apresentado uma tela de opções, onde o usuário possui a opção de encerrar sua sessão no aplicativo e retornar para a tela de entrada.

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

Tela de opções. Fonte: Autor, 2024.

## Arquitetura da Solução

### Diagrama de Casos de Uso

Na Figura é apresentado o Diagrama de Casos de Uso que exibe as funcionalidades que serão implementadas no sistema, as mesmas se dão origem através do trabalho de levantamento de requisitos.

Diagrama

Descrição gerada automaticamente

Diagrama de casos de uso. Fonte: Autor, 2024

### Diagrama de Classes

Linha do tempo

Descrição gerada automaticamenteNos diagramas são apresentadas as classes de negócio que foram implementadas seguindo o trabalho de levantamento de requisitos. É possível visualizar as classes, seus atributos métodos e dependências. Na Figura é possível observar o diagrama de classe para as Transações.

Uma imagem contendo Diagrama

Descrição gerada automaticamenteNa Figura é possível observar o diagrama de classes para as Categorias de Transação

Na Figura é possível observar o digrama de classes para as classes de Usuário.

Tabela

Descrição gerada automaticamente com confiança baixa

### Modelagem Objeto Relacional

Diagrama

Descrição gerada automaticamente

## Plano de Testes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nº** | **Caso de uso** | **Objetivo do caso de teste** | **Entradas** | **Resultados esperados** |
| 1 | UC01 | Testar a funcionalidade de login do usuário. | Usuário e senha válidos | Acesso concedido ao sistema. |
| 2 | UC01 | Testar a funcionalidade de login com usuário inválido. | Usuário inválido e senha válida | Mensagem de erro indicando que o usuário não existe. |
| 3 | UC01 | Testar a funcionalidade de login com senha inválida. | Usuário válido e senha inválida | Mensagem de erro indicando senha incorreta. |
| 4 | UC02 | Testar o cadastro de um novo usuário. | Nome, e-mail válido, senha válida | Novo usuário cadastrado com sucesso. |
| 5 | UC02 | Testar o cadastro com e-mail já existente. | Nome, e-mail existente, senha válida | Mensagem de erro indicando que o e-mail já está em uso. |
| 6 | UC03 | Verificar saldo atual do usuário logado. | Usuário logado | Exibição correta do saldo atual do usuário. |
| 7 | UC04 | Verificar visualização dos gastos do mês atual. | Usuário logado | Exibição correta dos gastos do mês atual. |
| 8 | UC05 | Verificar visualização das receitas do mês atual. | Usuário logado | Exibição correta das receitas do mês atual. |
| 9 | UC06 | Verificar visualização dos gastos por categoria. | Usuário logado | Exibição correta dos gastos por categoria. |
| 10 | UC07 | Verificar visualização de todas as transações do usuário. | Usuário logado | Exibição correta de todas as transações do usuário. |
| 11 | UC08 | Testar a funcionalidade de deletar uma transação. | Usuário logado, ID da transação | Transação deletada com sucesso. |
| 12 | UC09 | Verificar detalhes de uma transação específica. | Usuário logado, ID da transação | Exibição correta dos detalhes da transação. |
| 13 | UC10 | Verificar visualização das transações em um mês específico. | Usuário logado, mês selecionado | Exibição correta das transações do mês selecionado. |
| 14 | UC11 | Testar a adição de uma transação de entrada. | Usuário logado, detalhes da transação | Transação de entrada adicionada com sucesso. |
| 15 | UC12 | Testar a adição de uma transação de saída. | Usuário logado, detalhes da transação | Transação de saída adicionada com sucesso. |
| 16 | UC13 | Testar a adição de novas categorias para transações de saída. | Usuário logado, nova categoria | Nova categoria adicionada com sucesso. |
| 17 | UC14 | Testar a funcionalidade de logout do sistema. | Usuário logado | Usuário deslogado com sucesso. |

## Relatório de Execução de Testes de Software

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nº** | **Caso de teste** | **Saída esperada** | **Resultados encontrados** | **Aprovado?** |
| 1 | Testar a funcionalidade de login do usuário com credenciais válidas. | Acesso concedido ao sistema. | Acesso concedido ao sistema. | Sim |
| 2 | Testar a funcionalidade de login com usuário inválido. | Mensagem de erro indicando que o usuário não existe. | Mensagem de erro indicando que o usuário não existe. | Sim |
| 3 | Testar a funcionalidade de login com senha inválida. | Mensagem de erro indicando senha incorreta. | Mensagem de erro indicando senha incorreta. | Sim |
| 4 | Testar o cadastro de um novo usuário com dados válidos. | Novo usuário cadastrado com sucesso. | Novo usuário cadastrado com sucesso. | Sim |
| 5 | Testar o cadastro com e-mail já existente. | Mensagem de erro indicando que o e-mail já está em uso. | Mensagem de erro indicando que o e-mail já está em uso. | Sim |
| 6 | Verificar saldo atual do usuário logado. | Exibição correta do saldo atual do usuário. | Exibição correta do saldo atual do usuário. | Sim |
| 7 | Verificar visualização dos gastos do mês atual. | Exibição correta dos gastos do mês atual. | Exibição correta dos gastos do mês atual. | Sim |
| 8 | Verificar visualização das receitas do mês atual. | Exibição correta das receitas do mês atual. | Exibição correta das receitas do mês atual. | Sim |
| 9 | Verificar visualização dos gastos por categoria. | Exibição correta dos gastos por categoria. | Exibição correta dos gastos por categoria. | Sim |
| 10 | Verificar visualização de todas as transações do usuário. | Exibição correta de todas as transações do usuário. | Exibição correta de todas as transações do usuário. | Sim |
| 11 | Testar a funcionalidade de deletar uma transação. | Transação deletada com sucesso. | Transação deletada com sucesso. | Sim |
| 12 | Verificar detalhes de uma transação específica. | Exibição correta dos detalhes da transação. | Exibição correta dos detalhes da transação. | Sim |
| 13 | Verificar visualização das transações em um mês específico. | Exibição correta das transações do mês selecionado. | Exibição correta das transações do mês selecionado. | Sim |
| 14 | Testar a adição de uma transação de entrada. | Transação de entrada adicionada com sucesso. | Transação de entrada adicionada com sucesso. | Sim |
| 15 | Testar a adição de uma transação de saída. | Transação de saída adicionada com sucesso. | Transação de saída adicionada com sucesso. | Sim |
| 16 | Testar a adição de novas categorias para transações de saída. | Nova categoria adicionada com sucesso. | Nova categoria adicionada com sucesso. | Sim |
| 17 | Testar a funcionalidade de logout do sistema. | Usuário deslogado com sucesso. | Usuário deslogado com sucesso. | Sim |

## Código da Aplicação

## O código da aplicação desenvolvida foi disponibilizado no GitHub no seguinte repositório: https://github.com/thiagobrg/tcc-pos. Este repositório contém todos os arquivos necessários para a execução do projeto, incluindo a parte de backend e frontend.

## Referências

ALVES, M. B. M. **Bolso Virtual:** **Aplicação Web Móvel para controle de finanças pessoais.** São Cristóvão - Sergipe: Universidade Federal de Sergipe, 2017.

AUSPEX. **Minhas Economias.** Minhas Economias, 2020. Disponivel em: <https://wwws.minhaseconomias.com.br>.

CONFEDERAÇÃO NACIONAL DO COMÉRCIO DE BENS, S. E. T. **CNC: endividamento das famílias alcança 66,5% em maio.** CNC: endividamento das famílias alcança 66,5% em maio, 2020. Disponivel em: <https://agenciabrasil.ebc.com.br/economia/noticia/2020-05/cnc-endividamento-das-familias-alcanca-665-em-maio#:~:text=O%20percentual%20de%20endividamento%20foi,foi%20divulgada%20hoje%20(20).>.

DEVMEDIA. **Java EE: Entendendo a plataforma.** Devmedia, 2020. Disponivel em: <https://www.devmedia.com.br/java-ee-entendendo-a-plataforma/30195>.

FLANAGAN, D. **JavaScript:** The Definitive Guide. [S.l.]: Bookman, v. 6E, 2013.