**FACULDADE DE TECNOLOGIA DE ARAÇATUBA**

**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**DIOGO ESPINDOLA DA SILVA**

**GABRIEL HENRIQUE SOUZA DE CARVALHO**

**FINANCE**: GESTÃO INTELIGENTE

**ARAÇATUBA - SP**

**2022**

**FACULDADE DE TECNOLOGIA DE ARAÇATUBA**

**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**DIOGO ESPINDOLA DA SILVA**

**GABRIEL HENRIQUE SOUZA DE CARVALHO**

**FINANCE**: GESTÃO INTELIGENTE

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Araçatuba, do Centro Estadual de Educação Tecnológica Paula Souza, como requisito parcial para conclusão do curso de Análise e Desenvolvimento de Sistemas sob a orientação do Prof. Me. Alexandre Marcelino da Silva.

**ARAÇATUBA - SP**

**2022**

**RESUMO**

As finanças pessoais é um assunto de extrema importância, assunto este que, segundo algumas pessoas e até mesmo profissionais da área, deveria ser ensinado no ensino médio. Em um cenário caótico como o atual de pós-pandemia, onde a inflação mesmo tendo sido listada no 4º lugar como a menor inflação do G20 segundo o site poder360 (2022), onde fazer as compras do mês pode custar até o dobro do valor de 3 anos atrás, onde se há demissões e layoffs em massa alcançando pouco mais de 9 milhões de desempregados no terceiro trimestre de 2022 segundo o IBGE (2022), foi feita uma pesquisa em conjunto com várias análises, a viabilidade do desenvolvimento de uma solução que seja eficaz no principal problema do brasileiro atualmente, que é organizar suas finanças pessoais e evitar problemas que a não organização pode causar.

**Palavras-Chave:** Finanças; Pessoais; Pandemia; Inflação.

**ABSTRACT**

Personal finance is a subject of extreme importance, a subject that, according to some people and even professionals in the area, should be taught in high school. In a chaotic scenario like the current post-pandemic one, where inflation even though it was listed in 4th place as the lowest inflation in the G20 according to the website Poder360 (2022), where shopping for the month can cost up to twice the value of 3 years ago, when there were mass layoffs and layoffs reaching just over 9 million unemployed people in the third quarter of 2022 according to the IBGE (2022), a survey was carried out in conjunction with several analyses, the feasibility of developing a solution that be effective in the main problem of Brazilians today, which is to organize their personal finances and avoid problems that non-organization can cause.

**Key words:** Finance; Personal; Pandemic; Inflation.

**LISTA DE FIGURAS**

Figura 1 – Caso de Uso – Cadastrar Usuário(exemplo)..................................................24

**LISTA DE QUADROS**

Quadro 1 – Caso de Uso – Cadastrar Usuário(exemplo).................................................15

Quadro 2 – Requisito funcional – Cadastro de informações adicionais...................................16

Quadro 3 – Requisito funcional – Balanço de gastos.............................................................16

Quadro 4 – Requisito funcional – Gráfico em linha de gastos................................................16

Quadro 5 – Requisito funcional – Gráfico de categoria de gastos...........................................17

Quadro 6 – Requisito funcional – Criação de planejamento mensal........................................17

Quadro 7 – Requisito não funcional – Escalabilidade.....................................................17

Quadro 8 – Requisito não funcional – Interface interativa UI/UX..........................................17

Quadro 9 – Requisito não funcional – Banco de dados não-relacional....................................18

Quadro 10 – Lista de Casos de Uso.................................................................................18

**LISTA DE SIGLAS**

CSS – Cascading Style Sheets

**SUMÁRIO**

**1 INTRODUÇÃO..........................................................................................................12**

**2 REVISÃO DA LITERATURA.................................................................................13**

**1 INTRODUÇÃO**

O Brasil é um dos países que mais possuem

**2** **REVISÃO DA LITERATURA**

**2.1 – Contextualização**

2.1.1 - Economia

A economia de forma geral é um assunto que deve ser levado em extrema consideração, entretanto a finança pessoal é de suma importância para a sobrevivência das pessoas diante dos problemas enfrentados no dia-a-dia, isto sem contar a pandemia global causada pelo vírus SARS-CoV-2 e suas variantes que fora nomeada por Covid-19.

Segundo Juliana, Bárbara e Thais (2020), pudemos acompanhar uma crise de forma mundial, onde a divulgação do PIB real de 2019 mostrou ser menor do que o esperado, onde apresentou um crescimento de 1.1% ao ano, que ia contra a previsão do *Focus* que era de 2.5% só no início de 2019. Tendo estas informações como parâmetro, podemos ter uma breve noção de como as finanças pessoais pode ser importante, pois uma vez que a economia sofre alterações de forma negativa, devemos nos ajustar ao novo panorama e buscar gerir da melhor forma as finanças pessoais.

Na realidade brasileira, é evidente a má gerencia do aspecto financeiro tanto no âmbito privado quanto na administração pública. Segundo Carlos Thadeu, Chefe da Divisão Econômica da Confederação Nacional do Comércio de Bens e Turismo (CNC), o percentual de famílias que demonstraram ter dívidas a vencer em setembro de 2021 alcançou74, já a inflação, medida pelo INPC/IPCA, ultrapassa os 10% nos últimos 12 meses (GOMES, 2021).

Pouca atenção tem sido dada a transmissão de conhecimentos necessários para que o cidadão aja no contexto social em que se vê inserido. As crianças brasileiras que têm a oportunidade de ir à escola, estudam História, Geografia, Português, Matemática e Química, entretanto não tomam conhecimento de saber o que fazer com o dinheiro que irão conquistar ou qual a maneira correta para administrar o patrimônio que há de ser construído. Muitas dessas crianças têm grandes oportunidades, mas muitos vão acabar se tornando uma população majoritariamente endividada.

**2.2 - Tecnologias Utilizadas**

2.2.1 - Typescript

Segundo a Typescript Lang Docs. (2022), “Typescript tem uma relação incomum com JavaScript. Typescript oferece todas as features do Javascript, e uma camada adicional no topo deste: o sistema de tipos do Typescript.”. Em outras palavras, o Typescript é uma linguagem que utiliza da mesma sintaxe e funções do Javascript, porém, adiciona um ingrediente extra, que é a tipagem estática. O Javascript trata-se de uma linguagem com tipagem dinâmica, o que significa que o tipo da variável pode ser alterado durante o tempo de execução, o que pode facilitar a existência de erros de tipagem. O Typescript surgiu para corrigir estre problema, ela oferece uma tipagem estática (assim como linguagens como C, C++, Java e C#), evitando assim erros de tipagens sem precisar tornar a linguagem tão verbosa como as citadas como exemplo.

2.2.2 – Vue.js

Vue é um **framework progressivo** para a construção de interfaces de usuário que foi criado em 2014 por Evan You. Ao contrário de outros frameworks monolíticos, Vue foi projetado desde sua concepção para ser adotável incrementalmente. A biblioteca principal é focada exclusivamente na camada visual (view layer), sendo fácil adotar e integrar com outras bibliotecas ou projetos existentes. Por outro lado, Vue também é perfeitamente capaz de dar poder a sofisticadas Single-Page Applications quando usado em conjunto com ferramentas modernas e bibliotecas de apoio.

Vue está acumulando em volta de 10 milhões de downloads mensais, segundo o Stackshare (2021d), Vue é usado por: Alibaba; Trivago; 9GAG; GitLab; Stack. A quantidade de pacotes disponíveis no NPM para Vue é de 47 mil, enquanto para o Angular são apenas 12 mil.

2.2.3 - Node.js.

De maneira simples, Node.js é mais como um ambiente onde você pode utilizar o JavaScript para atuar como uma ferramenta de script do lado do servidor. Segundo a w3tech, Node.js é usado por pelo menos 30 milhões de websites no mundo todo, o que prova sua eficiência naquilo que ele propõe. Outro dado que mostra a eficiência do Node.js são as grandes empresas que usam dos seus recursos, alguns exemplos são: Amazon, Netflix, Reddit, LinkedIn e etc. De acordo com a StackOverflow 2021 Developer Survey, 36,19% dos desenvolvedores profissionais usaram Node.js para desenvolvimento extensivo no ano de 2021, e há garantia de que o número está crescendo em 2022. Isso porque o Node.js oferece uma gama abrangente de recursos como sistemas escalonáveis, simplicidade de desenvolvimento e etc.

Segundo Netguru *(A Netguru é uma empresa polonesa de desenvolvimento e consultoria de software fundada em 2008.),* o Node.js ajuda a reduzir os custos da produção de desenvolvimento em até 58%. Ter seu time de desenvolvimento frontend e Backend falando na mesma linguagem com certeza impacta na eficiência da produção dos projetos.

2.2.4 – MongoDB Atlas

O MongoDB Atlas, de acordo com o próprio site da MongoDB, se trata de um Banco de Dados de administração em versão remota, onde o usuário/desenvolvedor pode integrar a conexão ao banco de dados em sua aplicação através de um link de acesso, onde o mesmo necessita de ser enviado na própria URL do projeto seus dados de acesso, como Login e Senha. O MongoDB em si se trata de um banco NoSQL, cuja sigla se dá por “­*No Only SQL”*, o que traduzindo fica *“Não somente SQL”*, o que significa dizer que ele é um banco não relacional e não trabalha com tabelas assim como o SQL.

**3 METODOLOGIA DA PESQUISA**

**3.1 – Levantamento de requisitos**

O levantamento de requisitos trata-se do processo de identificação e coleta de informações para a estruturação de uma documentação. Nesta documentação, devem ser informadas todas as necessidades e expectativas dos usuários ou quaisquer pessoas envolvidas no sistema, produto ou serviço, sendo estes requisitos como funcionalidades, recursos, desempenho, segurança, usabilidade e vários outros atributos.

3.1.1 – Requisitos Funcionais

Os requisitos são separados em duas partes, sendo o primeiro os Requisitos Funcionais, que devem abordar as funcionalidades do sistema de acordo com uma determinada entrada. Basicamente é a parte que descreve o que o sistema deve fazer, como se comportar e quais ações ele deve executar de acordo com a interação de usuários ou outros sistemas. Geralmente, os recursos informados aqui são interfaces, menus, botões, funcionalidades e etc.

Quadro 1 – Requisito funcional – Cadastrar usuário.

|  |  |
| --- | --- |
| **Identificação:** RF01 | **Nome:** Cadastro de Usuário |
| **Descrição** | Realizar o cadastro dos dados dos usuários no sistema, para isso será necessário:  - Nome;  - Renda Mensal;  - Reserva monetária; |
| **Importância** | **Necessário** |

Quadro 2 – Requisito funcional – Cadastro de informações adicionais.

|  |  |
| --- | --- |
| **Identificação:** RF02 | **Nome:** Cadastro Informações adicionais |
| **Descrição** | Realizar o cadastro dos dados adicionais, para isso será necessário:  - Pessoas que compartilham da mesma renda;  - Estado Civil;  - Custo Mensais;  - Custos Anuais; |
| **Importância** | **Necessário** |

Quadro 3 – Requisito funcional – Balanço de gastos.

|  |  |
| --- | --- |
| **Identificação:** RF03 | **Nome:** Balanço de gastos |
| **Descrição** | Mostra as despesas e receitas do usuário e o saldo em contas cadastradas pelo usuário. |
| **Importância** | **Necessário** |

Quadro 4 – Requisito funcional – Gráfico em linha de gastos.

|  |  |
| --- | --- |
| **Identificação:** RF04 | **Nome:** Gráfico em linha de gastos |
| **Descrição** | Um gráfico em linha que mostra os gastos do usuário e o compara com o limite imposto pelo usuário. |
| **Importância** | **Necessário** |

Quadro 5 – Requisito funcional – Gráfico de categoria de gastos.

|  |  |
| --- | --- |
| **Identificação:** RF05 | **Nome:** Gráfico de categoria de gastos |
| **Descrição** | Gráfico que mostra os gastos do usuário por categoria (saúde, alimentação, etc). |
| **Importância** | **Necessário** |

Quadro 6 – Requisito funcional – Criação de planejamento mensal.

|  |  |
| --- | --- |
| **RF - 06** | **Nome:** Criação de planejamento mensal |
| **Descrição** | Um gráfico em linha que mostra os gastos do usuário e o compara com o limite imposto pelo usuário. |
| **Importância** | **Necessário** |

3.1.2 – Requisitos não Funcionais

Os requisitos não funcionais são especificações mais relacionadas a características do sistema que não estão diretamente relacionadas às suas funcionalidades, sendo estas, características que incluem qualidade, desempenho, confiabilidade, escalabilidade e vários outros aspectos. Estes requisitos são de suma importância para ajudar a garantir que o sistema possa atender os requisitos de qualidade e desempenho da aplicação durante sua operação.

Quadro 7 – Requisito não funcional – Escalabilidade.

|  |  |
| --- | --- |
| **Identificação:** RNF01 | **Nome:** Escalabilidade |
| **Descrição** | O sistema deve ter uma estrutura escalável, ou seja, ter uma infraestrutura que permita com que o sistema receba upgrades sem a necessidade de grandes refatorações estruturais. |

Quadro 8 – Requisito não funcional – Interface interativa UI/UX.

|  |  |
| --- | --- |
| **Identificação:** RNF02 | **Nome:** Interface interativa UI/UX |
| **Descrição** | É interessante que o sistema possua interfaces interativas que chamem a atenção do usuário, fazendo assim com que a interação do mesmo com a aplicação não algo “monótono”. |

Quadro 9 – Requisito não funcional – Banco de dados não-relacional.

|  |  |
| --- | --- |
| **Identificação:** RNF03 | **Nome:** Banco de dados não-relacional |
| **Descrição** | O sistema nada mais é do que um grande armazenador de dados com algumas outras funcionalidades. O que significa que não há a necessidade de relacionamento entre tabelas, pois os dados já estarão ligados a um único usuário de forma padronizada. |

\*Alterar o **RNF03** para o motivo do uso do banco de dados. Ex: mais simples, mais seguro, etc.

**3.2 – Análise e Projeto do Sistema**

3.2.1 – Lista de Casos de Uso

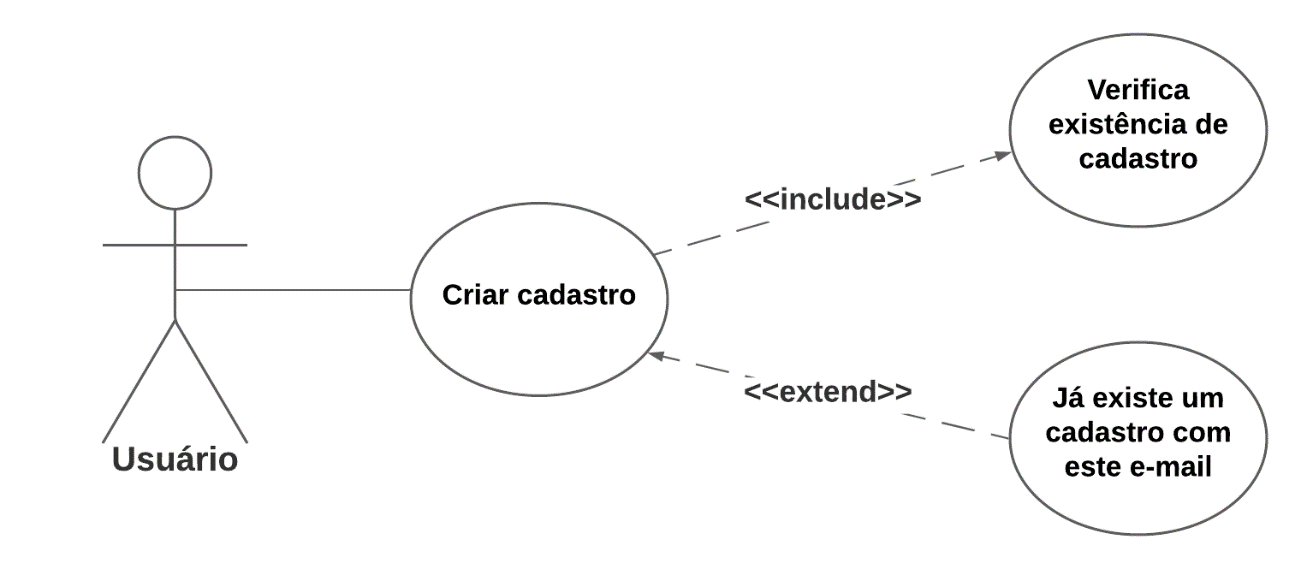
Quadro 10 – Lista de Casos de Uso.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nº** | **Descrição** | **CASO DE USO** | **ENTRADA** | **SAIDA** |
| **01** | O usuário cria uma nova conta de usuário | Criar cadastro | Dados de usuário | “Sua conta foi criada com sucesso!” |
| **02** | O usuário efetua login no sistema | Efetuar login | Dados de acesso | Acesso ao sistema |
| **03** | O usuário informa as contas bancárias que utilizada junto com um saldo inicial | Registrar contas bancárias | Nome da(s) conta(s) e saldo inicial |  |
| **04** | O usuário informa as contas a pagar do mês | Registrar contas a pagar | Contas a serem pagas |  |
| **05** | O usuário informa seus cartões de crédito junto com o saldo disponível e o valor da fatura | Registrar cartões de crédito | Cartões de crédito, saldo disponível e valor da fatura |  |
| **06** | O usuário deve informar as contas a receber | Registrar contas a receber | Contas a receber |  |
| **07** | O sistema deverá listar os dados informados | Informar listagem de dados | Todos os dados informados pelo usuário | Os dados informados pelo usuário |
| **08** | O sistema deverá possuir uma aba de informações por gráficos | Informar gráfico de dados | Todos os dados informados pelo usuário | Os dados informados pelo usuário em formato de gráfico |

Fonte: Autoral (2023).

3.2.2 – Diagrama de Caso de Uso

* + - 1. – Criar cadastro



\* Retirar os includes e deixar somente o “Criar cadastro”.

Figura 1 – Caso de Uso – Criar Cadastro

Fonte: Autoral (2023)

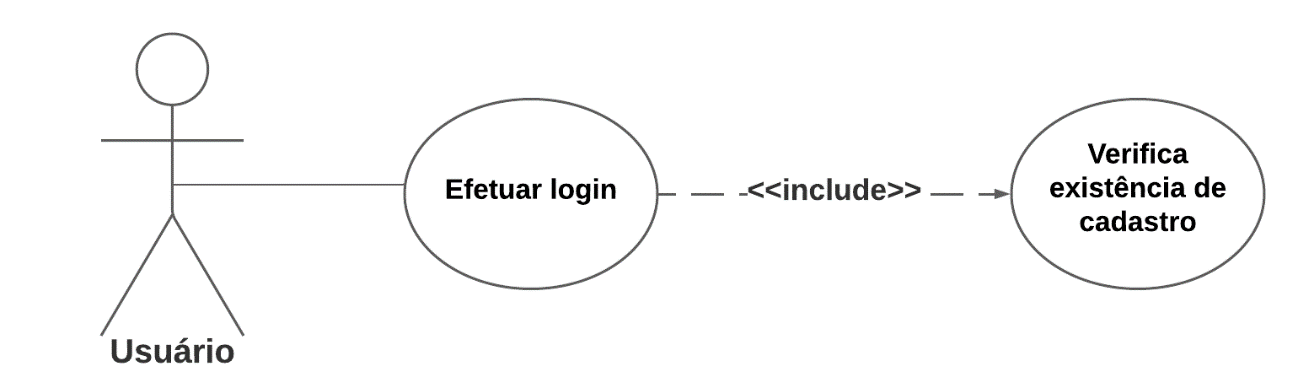
Fluxo Normal

1. O usuário acessa o sistema e inicia um novo cadastro;
2. O usuário informa seu e-mail;
3. O sistema verifica se o e-mail já está cadastrado;
4. O usuário informa os dados de usuário, <include> Verifica existência de cadastro;
5. O sistema grava os dados e exibe a mensagem “Cadastro realizado com sucesso!”;

Fluxo de Exceção

1. O sistema verifica se o e-mail já está cadastrado;
2. O sistema exibe a mensagem “E-mail já cadastrado!”;
3. O usuário informa um novo e-mail <Verifica existência de cadastro>;

3.2.2.2 – Efetuar login



\* Retirar o include e deixar somente o “Efetuar login”.

Figura 2 – Caso de Uso – Efetuar login

Fonte: Autoral (2023)

Fluxo Normal

1. Usuário informa e-mail e senha;
2. O sistema verifica a existência de cadastro;
3. O sistema verifica a veracidade dos dados informados;
4. O sistema permite o acesso do usuário ao dashboard de dados;

Fluxo de Exceção

1. O sistema identifica que o e-mail não foi registrado;
2. O sistema exibe a mensagem “E-mail não encontrado”;
3. O sistema analisa a senha informada;
4. O sistema exibe a mensagem “Senha incorreta”;
   * + 1. – Contas bancárias

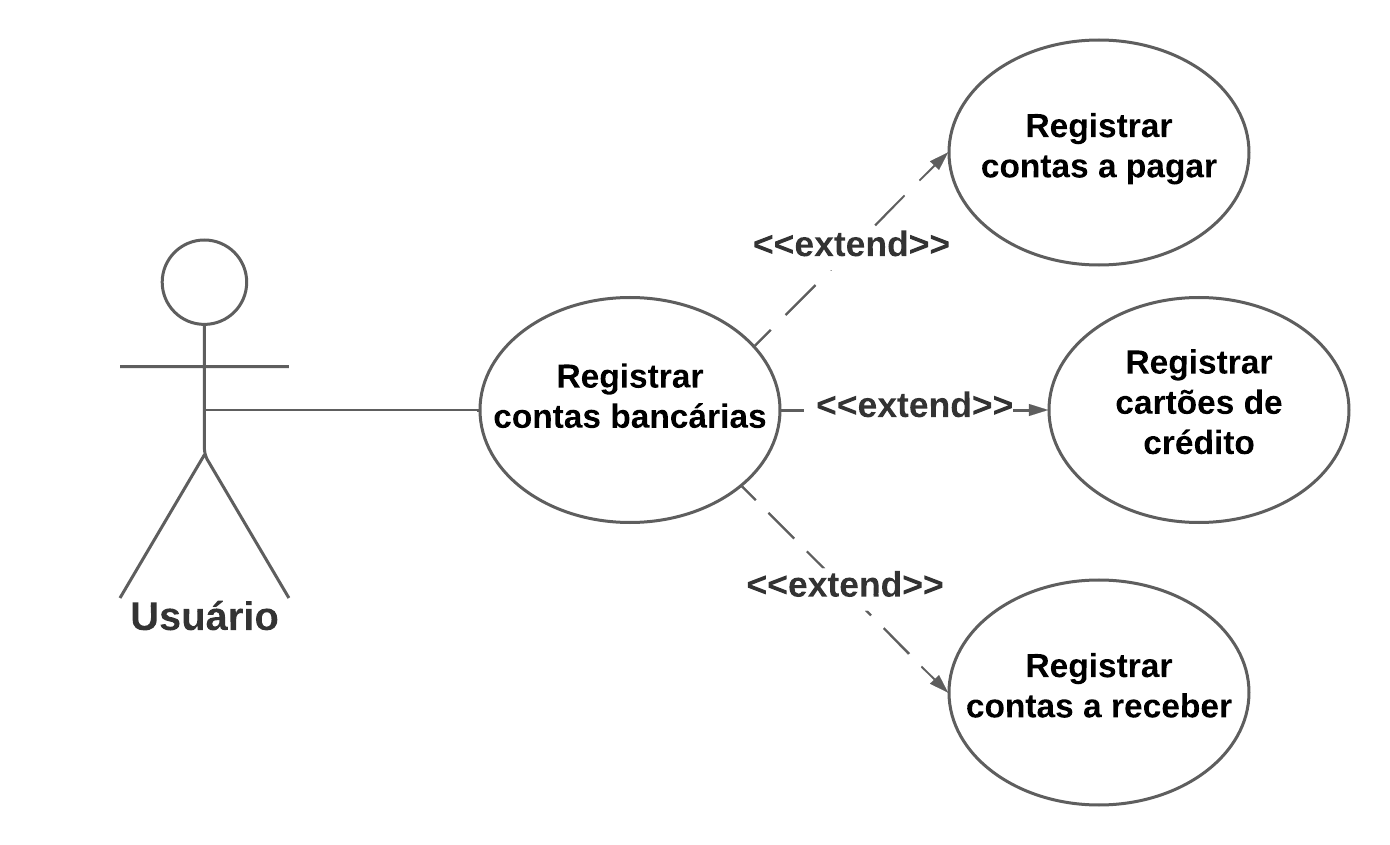


Figura 3 – Caso de Uso – Registrar contas bancárias

Fonte: Autoral (2023)

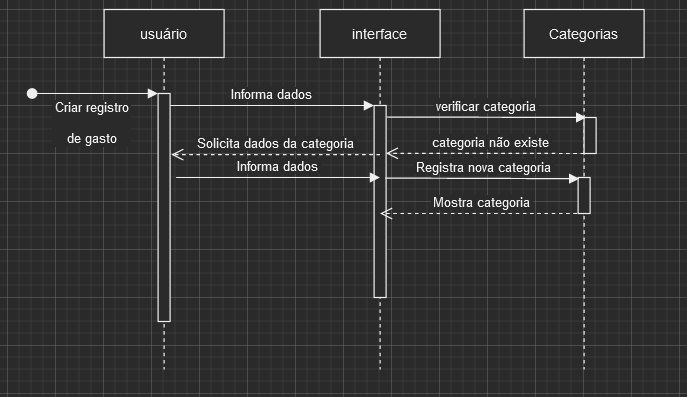
Fluxo Normal

1. O usuário registra uma conta bancária com um saldo inicial;
2. O usuário registra as contas a pagar;
3. O usuário registra os cartões de crédito;
4. O usuário registra as contas a receber;

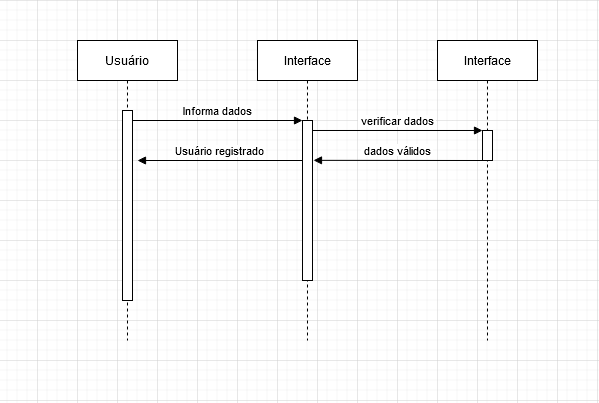
Fluxo de Exceção

1. O usuário não registra nenhuma conta bancária;
2. O usuário não pode registrar mais nenhum dado sem informa uma conta bancária;

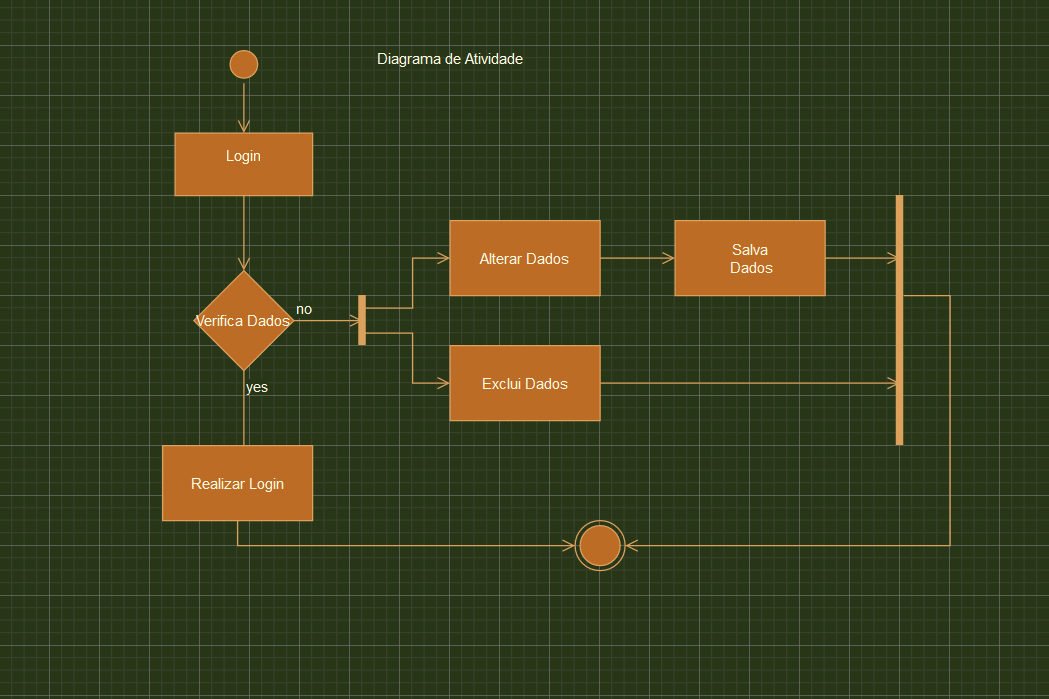
**8.0 – Diagrama de sequência**



**8.1 – Diagrama de sequência 2**



**7.0 – DIAGRAMA DE ATIVIDADE**

****

**CONCLUSÃO**

Conclui-se então que, devida a grande falta de incentivo ao gerenciamento de finanças pessoais no Brasil, a necessidade de soluções assim como a apresentada é de grande importância tanto para quem desenvolve, quanto para usuários, pois pode facilitar e tornar mais prático o gerenciamento financeiro pessoal.

Também se conclui que as ferramentas utilizadas no desenvolvimento do sistema são de grande utilidade, pois elas permitem que o sistema se torne algo escalável, ou seja, que porá ser expandido futuramente sem grandes complicações.

**TRABALHOS FUTUROS**

A principal ideia quando se trata do sistema é a expansão do mesmo, visando que novas funcionalidades podem facilitar ainda mais a utilização e a interação com usuário, fazendo com que a gestão financeira pessoal se torne algo mais agradável. Também há a ideia de expansão da aplicação para aplicativos móveis, pois mesmo com a facilidade que o sistema já vai possuir em questão de adaptação para diversos formatos de tela, também têm-se consciência de que uma aplicação mobile pode melhorar ainda mais a experiência do usuário.

**REFERÊNCIAS**

64 vantagens do Node.js que provam sua grandiosidade, 06 abr. 2022. Disponível em < <https://webtribunal.net/blog/node-js-stats/#gref> >. Acesso em 20 ago. 2022.

Documentação oficial do Typescript, Typescript para programadores Javascript. Disponível em:< [https://www.typescriptlang.org/pt/docs/handbook/typescript-in-5-minutes.html](https://www.typescriptlang.org/pt/docs/handbook/typescript-in-5-minutes.html%20) >. Acesso em 13 fev. 2023.

MongoDB, Disponível em:< <https://www.mongodb.com/pt-br> >. Acesso em 20out. 2022. ***NoSQL no desenvolvimento de aplicações Web colaborativa -*** Bernadette Farias Lóscio, Hélio Rodrigues de Oliveira, Jonas César de Sousa Pontes. Disponível em: < <https://www.addlabs.uff.br/sbsc_site/SBSC2011_NoSQL.pdf> >. Acesso mar. 2022.

Ademir José Ferreira Júnior, ***COMPARAÇÃO DOS PRINCIPAIS FRAMEWORKS JAVASCRIPT PARA DESENVOLVIMENTO WEB.*** Disponível em < <https://repositorio.animaeducacao.com.br/bitstream/ANIMA/19381/1/Artigo.pdf> >. Acesso em mar. 2022.

Hamilton Ferrari, ***Brasil tem a 4º menor inflação do G20 em 2022.*** Disponível em < https://www.poder360.com.br/economia/brasil-tem-a-4a-menor-inflacao-do-g20-em-2022/ >. Acesso em fev. 2023.

IBGE, ***Desemprego.*** Disponível em < https://www.ibge.gov.br/explica/desemprego.php >. Acesso em fev. 2023.

Sem autor, ***Comparação com outros frameworks***. Disponível em < <https://br.vuejs.org/v2/guide/comparison.html> >. Acesso em mar. 2022.

Sem autor, ***Introdução – Vue.js***. Disponível em < <https://br.vuejs.org/v2/guide/index.html> >. Acesso em mar. 2022.

Branko K, ***64 Node JS stats that proves its awesomeness in 2022***. Disponível em: < <https://webtribunal.net/blog/node-js-stats/#gref> > . Acesso em mar. 2022.