## Centro Universitário - UNIESP Sistemas para internet/informação Iarlon Oliveira

## Feliphe Blatt Fellype Gabriel Fernando Batista

## Ryan Carrazoni Rafael Oliveira

## Carlos Felipe

**Gustavo Carvalho**

**DOCUMENTAÇÃO DE REQUISITOS**

## João Pessoa 2025

## Controle Financeiro

Projeto apresentado ao curso Sistemas para internet e Sistemas de informação, na Disciplina Projeto extensionista 2, para obtenção da segunda nota, sob orientação dos professores Demetrius de Castro do Amaral.

## João Pessoa 2025

**SUMÁRIO**

1. [**Introdução**](#_nzbj0ec3equy) **4**
2. [**Escopo.**](#_gvwfv8w8r39) **4**
3. [**Requisitos funcionais**](#_jjh7my56sqga) **5**
4. [**Priorização dos requisitos.**](#_ckcy56k0lmsz) **7**
5. [**Diagrama de caso de uso.**](#_12zb49r22fp9) **7**
6. [**Restrições e limitações**](#_k78q6lofkjmw) **10**
7. [**Requisitos não funcionais**](#_iin848ionblh) **11**
8. [**Conclusão**](#_u7urqskaa9mk) **12**
9. [**Bibliografia**](#_hwfxfd5bgfu6) **12**

# INTRODUÇÃO

Atualmente, muitos jovens enfrentam dificuldades para gerenciar suas finanças pessoais de forma eficiente. A falta de planejamento pode levar a problemas como gastos impulsivos, dificuldade para pagar contas em dia e desafios para alcançar objetivos financeiros. Além disso, a ausência de uma visão clara sobre receitas, despesas e saldo disponível pode gerar insegurança e estresse. Com isso, este projeto propõe o desenvolvimento de um sistema web intuitivo e acessível, voltado para o público jovem, que busca melhorar sua organização financeira. A plataforma permitirá o acompanhamento de entradas e saídas de dinheiro, planejamento de metas e controle de contas a pagar, ajudando os usuários a tomarem decisões financeiras mais conscientes. Com isso, espera-se que os jovens consigam estabelecer um equilíbrio financeiro, evitar endividamento desnecessário e alcançar seus objetivos com mais facilidade.

**2. Escopo**

O projeto contempla o desenvolvimento de um sistema completo de controle financeiro pessoal, envolvendo as seguintes tecnologias e ferramentas:

**Back-end**

O back-end será desenvolvido em Java, utilizando o Spring Framework. A escolha do Spring se deve à sua robustez, escalabilidade e amplo suporte para aplicações corporativas. Dentre os principais recursos utilizados, destacam-se:

* Injeção de dependência
* ORM (Object-Relational Mapping) com JPA (Java Persistence API)
* Segurança (Spring Security)
* Facilidade para criação de APIs RESTful

**Banco de Dados**

O banco de dados utilizado será o PostgreSQL, escolhido pelo conhecimento prévio da equipe e pela sua adequação às necessidades do projeto, oferecendo confiabilidade e desempenho.

**Front-end**

O front-end será desenvolvido utilizando JavaScript, HTML e CSS, proporcionando uma interface responsiva e de fácil usabilidade.

**Ferramentas de Teste e Monitoramento:**

* Insomnia: Utilizado para testar as APIs RESTful do back-end, permitindo a verificação de endpoints, requisições e respostas.
* DBeaver: Utilizado para monitorar e interagir com o banco de dados PostgreSQL, facilitando a análise e validação dos dados.

# REQUISITOS FUNCIONAIS

Tabela de Requisitos Funcionais e prioridades

| **Código** | **Requisito Funcional** | **Prioridade** | **Justificativa/Observações** |
| --- | --- | --- | --- |
| **RF01** | **Cadastro de usuários** | **Essencial** | **Sem cadastro, não há controle individual de dados e nem autenticação; tela cadastro.html, autenticado via backend.** |
| **RF02** | **Login/autenticação de usuários** | **Essencial** | **Necessário para garantir segurança e individualização das informações financeiras; tela login.html, controle JWT.** |
| **RF03** | **Registro de receitas e despesas** | **Essencial** | **Função central do sistema; sem ela, não há gerenciamento financeiro; tela transacoes.html, integração REST.** |
| **RF04** | **Listagem e filtro de transações** | **Essencial** | **Usuário precisa visualizar e analisar suas movimentações; tela transacoes.html, filtros por tipo, categoria, data.** |
| **RF06** | **Categorização das transações por tipo (fixa, variável, extra)** | **Importante** | **Facilita análise financeira, mas o sistema funciona sem categorização detalhada; implementação backend + frontend.** |
| **RF05** | **Exportação de transações (CSV)** | **Importante** | **Cliente geralmente quer exportar dados, mas não é obrigatório; opção de exportação em transacoes.html.** |
| **RF07** | **Visualização de dashboard com gráficos e saldo** | **Importante** | **Ajuda na tomada de decisão, mas o sistema pode funcionar apenas com listagem textual; index.html + dashboard.js + Chart.js.** |
| **RF08** | **Geração de extratos/relatórios** | **Importante** | **Similar ao dashboard, agrega valor mas não é obrigatório para operar; exportação disponível (CSV).** |
| **RF09** | **Configuração de preferências do usuário** | **Desejável** | **Personalização não é essencial para o uso básico do sistema; tela config.html, script config.js.** |
| **RF10** | **Interface responsiva** | **Desejável** | **Melhora usabilidade em múltiplos dispositivos, mas não impede funcionamento em desktop; Bootstrap 5 e CSS.** |
| **RF11** | **Alternância de tema claro/escuro** | **Desejável** | **Apenas incrementa a experiência do usuário, não é essencial; script tema.js.** |

## Legenda das Prioridades:

Essencial: São aqueles requisitos que são obrigatórios para o funcionamento do projeto.

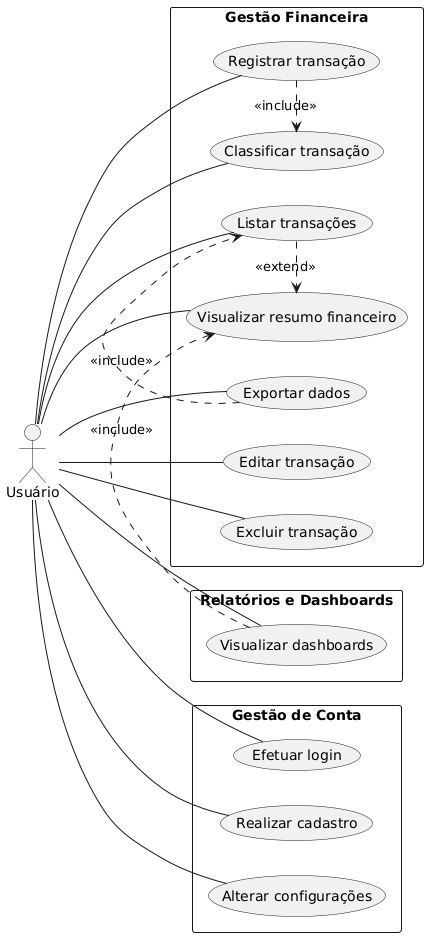
Importante: São os requisitos que o cliente listou como prioritários para o seu projeto.

Desejável: São os requisitos que podem ser deixados para uma segunda implementação, não sendo necessários para o funcionamento do projeto e nem para

o trabalho.

# DIAGRAMA DE CASO DE USO

* 1. Diagrama de caso de uso



**4.2 Tabela de Casos de Uso – Sistema de Controle Financeiro**

| **Caso de Uso** | **Ator** | **Descrição** | **Fluxo Principal** | **Pré-condições** | **Pós-condições** |
| --- | --- | --- | --- | --- | --- |
| Efetuar login | Usuário | Usuário autenticar-se no sistema | Usuário informa credenciais; sistema valida e concede acesso | Usuário cadastrado | Usuário autenticado |
| Realizar cadastro | Usuário | Novo usuário se cadastra no sistema | Usuário preenche dados, sistema valida e cria conta | - | Usuário cadastrado |
| Registrar transação | Usuário | Cadastrar receita ou despesa | Usuário informa dados da transação, categoria, valor, etc.; sistema registra | Usuário autenticado | Transação registrada |
| Editar transação | Usuário | Alterar dados de uma transação existente | Usuário seleciona transação, altera dados e salva | Usuário autenticado | Transação atualizada |
| Excluir transação | Usuário | Remover uma transação do sistema | Usuário seleciona transação e confirma exclusão | Usuário autenticado | Transação removida |
| Listar transações | Usuário | Visualizar e filtrar transações | Usuário acessa listagem; sistema exibe transações cadastradas | Usuário autenticado | Transações exibidas |
| Categorização de transação | Usuário | Classificar transação como fixa, variável ou extra | Ao registrar/editar, usuário escolhe tipo e categoria | Usuário autenticado | Transação categorizada |
| Visualizar dashboards | Usuário | Ver gráficos e resumo financeiro | Usuário acessa dashboard; sistema apresenta gráficos, saldos e indicadores | Usuário autenticado | Dados financeiros exibidos |
| Exportar dados | Usuário | Exportar transações em arquivo (ex: CSV) | Usuário solicita exportação; sistema gera e disponibiliza arquivo | Usuário autenticado | Arquivo gerado |
| Alterar configurações | Usuário | Alterar preferências da conta | Usuário acessa configurações, altera dados e salva | Usuário autenticado | Preferências atualizadas |

# 

# 4 RESTRIÇÕES E LIMITAÇÕES

# Uso obrigatório de receitas e despesas: O sistema exige que o usuário insira receitas e despesas para que seja possível gerar relatórios e visualizar o dashboard. Sem essas informações, as funcionalidades de análise financeira não funcionam.

# Cadastro e login obrigatórios: Para acessar e utilizar qualquer funcionalidade da aplicação, é necessário realizar cadastro e login. O acesso é restrito a usuários autenticados.

# Sem integração automática com bancos: O sistema não realiza integração automática com instituições financeiras. Todos os lançamentos (receitas e despesas) devem ser feitos manualmente ou por importação de arquivos.

# Importação de arquivos limitada a formatos específicos: Apenas arquivos no formato CSV são aceitos na importação de dados financeiros. Outros formatos, como OFX, não são suportados.

# Alterações e remoções de dados: Após a exclusão de uma receita ou despesa, não é possível recuperá-la através do sistema. Não existe funcionalidade de lixeira ou restauração de transações.

# Sem funcionalidade offline: A aplicação depende de uma conexão ativa com a internet para autenticação, consulta, cadastro e visualização de dados. Não há suporte para uso offline.

# Opções de categoria limitadas: O usuário pode escolher apenas entre categorias pré-definidas, que são:

# Para receitas: SALARIO, BONUS, FREELANCER, VENDA, RENDIMENTO, INVESTIMENTO, OUTROS

# Para despesas: LAZER, EDUCACAO, MORADIA, TRANSPORTE, ALIMENTACAO, SAUDE, PRESENTES, PET, INVESTIMENTOS, ASSINATURAS, OUTROS

# Não é possível criar novas categorias personalizadas.

# REQUISITOS NÃO FUNCIONAIS

| **Código** | **Requisito Não-Funcional** | **Prioridade** | **Justificativa/Observações** |
| --- | --- | --- | --- |
| **RNF01** | **Autenticação segura via JWT** | **Essencial** | **Fundamental para segurança e individualização dos dados; protege acesso às informações financeiras.** |
| **RNF02** | **Criptografia de senha (BCrypt)** | **Essencial** | **Garante que as senhas dos usuários não sejam armazenadas em texto puro; requisito de segurança básico.** |
| **RNF03** | **Validação de dados de entrada** | **Essencial** | **Evita inconsistências e falhas; protege o sistema contra entradas inválidas ou maliciosas.** |
| **RNF04** | **Proteção contra CSRF e configuração CORS** | **Essencial** | **Previne ataques comuns a sistemas web e garante que apenas origens confiáveis possam acessar a API.** |
| **RNF07** | **Utilização de banco de dados PostgreSQL** | **Essencial** | **Banco relacional robusto, fundamental para a persistência e integridade dos dados financeiros.** |
| **RNF10** | **Arquitetura em camadas (Controller, Service, Repository, Model, DTO, etc.)** | **Essencial** | **Melhora organização, manutenção e escalabilidade do sistema; separação clara de responsabilidades.** |
| **RNF05** | **Interface responsiva** | **Importante** | **Garante melhor experiência de uso em dispositivos móveis e diferentes resoluções, mas não impede o uso.** |
| **RNF06** | **Estrutura modular de frontend (HTML + JS por tela)** | **Importante** | **Facilita manutenção e evolução do frontend, mas o funcionamento básico não depende disso.** |
| **RNF08** | **Testes automatizados e coleções para Insomnia/Postman** | **Importante** | **Melhora a qualidade e facilita a validação das funcionalidades, mas não é obrigatório para rodar o sistema.** |
| **RNF09** | **Documentação da API** | **Importante** | **Facilita integração, manutenção e uso do sistema por terceiros ou pela equipe, mas não é essencial.** |
| **RNF11** | **Permitir execução via Docker** | **Desejável** | **Facilita o deploy e a replicação do ambiente, mas o sistema pode ser executado sem containerização.** |

# CONCLUSÃO

O desenvolvimento do Web App para a gestão financeira se mostrou uma solução eficaz para a organização e controle financeiro pessoal. Ao longo do processo, foi possível identificar as necessidades dos usuários, como a simplicidade na categorização de despesas, a visualização clara dos gastos ao longo do tempo e a criação de relatórios detalhados que permitem uma análise mais assertiva dos hábitos financeiros. A estrutura da aplicação está planejada para seguir uma estética limpa, simples e intuitiva, para que usuários com diferentes níveis de experiência possam utilizá-la com facilidade. Além disso, a implementação de funcionalidades como a possibilidade de inserir despesas recorrentes, alertas para metas de orçamento e gráficos interativos contribui para o engajamento contínuo dos usuários.

# BIBLIOGRAFIA

<https://developers.google.com/chart/interactive/docs/gallery?hl=pt-br>;

[https://www.indeed.com/career-advice/career-development/types-of-graphs-and-](https://www.indeed.com/career-advice/career-development/types-of-graphs-and-charts#%3A~%3Atext%3DWhile%20many%20people%20use%20%27graph%2Cand%20show%20trends%20over%20time) [charts#:~:text=While%20many%20people%20use%20%27graph,and%20show%20tr](https://www.indeed.com/career-advice/career-development/types-of-graphs-and-charts#%3A~%3Atext%3DWhile%20many%20people%20use%20%27graph%2Cand%20show%20trends%20over%20time) [ends%20over%20time](https://www.indeed.com/career-advice/career-development/types-of-graphs-and-charts#%3A~%3Atext%3DWhile%20many%20people%20use%20%27graph%2Cand%20show%20trends%20over%20time);