

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
PUC Minas Virtual
Pós-graduação *Lato Sensu* em Engenharia de *Software*

Projeto Integrado

Relatório Técnico

Granada

Matheus Felipe Araújo Calaça

Belo Horizonte
Agosto de 2022.

Projeto Integrado

Sumário

Projeto Integrado	3
1. Cronograma de Trabalho	4
2. Introdução	6
3. Definição Conceitual da Solução	7
3.1 Diagrama de Casos de Uso	8
3.2 Requisitos Funcionais	9
3.3 Requisitos Não-funcionais	12
4. Protótipo Navegável do Sistema	13
5. Diagrama de Classes de Domínio	15
6. Arquitetura da Solução	15
6.1 Padrão Arquitetural	15
6.2 C4 model - Diagrama de Contexto	16
7. Frameworks de Trabalho	16
8. Estrutura Base do Front-End	17
9. Apropriação de Horas no Projeto	19
10. Código da Aplicação	19
11. Referências	20

1. Cronograma de Trabalho

Datas		Atividade / Tarefa	Produto / Resultado
De	Até		
05 / 12 / 21	10 / 01 / 22	1. Pesquisa e Estudos sobre a ideia de desenvolver um sistema de controle financeiro	Entender se é viável construir um sistema financeiro, para auxiliar o controle financeiro da população em geral
10 / 12 / 21	10 / 01 / 22	2. Estudar qual é a melhor forma de aplicar as propostas de ajuda financeira no sistema.	Achar quais as opções e soluções para ajudar no controle financeiro, que encaixam dentro de um software.
11 / 01 / 22	21 / 01 / 22	3. Criar um diagrama de caso de uso para o sistema	Construir um diagrama de caso de uso do sistema para entender quais funcionalidades serão necessárias
21 / 01 / 22	30 / 01 / 22	4. Levantar os requisitos funcionais para a construção do sistema	Ter os requisitos funcionais do sistema descrito e as dúvidas negociais resolvidas
25 / 01 / 22	05 / 02 / 22	5. Fazer um levantamento dos requisitos não funcionais	Ter todos os requisitos não funcionais escritos e mapeados.
05 / 02 / 22	07 / 02 / 22	6. Revisar tudo que foi feito e escrito com a intenção de encontrar algo que passou despercebido	Caso encontre algo a acrescentar ou retirar atualizar os requisitos escritos e os casos de uso
07 / 02 / 22	10 / 02 / 22	7. Estudo de interfaces de sistemas financeiro	Entender o fluxo navegável de sistemas financeiros ou similares. Para usar como base na construção da interface do sistema.
10 / 02 / 22	20 / 02 / 22	8. Desenhar um protótipo da interface de navegação do cliente	Está com o protótipo de navegação do sistema desenhado.
20 / 02 / 22	20 / 03 / 22	9. Criar um MVP navegável das telas do sistema	Ter um MVP das telas do sistema pensando nos três principais requisito. Sem conexão com o back-end
20 / 03 / 22	25 / 03 / 22	10. Revisar o protótipo de navegação do sistema	Corrigido e encontrado erros que podem ter ficado durante a construção do protótipo navegável do sistema
25 / 03 / 22	05 / 03 / 22	11. Construir o diagrama de classe de domínio	Estar com o digrama de classe de domínio pronto e revisado
05 / 03 / 22	08 / 03 / 22	12. Estudar padrões arquiteturais que melhor se enquadra no projeto levando em consideração todos os estudos já realizados	Padrão arquitetural definido
08 / 03 / 22	15 / 03 / 22	13. Desenhar a arquitetura da solução	Está com a arquitetura da solução desenhada e escrita em um diagrama

Projeto Integrado – Engenharia de *Software* - PMV

			de contexto
15 / 03 / 22	18 / 03 / 22	14. Estudar e Escolher frameworks que serão usados no projeto	Ter os frameworks de front e back definidos para o projeto.
18 / 03 / 22	31 / 03 / 22	15. Desenvolver o front das três principais regras de negócio escolhidas para produção do MVP	Estar com o código desenvolvido e revisado das três principais atividades escolhidas
31 / 03 / 22	30 / 04 / 22	16. Escrever uma das funcionalidades de back-end escolhidas para o MVP	Ter o código de uma funcionalidade escolhida para o MVP construído revisado e integrado com o Front
01 / 05 / 22	30 / 05 / 22	17. Desenvolver as outras duas funcionalidades do back escolhidas para o MVP	Ter o código da funcionalidade escolhida para o MVP construído revisado e integrado com o Front
30 / 05 / 22	20 / 06 / 22	18. Estudar e criar processo de autenticação e segurança das requisições	Estar com o processo de segurança de login e requisições escolhido e implementado para o back e front
20 / 06 / 22	10 / 07 / 22	19. Criar plano de teste do sistema	Ter um plano e relatório de teste para cada um dos casos de uso
10 / 07 / 22	20 / 07 / 22	20. Revisar todo o projeto verificar necessidade de Funções adicionais	Ter revisado todo o projeto. Caso tenha necessidade de implementar alguma função adicional ter o plano e o cronograma criado

2. Introdução

Segundo a Confederação Nacional do Comercio de Bens, Serviços e Turismo o número de brasileiros endividados bateu recorde histórico. Pesquisa de Endividamento e Inadimplência do Consumidor, realizada em julho de 2021, chegou ao resultado que 71,4% do total dos consumidores carregam alguma dívida ativa.

A constante elevação de taxa básica de juros no Brasil e a alta taxa de endividamento. Que segundo o banco central cerca de 58,5% da renda do brasileiro está comprometida com alguma dívida ativa. Somado a estudos realizados por economistas como Marcela Kawauti, demonstram que as projeções futuras são de aumento da parcela de brasileiros com dívida ativa nos próximos anos.

Os economistas da Serasa Luiz Rabi e Luiza Rodrigues em seu artigo que desmistifica o endividamento dizem que, não existe formula mágica para acabar com a dívida. E sugerem que existe várias estratégias que os consumidores podem incorporar no seu dia a dia para auxiliar a reduzir as dívidas e sanar elas. Estratégias como fazer diagnostico e conhecimento da dívida, trocar todas por uma só, anotar todos os gastos e fazer um controle da vida financeira, dentre outras opções.

O brasileiro é um povo que passa por inúmeros sofrimentos, desde descaso dos órgãos públicos até desastres naturais. Apesar de tudo é senso comum que mesmos com todas as dificuldades o brasileiro é um povo guerreiro, que em seu dia a dia trabalha duro para conquistar suas coisas. E nada mais justo que o auxiliar a conseguir livra-se de suas dívidas e conquista uma boa saúde financeira de suas economias.

O objetivo deste trabalho e criar um sistema que auxilie a população controlar sua vida financeira, de gastos e ganhos, sendo objetivos específicos:

- Criar uma aplicação para o usuário adicionar as suas despesas diárias.
- Disponibilizar para o usuário uma forma de acompanhar sua vida financeira.
- Trazer um resumo de gastos e economia.
- Proporcionar ferramentas para ajudar a manter controle de suas finanças.

3. Definição Conceitual da Solução

Cerca de três em quatro brasileiros possui alguma dívida. Fatores como alta taxa de desemprego dos últimos anos, juros elevados e a crise causada pela pandemia contribuíram para estes números elevados. Isto é o que revela a pesquisa de Endividamento e Inadimplência, divulgada pela Confederação Nacional de Bens, Serviços e Turismo.

Ante a este cenário econômico, oferta de crédito facilitadas, por governo e instituições privadas e falta de conhecimento sobre a educação financeira. Apontam cenários pessimistas para os próximos anos. Como descrito no artigo científico educação financeira e inadimplência no Brasil escrito por Marcelo Santos Chaves do Instituto Federal de Educação, Ciência e Tecnologia do Pará

Números elevados como os citados acima, levantam questões como. Qual o melhor caminho para se livrar das dívidas e fugir do endividamento? Como ter uma vida financeira saudável? Como controlar as dívidas? Como gastar conscientemente?

No livro como fazer seu orçamento mensal realize suas metas financeiras com exercício da autora Kelen de Oliveira. Ela descreve alguns passos para ajudar a ter um controle maior da saúde financeira e sair do endividamento. Dentre os passos estão anotar todos os gastos e ganhos, classificar receitas e despesas, apurar o saldo entre ganhos e despesas, criar uma reserva de emergência, definir metas, entre outros.

Passos como os descritos acima são muito parecidos com os encontrados no artigo Cinco dicas para melhorar sua saúde financeira, escrito por Rodrigo Martiniano da Rocha, colaborador da Uniprime. Dicas semelhantes também são encontrados em vários blogs e vídeos na internet, que tem o objetivo de ajudar o brasileiro a ter uma vida financeira no azul e conquistar seus desejos como casa própria, carro novo etc.

Podemos concluir que um fator importante que leva o brasileiro a se endividar, é a falta de controle de suas próprias finanças. Onde em inúmeros casos a população não tem o conhecimento de qual o valor de suas despesas e o que ainda pode gastar. Como apontado em pesquisa educação financeira e a gestão do orçamento pessoal realizada pelo Serviço de Proteção ao Crédito (SPC Brasil) e pela Confederação Nacional de Dirigentes Lojistas (CNDL). Que apontam que 45,8% dos entrevistados não realizam um controle do seu orçamento e 29,3% fazem esse orçamento apenas ‘de cabeça’. Recorrendo a métodos pouco confiáveis para organizar suas finanças.

Diante deste cenário abre espaço para uma aplicação que podem auxiliar a população a ter um controle de suas despesas, através de um sistema que possibilite incluir seus gastos e recebimento do dia a dia de forma simplificada. Parecido com os softwares de controle financeiro encontrados em empresas. Com o diferencial de ser uma aplicação focada para o uso pessoal, de forma simples. Não tornando cansativo do usuário preencher.

O projeto consiste em desenvolver um sistema de controle de gastos pessoais que possibilita maior visualização e controle de receita versus despesas, gerenciar suas metas financeiras, sugestões financeiras para o utilizador da aplicação, aumentar o conhecimento sobre finanças. Com isso o usuário poderá ter uma visão mais completa de como anda suas finanças e qual é a melhor forma de dar os próximos passos seja para liquidar uma dívida ou conquistar a suas metas.

3.1 Diagrama de Casos de Uso

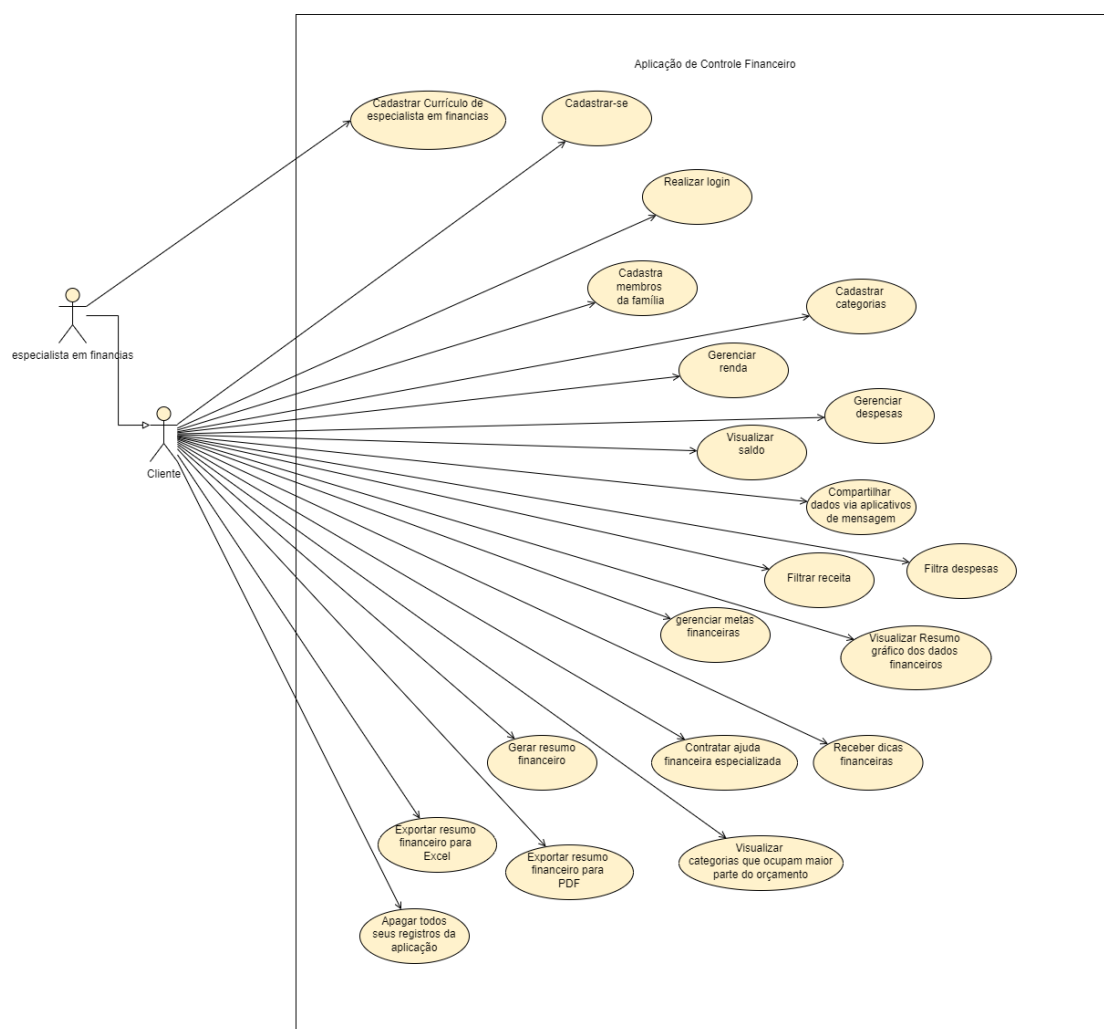
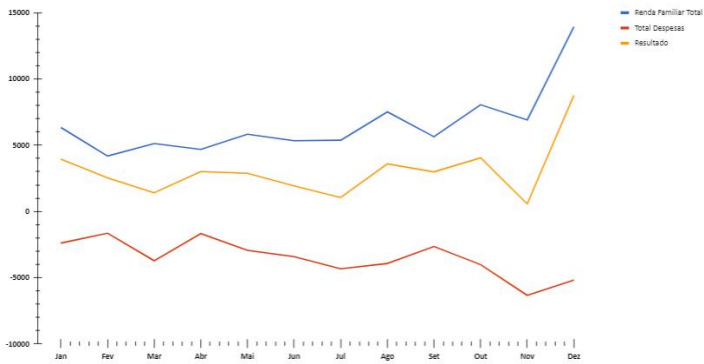
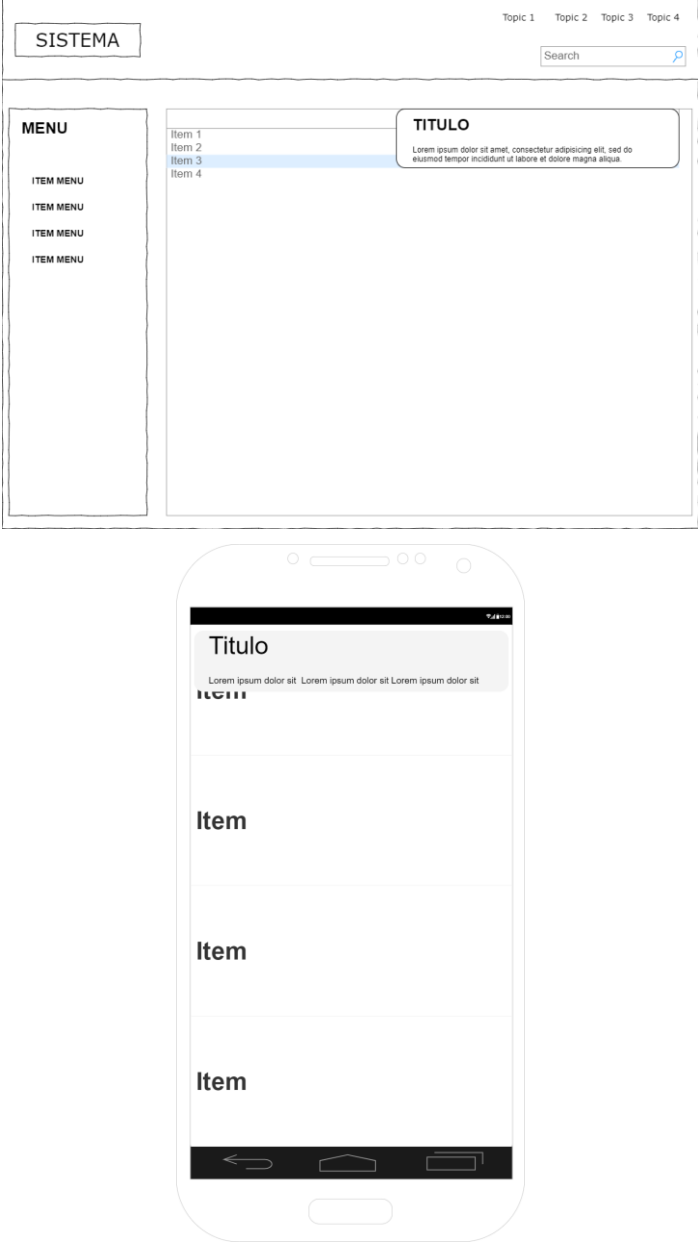


Figura 1 - Visão do diagrama de caso de uso. Fonte: <https://github.com/matheusCalaca/sistema->

3.2 *Requisitos Funcionais*

ID	Descrição Resumida	Dificuldade (B/M/A) *	Prioridade (B/M/A) *
RF01	O usuário deve auto cadastrar-se no sistema, podendo escolher um perfil de cliente, ou especialista financeiros.	B	A
RF02	O usuário pode cadastrar membros da família que tem registro na aplicação.	A	B
RF03	O Usuário pode compartilhar acesso aos seus gastos e despesa com outros usuários que estejam cadastrados com membros de sua família ou especialista contratado por ele.	A	B
RF04	O usuário pode incluir deletar e atualizar categorias de despesas	B	A
RF05	A aplicação deve conter meios de pagamentos cadastrados como padrão. Os meios de pagamento padrão deveram ser: <ul style="list-style-type: none"> • Cartão Crédito • Cartão Debito • Boleto • Dinheiro • PIX 	B	A
RF06	O usuário pode incluir deletar e atualizar suas receitas. Cada receita deve conter uma data um título o valor da receita e uma descrição opcional caso ele deseja incluir.	B	A
RF07	O sistema deve permitir que ao usuário adicione uma receita inicial. Receita que ele possui antes de iniciar o uso da aplicação	M	M
RF08	O usuário pode incluir deletar e atualizar suas despesas. O cadastro de despesa dever conter os seguintes dados: <ul style="list-style-type: none"> • Nome • Valor • Meio de pagamento • Onde foi feito a despesa • Por qual motivo foi realizado aquela despesa • Categoria da despesa • Data que foi feito a despesa 	M	A
RF09	O sistema deve calcular automaticamente as despesas e receita do cliente.	M	M

RF10	<p>A aplicação deve mostrar o saldo mensal do usuário</p> <p>Total = Σ receitas do mês corrente - Σ Despesas do mês corrente</p>	M	M
RF11	<p>O sistema deve mostra o saldo total considerando todos os meses de uso.</p> <p>Total = Σ receita - Σ despesa</p>	M	M
RF12	<p>O sistema deve ter a funcionalidade de compartilhar os dados financeiro do cliente caso ele solicite. Via aplicativo de mensagem, exemplo: WhatsApp, Facebook messenger, entre outros</p>	A	B
RF13	<p>A aplicação deve permitir que o usuário filtre suas despesas e recitas por data, categoria, valor, nome do item, meio de pagamento</p>	M	M
RF14	<p>Disponibilizar para o usuário um gráfico com suas despesas e suas receitas</p> 	A	M
RF15	<p>A aplicação deve deixar o usuário criar, deletar e atualizar metas financeira sendo obrigatório definir um valor para a meta. Para cada meta criada o usuário deve definir uma data futura como previsão de conclusão desta</p>	M	B
RF16	<p>Para usuários com perfil especialista, a aplicação deve ter a opção de cadastro atualização de dicas sobre finança, que serão exibidas para usuários com perfil comum.</p>	B	B
RF17	<p>O sistema deverá mostrar dicas financeiras cadastradas de forma aleatórias. Em formato de notificação, na área logada dá aplicação. Essas notificações tem o intuito de aumentar o conhecimento do usuário sobre termos e estratégias financeiras. Segue exemplos:</p>	B	B

			
RF18	O sistema deve ter uma área onde profissionais especializados podem cadastrar seu currículo, com nome, telefone, e um resumo de seu currículo.	M	B
RF19	O sistema deve conter uma área para que clientes possam visualizar os dados de profissionais de especialistas financeiros cadastrados na aplicação com seu telefone de contato.	M	B
RF20	<p>O sistema deve possibilitar que o cliente visualize um resumo com o valor total gasto em cada categoria em ordem decrescente. Este resumo deve ser construído tanto por mês como anualmente.</p> <p>Formula:</p> <p>Mensal: Σ gatos por categoria mensal</p>	M	M

	Anual: Σ mensal		
RF21	A aplicação deve destacar as três categorias que representam as maiores despesas do cliente, com uma cor distinta para que ele possa buscar uma forma de reduzir o gasto nelas ou encontrar uma solução de contorno.	M	B
RF22	A aplicação deve mostrar um resumo financeiro do usuário, com o total de sua receita, o total de suas despesas tanto mensal com anual. Além de um resumo detalhado com todos seus gastos mensais. O cliente deve ter a opção de baixar este resumo financeiro em PDF ou exportar todos os dados para um Excel	A	M
RF23	A aplicação deve conter um botão para o cliente deletar todos os dados cadastrados.	M	B

* B = Baixa, M = Média, A = Alta.

3.3 *Requisitos Não-funcionais*

ID	Descrição	Prioridade B/M/A
RNF01	O sistema deve apresentar tempo de resposta abaixo de 200 ms no processamento de 90% das operações de consulta.	M
RNF02	O sistema deve apresentar uma disponibilidade de 95% do tempo online durante doze meses	A
RNF03	As primeiras informações do sistema devem aparecer em até 3s após o cliente acessar o mesmo	B
RNF04	O sistema deve funcionar em navegadores, e dever ser responsivo para adaptar nas telas de computadores e celulares	A
RNF04	O sistema deve obedecer a todas as leis de proteção de dados vigente em território brasileiro	M
RNF06	O sistema deve possuir mecanismo de segurança nas requisições entre o front-end. e o back-end	A
RNF07	O sistema deve utilizar a linguagem de programação Java para a escrita do back-end e a linguagem de programação Javascript para a escrita do front-end. Podendo utilizar frameworks para facilitar o desenvolvimento.	B

4. Protótipo Navegável do Sistema

Desenvolvido um protótipo navegável do sistema que está disponível em <https://app.moqups.com/8ZsfM1H1KGRDxuvSzi8tnv8th5R67XTv/view/page/aa9df7b72> também foi criado um arquivo que detalha o fluxo que está disponível em PDF e pode ser baixado neste link <https://github.com/matheusCalaca/sistema-financeiro-frontend/blob/main/resource/telas/fluxodetelas.pdf> para melhor visualização.

Para maior detalhe do fluxo de protótipo de interface, foi criado uma apresentação em vídeo que está disponível para visualização através do link <https://youtu.be/QGP4Mr9GkJk> ou podendo baixa-la em mp4 pelo link <https://github.com/matheusCalaca/sistema-financeiro-frontend/blob/main/resource/telas/videoDoFluxoDeInterface.mp4>



Figura 2 - Visão Geral do fluxo de telas. Fonte: <https://github.com/matheusCalaca/sistema-financeiro-frontend/blob/main/resource/telas/fluxodetelas.png>

5. Diagrama de Classes de Domínio

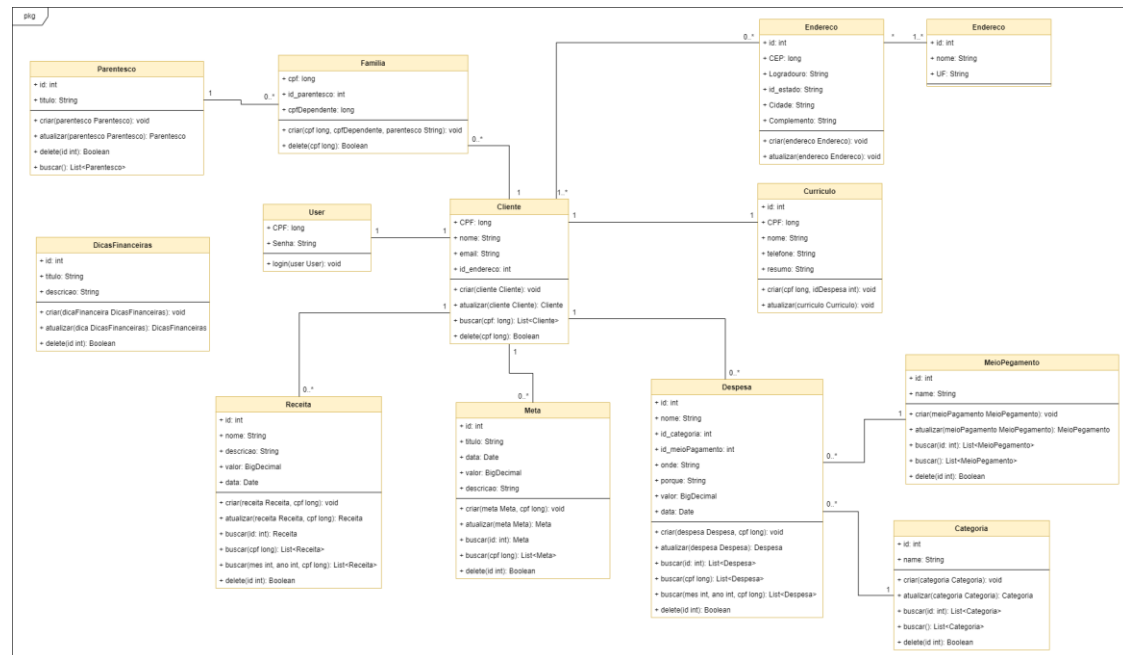


Figura 3 - Visão Geral do Diagrama de classe de domínio. Fonte: <https://github.com/matheusCalaca/sistema-financeiro-frontend/blob/main/resource/diagrama%20de%20classe/diagrama%20de%20classe%20de%20dominio.jpg>

6. Arquitetura da Solução

6.1 Padrão Arquitetural

Os padrões arquiteturais tem objetivo de separar as responsabilidades de visualizar processar e gerenciar dados para aplicativos de interface de usuário, para facilitar a manutenção e desenvolvimento do software.

Dentre os padrões arquiteturais um dos mais usados é o model view controller MVC, que é um padrão que foi projetado originalmente por trygve reenskaug, que passou a ser descrito com profundidade no livro “Padrões de Design: Elementos do Software Orientado a Objeto Reutilizável”.

O padrão MVC divide o sistema em três módulos de responsabilidade são eles Model que tem a responsabilidade de armazenar a lógica da aplicação. View é a camada para apresentar os dados da aplicação. Controller é a camada de controle da aplicação, responsável por fazer a ligação entre a model e a view.

Com base em estudos e levando em consideração o padrão descrito. Foi decidido usar a arquitetura MVC para a construção do projeto. Pois se mostra uma

arquitetura coesa, bem dividida, de fácil compressão e difundida no mercado. Atendendo todos os requisitos para a construção da aplicação além de estar pronta para que a aplicação possa crescer.

6.2 C4 model - Diagrama de Contexto

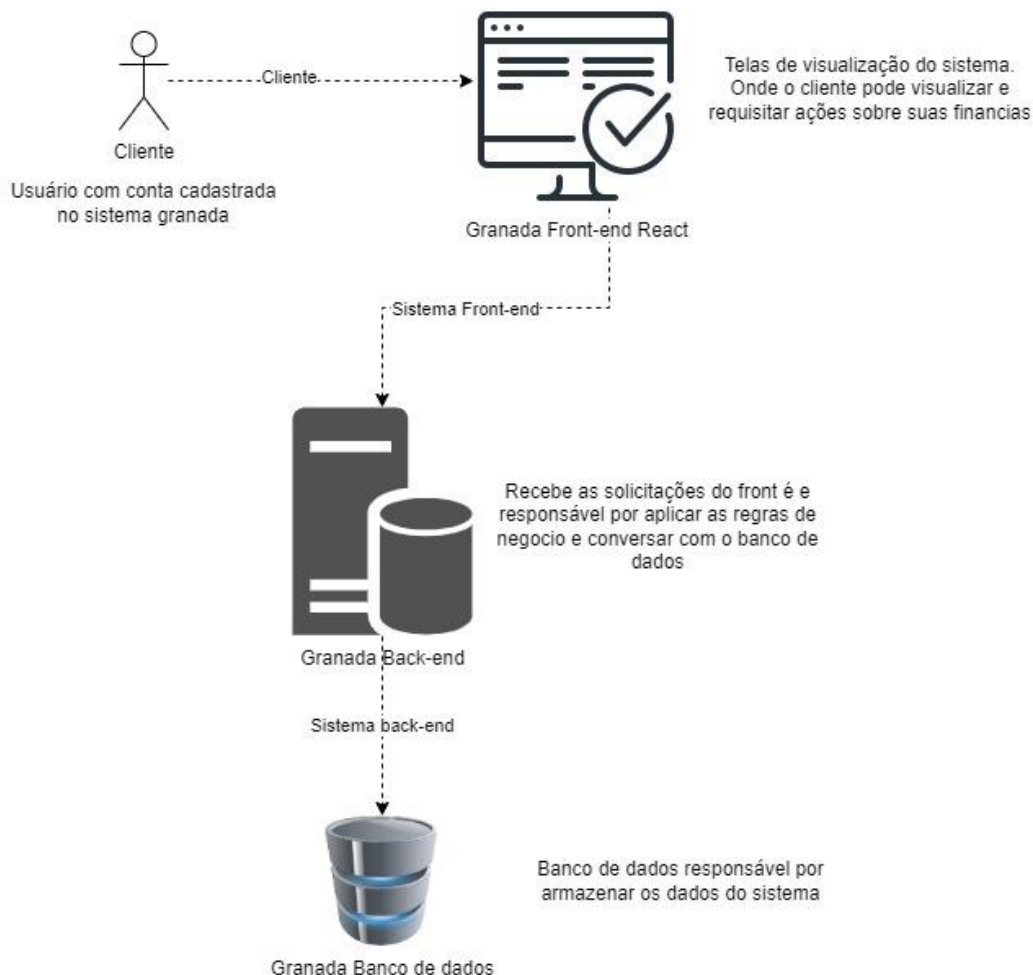


Figura 4 - Visão Geral do Diagrama de classe de Contexto. Fonte:

<https://github.com/matheusCalaca/sistema-financeiro-frontend/blob/main/resource/Diagrama%20de%20contexto%20C4/Diagrama%20de%20contexto.jpg>

A Figura 4 mostra o diagrama de contexto da solução proposta, com todos os seus principais módulos e fluxo de conexão. Além de descrever a principal funcionalidade de cada um dos módulos.

7. Frameworks de Trabalho

Frameworks são instruções e recursos pré-programadas que tem o intuito de tornar o sistema programável mais fácil e rápido, utilizando reaproveitamento de código, e disponibilizando para o desenvolvedor ferramentas para facilitar a criação do sistema.

Para a criação das telas foi escolhido a biblioteca de desenvolvimento React. Que permite liberdade na construção do código, facilita a criação de interfaces de usuário criativas. Permitindo que projete soluções simples para cada estado da aplicação, facilitando a renderização e reutilização de componente. Conforme definição descrita em seu site <https://reactjs.org/> acessado em 04/08/2022.

Dentre os principais motivos para a escolha do React framework, está a facilidade de desenvolvimento de telas, alto desempenho com virtual DOM, ser um framework amigável para SEO e uma tecnologia que já está difundida no mercado e possui uma base solida de comunidade. Como descrito no artigo “REACT: o que é, como funciona e porque usar e como aprender” escrito por Ugo Roveda.

Para o back-end foi escolhido a linguagem de programação Java com o framework Spring. Java e descrito em sua documentação oficial como uma linguagem de programação, de início humilde que foi adquirida pela à Oracle, cujo e responsável atualmente por manter e organizar as releases da linguagem.

Conforme descrito em sua documentação Java e uma linguagem baseada em classes e orientada a objeto, com foco em segurança, portabilidade e performance. E uma linguagem que possui uma extensa biblioteca de rotinas e APIs além de um poderoso gerenciamento automático de memória.

Levando em consideração todo o escopo do projeto a ser desenvolvido levou a escolha de Java com a linguagem de programação para o back-end, além de dados que mostram ser uma das linguagens mais utilizadas em 2022 e está presente em grande parte dos sistemas que utilizamos no nosso dia a dia, conforme descrito em pesquisa anual lançada pela a empresa Stack Overflow que pode ser acessada no site <https://survey.stackoverflow.co/2022/#technology> . Pesquisa que é e pode ser usada para obter grandes insights sobre o status do ecossistema de programação. Que ajudaram no fortalecimento da escolha da linguagem.

Para framework foi escolhido o Spring que tem como objetivo tornar a programação Java mais rápida, fácil e segura. Tendo um foco em velocidade, simplicidade e produtividade.

Spring e um framework flexível que abrangem várias bibliotecas de terceiros em sua essência, possuir preocupações com velocidade produtividade em sua construção e um framework compatível com várias arquiteturas com a de micros serviços e arquiteturas assíncronas entre outras. Ele já possui uma base solida de aplicações no mercado um uma comunidade grande, que facilita na troca de informações e aprendizagem.

Esse e um framework que se encaixa perfeitamente em todos os requisitos levantados, que vai facilitar o desenvolvimento do back-end. Trazendo todos os benefícios citados.

8. Estrutura Base do Front-End



Figura 5 - Visão Geral do layout mestre. Fonte:

<https://github.com/matheusCalaca/sistema-financeiro-frontend/blob/main/resource/telas/layout%20mestre.png>

Utilizando como base os requisitos levantados e todo o estudo feito foi criado um layout mestre que apresenta o resumo das transações realizadas no mês corrente, com um resumo gráfico e valor monetário aparecendo, para o usuário ter uma visão clara de seus gastos. E na parte superior um menu com as principais regras de negócio e funções do sistema, controle de Receitas, controle de Despesas e Metas do usuário. Para que o sistema seja flexível foi pensado um menu retrátil quando acessado em telas menores e um maior espaçamento entre os gastos de Receitas e Despesas presente no Layout Mestre. Conforme demonstrado na imagem abaixo

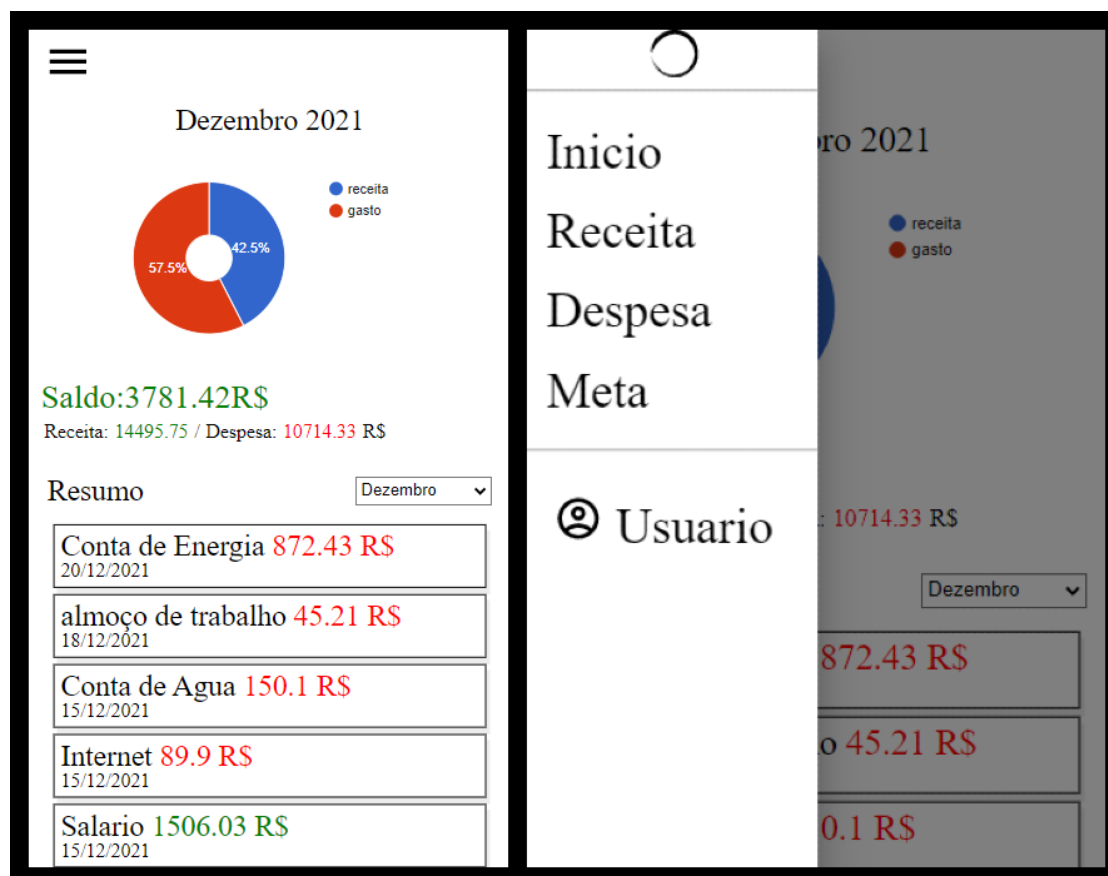


Figura 6 - Visão Geral do layout mestre dispositivos mobile. Fonte:

<https://github.com/matheusCalaca/sistema-financeiro-frontend/blob/main/resource/telas/layout%20mestre%20mobile.png>

9. Apropriação de Horas no Projeto

Histórico de apropriação de horas		
Data do registro	Atividade	Quantidade de horas
10/12/2021	Pesquisa e Estudos sobre a ideia de desenvolver um sistema de controle financeiro	20 Horas
08/01/2022	Estudar qual é a melhor forma de aplicar as propostas de ajuda financeira no sistema.	18 Horas
15/01/2022	Criar um diagrama de caso de uso para o sistema	23 Horas
30/01/2022	Levantar os requisitos funcionais para a construção do sistema	33 Horas
05/02/2022	Fazer um levantamento dos requisitos não funcionais	08 Horas
07/02/2022	Revisar tudo que foi feito e escrito com a intenção de encontrar algo que passou despercebido	04 Horas
10/02/2022	Estudo de interfaces de sistemas financeiro	10 Horas
25/02/2022	Desenhar um protótipo da interface de navegação do cliente	36 Horas
08/03/2022	Construir o diagrama de classe de domínio	27 Horas
10/03/2022	Revisar o protótipo de navegação do sistema	05 Horas
08/05/2022	Levantamento e estudo para a escolha do padrão arquitetural.	08 Horas
10/05/2022	Desenvolvimento do Diagrama de contexto	05 Horas
12/05/2022	Estudo e levantamento do framework de back-end e front-end para o projeto	12 Horas
02/06/2022	Desenvolvimento da estrutura inicial do front-end	10 Horas
04/06/2022	Desenvolvimento da página principal	12 Horas
07/06/2022	Desenvolvimento da tela de login	10 Horas
27/06/2022	Desenvolvimento das tabelas para demonstrar os dados de Despesa, Receita e Metas	40 Horas
03/07/2022	Desenvolvimento Gráfico	08 Horas
28/07/2022	Desenvolvimento da estrutura inicial do back-end	08 Horas
01/08/2022	Desenvolvimento da estrutura de banco de dados	04 Horas
05/08/2022	Desenvolvimento dos end-points de Despesa	12 Horas
08/08/2022	Desenvolvimento dos end-points de Receita	10 Horas
11/08/2022	Desenvolvimento dos end-points de Metas	08 Horas
13/08/2022	Implementação do Back-end no front-end	15 Horas

10. Código da Aplicação

Desenvolvido um protótipo navegável do sistema que está disponível em <https://granada-matheus-calaca.vercel.app/>, nesta versão disponibilizada para teste não

foi implementado gerenciamento de perfil de usuário, para acessar o sistema basta clicar em “Entrar”. Também foi disponibilizado um link para acessar o protótipo de back-end <https://granada-matheus-calaca-puc.herokuapp.com/swagger-ui/#/>

para acesso ao seu código fonte foi disponibilizado o seguinte link para o front-end <https://github.com/matheusCalaca/sistema-financeiro-frontend> e para o back-end foi disponibilizado o link <https://github.com/matheusCalaca/sistema-financeiro-backend>.

Para maior detalhe do, foi criado uma apresentação em vídeo que está disponível para visualização através do link <https://youtu.be/p8ZLN0ctw-8> ou podendo baixa-la em mp4 pelo link <https://github.com/matheusCalaca/sistema-financeiro-frontend/blob/main/resource/Tcc%20etapa%202/video%20tecc%20etapa2.mp4>

11. Referências

GLOBO, g1.globo.com, 2021. **Endividamento chega a record de 71,4% dos brasileiros, segundo a CNC.** Disponível em: <<https://g1.globo.com/economia/noticia/2021/08/05/endividamento-chega-a-recorde-de-714percent-dos-brasileiros-segundo-a-cnc.ghtml>> Acesso em: 28 de dez. 2021.

CNN BRASIL, cnnbrasil.com.br, 2021. **Juros mais altos podem agravar endividamento no Brasil, dizem economistas.** Disponível em: <<https://www.cnnbrasil.com.br/business/juros-mais-altos-podem-agravar-endividamento-no-brasil-dizem-economistas/>> Acesso em: 28 de dez. 2021.

R7, noticias.r7.com, 2021. **Quer ficar no azul? Economistas ensinam.** Disponível em: <<https://noticias.r7.com/economia/quer-ficar-no-azul-economistas-ensinam-a-acabar-com-todas-as-dividas-26012014>> Acesso em: 28 de dez. 2021.

DIGITAL MONEY INFORME, digitalmoneyinforme.com.br, 2021. **Economista-chefe da SERASA desmistifica o endividamento.** Disponível em: <<https://www.digitalmoneyinforme.com.br/economista-chefe-da-serasa-desmistifica-o-endividamento/>> Acesso em: 28 de dez. 2021.

ASSOCIAÇÃO COMERCIAL E EMPRESARIAL, acecdlunai.com.br, 2021. **SPC Brasil dá dicas para o consumidor.** Disponível em:

<https://www.acecdlunai.com.br/site/conteudo/865-SPC_BRASIL_Da_DICAS_PARA_O_CONSUMIDOR> Acesso em: 28 de dez. 2021.

CORREIO BRAZILIENSE, correiobrasiliense.com.br, 2022. **Três a cada quatro brasileiros estão endividados, mostra pesquisa.** Disponível em: <<https://www.correiobrasiliense.com.br/economia/2021/10/4953473-tres-a-cada-quatro-brasileiros-estao-endividados-mostra-pesquisa.html>> Acesso em: 03 de jan. 2022.

ECONOMIA NO BRASIL, eumed.net, 2022. **Educação financeira e inadimplência no Brasil.** Disponível em: <<https://www.eumed.net/cursecon/ecolat/br/15/inadimplencia.html>> Acesso em: 03 de jan. 2022.

OLIVEIRA, Kelen Oliveira. **Realize suas metas financeiras: Como fazer um orçamento mensal, com exercícios.** 2021.

SISTAMA OCEPAR, paranacooperativo.coop.br, 2022. **Cinco dicas para melhorar sua saúde financeira.** Disponível em: <<https://www.paranacooperativo.coop.br/ppc/index.php/sistema-ocepar/comunicacao/2011-12-07-11-06-29/ultimas-noticias/136616-artigo-cinco-dicas-para-melhorar-sua-saude-financeira>> Acesso em: 03 de jan. 2022.

SPC BRASIL, spcbrasil.org.br, 2022. **46% dos brasileiros não controlam seu orçamento, revela pesquisa do SPC Brasil.** Disponível em: <https://www.spcbrasil.org.br/uploads/st_imprensa/release_educacao_financeira_v7.pdf> Acesso em: 04 de jan. 2022.

MEDIUM, medium.com, 2022. **MVC vs MVP vs MVVM.** Disponível em: <<https://levelup.gitconnected.com/mvc-vs-mvp-vs-mvvm-35e0d4b933b4#:~:text=References%20%E2%80%94%20In%20MVC%2C%20the%20View,entry%20point%20is%20the%20View.>> Acesso em: 11 de jun. 2022.

MUNDO DEVOPS, mundodevops.com, 2022. **Frameworks React: Conheça os mais utilizados** disponível em: <<https://mundodevops.com/blog/react->

Granada

frameworks/#:~:text=J%C3%A1%20o%20React%2C%20por%20si,para%20resolver%20um%20problema%20espec%C3%ADfico. > Acesso em: 18 de jun. 2022.

MUNDO DEVOPS, kenzie.com, 2022. **React: O que é, como funciona e porque usar e como aprender** Disponível em: <<https://kenzie.com.br/blog/react/#:~:text=Como%20funciona%20o%20React%3F,do%20que%20chamamos%20de%20componentes.>> Acesso em: 04 de ago. 2022.

REACT, reactjs.org, 2022. **Frameworks React** Disponível em: <<https://reactjs.org/>> Acesso em: 02 de ago. 2022.

JAVA, java.com, 2022. **Java Documentação** Disponível em: <https://www.java.com/pt-BR/download/help/whatis_java.html> Acesso em: 05 de ago. 2022.

SPRING, spring.io, 2022. **Why Spring** Disponível em: <<https://spring.io/why-spring>> Acesso em: 05 de ago. 2022.