Antonio Wictor Pereira de Sousa

Diogo Di Torres Alexandre

Gisele Gomes Costa

Nivea Hayane Gomes Miranda

Rafael Alves Rodrigues

Victor Manoel Pinheiro Coutinho

Documento de Visão e Escopo Projeto Integrador I

Projeto e modelagem de um sistema de controle financeiro pessoal: Poupaê

Crateús, Ceará
2025

SUMÁRIO

Análise de Requisitos – Poupaê

1. VISÃO GERAL

O surgimento da ferramenta se deu através da percepção coletiva dos alunos da Universidade Federal do Ceará, em específico das vivências dos que além de bolsistas remunerados, residem fora da sua cidade de origem. E com isso, arcam com despesas como aluguel, transporte intramunicipal, alimentação e entre outras despesas pessoais. Então, este sistema tem como objetivo permitir o controle detalhado das receitas, despesas e rendas para planejamento de metas financeiras do aluno. A aplicação oferece a geração de gráficos e resumos mensais, permitindo uma melhor compreensão da saúde financeira do usuário ao longo do tempo.

2. ATORES

Ator	Descrição
Usuário	Aluno que utilizará o sistema para registrar suas movimentações financeiras. Poderá fazer o cadastro, logar no sistema, no dashboard poderá inserir, editar e excluir receitas e despesas, além de visualizar gráficos, saldos e porcentagens.

3. REQUISITOS FUNCIONAIS

Código	Requisito Funcional	Descrição Técnica Detalhada
RF01	Autenticação do usuário	A tela de login deve permitir a entrada do nome de usuário e senha. Ao clicar em "Fazer Login", o sistema deve validar as credenciais. Se estiverem corretas, o acesso será liberado; se não, uma mensagem de erro será exibida.
RF02	Visualizar senha	O sistema deve permitir ao usuário visualizar ou ocultar o conteúdo digitado no campo de senha.
RF03	Redirecionar para cadastro	Através de um botão, o usuário poderá clicar e ser redirecionado para a tela de cadastro. O sistema deve permitir a navegação entre a tela de login e a tela de cadastro do usuário.

RF04	Cadastro de novo usuário	O sistema deve aceitar nome de usuário, e-mail, senha e confirmação da senha. Os campos devem ser validados: nenhum pode estar vazio, o e-mail deve ter formato válido e a senha deve coincidir com a confirmação.
RF05	Voltar para tela de login	A tela de cadastro possui um botão "Voltar à Login", que ao clicar ele retorna para a tela inicial de login, onde será necessário inserir as credenciais do RF01.
RF06	Inserir nova despesa	O sistema deve permitir que o usuário insira uma nova despesa informando categoria (ex.: alimentação, transporte), data (selecionável via calendário) e valor (em formato monetário). O registro será salvo em memória ou banco de dados e refletido imediatamente na tabela e gráficos.
RF07	Inserir nova receita	Similar ao RF01, mas o tipo será classificado como receita. Exemplos: bolsa, salário, presente.
RF08	Exibir tabela detalhada	Deve apresentar uma lista com todas as receitas e despesas, com colunas: categoria, data, valor e tipo (receita ou despesa).
RF09	Excluir registros	O sistema deve permitir que o usuário selecione uma linha da tabela e clique em um botão "Excluir". O dado será removido da lista e os totais e gráficos serão atualizados.
RF10	Calcular total de receitas	O sistema deve somar automaticamente os valores classificados como receita e exibir o resultado em um campo identificado (ex.: "Total de Receitas: R$ XXX,XX").
RF11	Calcular total de despesas	Similar ao RF05, mas considerando os valores classificados como despesas.
RF12	Calcular e exibir o saldo	O saldo deve ser calculado como: total de receitas - total de despesas.
RF13	Calcular porcentagem da receita gasta	A porcentagem será exibida em um campo com porcentagem formatada (ex.: 75%) e possível barra de progresso visual.

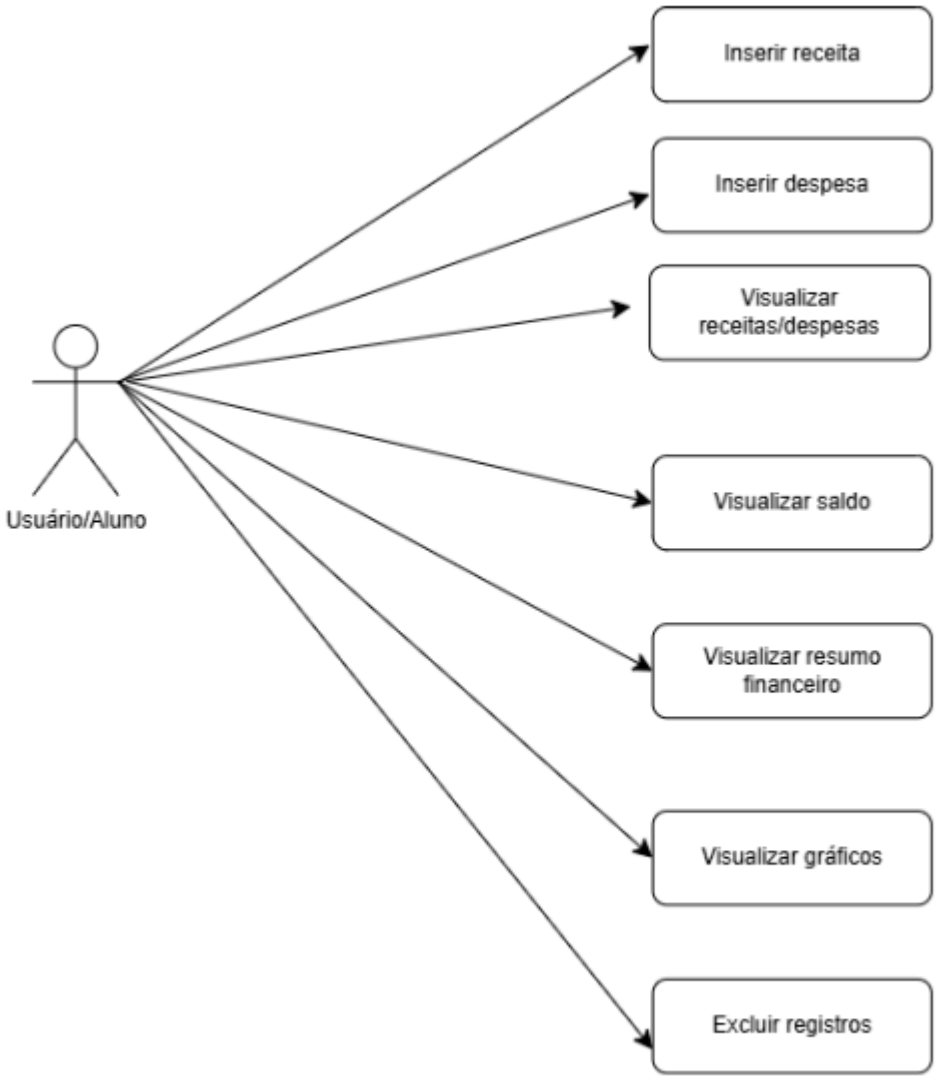
RF014	Exibir dashboard interativo	O sistema deve exibir um dashboard com gráficos visuais e dinâmicos comparando receitas, despesas e saldos mensais. Os gráficos deverão ser atualizados conforme os dados forem inseridos, editados ou excluídos.
RF015	Editar registros	Permitir que o usuário faça a edição dos registros. Após salvar, os valores e gráficos devem ser recalculados.
RF016	Filtrar por data ou categoria	Criar campos de busca para filtrar a visualização por período (mês, intervalo de datas) ou uma categoria específica (ex.: "Ver apenas transporte").
RF017	Ordenar a tabela	Criar campos de busca para filtrar a visualização por período (mês, intervalo de datas) ou uma categoria específica (ex.: "Ver apenas transporte").
RF018	Resumo mensal com alerta	Exibir mensagem: “Você gastou mais do que ganhou este mês” ou “Você tem saldo positivo”, com ícones de alerta ou sucesso.

4. REQUISITOS NÃO FUNCIONAIS

Código	**Requisito Não Funcional**	**Detalhamento**
RNF01	Validação Básica de Dados	Os campos devem aceitar apenas dados válidos, evitando erros como campos vazios.
RNF02	Usabilidade	A interface deve ser intuitiva, com ícones, cores suaves e bom contraste.
RNF03	Portabilidade	O sistema deve funcionar no Windows, no decorrer do projeto, iremos testar no Linux e no macOS.

RNF04	Desempenho	Em testes realizados em um notebook com processador Intel i5-10300H, 8GB de RAM DDR4 (3200MHz) e placa de vídeo GTX 1650 (4GB VRAM), o sistema foi executado com tempo médio de carregamento de 15 segundos, mesmo com outros aplicativos abertos. O executável principal possui cerca de 65,1 MB. O sistema também atualiza os valores e gráficos em tempo real após cada inserção ou exclusão. Observação: o aplicativo ainda está em desenvolvimento e poderá sofrer alterações, o que pode impactar no tamanho do arquivo e no tempo de espera.
RNF05	Manutenibilidade	O sistema deve ter código organizado e modular. Sugere-se que front-end e back-end se comuniquem via sockets, o que facilita a manutenção e escalabilidade.
RNF06	Confiabilidade	Os cálculos devem ser corretos e os dados mantidos com integridade ao inserir ou excluir.
RNF07	Interatividade	O sistema deve permitir que o usuário interaja com formulários e botões.

5. DIAGRAMA DE CASOS DE USO

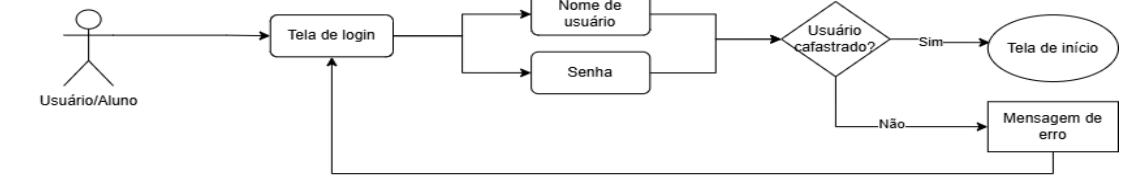


1. **Casos das principais funcionalidades do usuário:**

- Inserir receita/despesa
- Visualizar tabela
- Editar registros
- Excluir registros

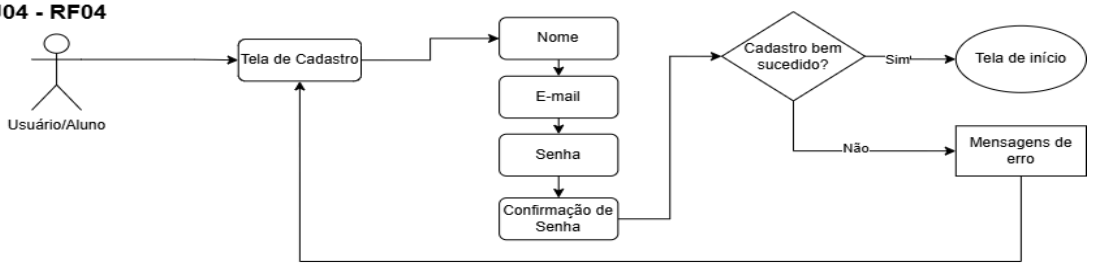
2. **Diagrama de casos de Uso dos Requisitos funcionais:**

CU01 - RF01

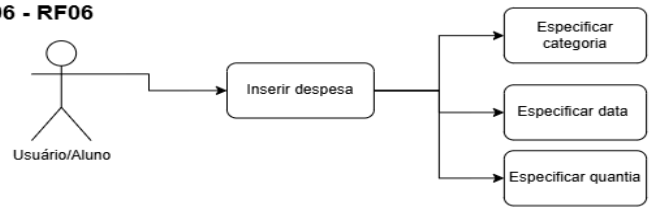


CU02 - RF03

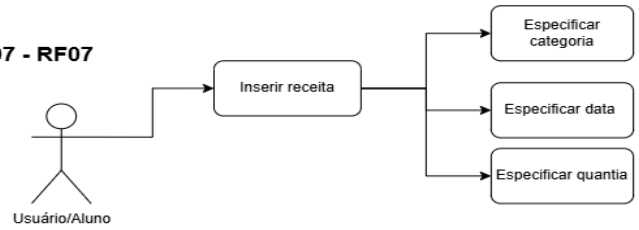
CU04 - RF04



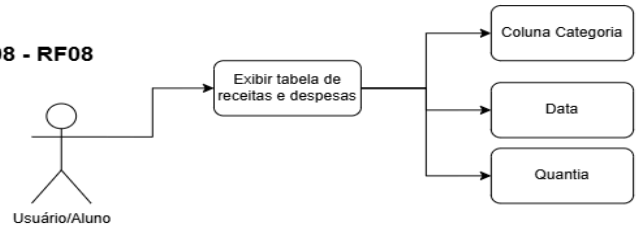
CU06 - RF06



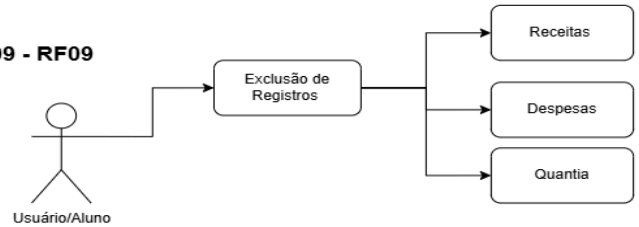
CU07 - RF07



CU08 - RF08



CU09 - RF09

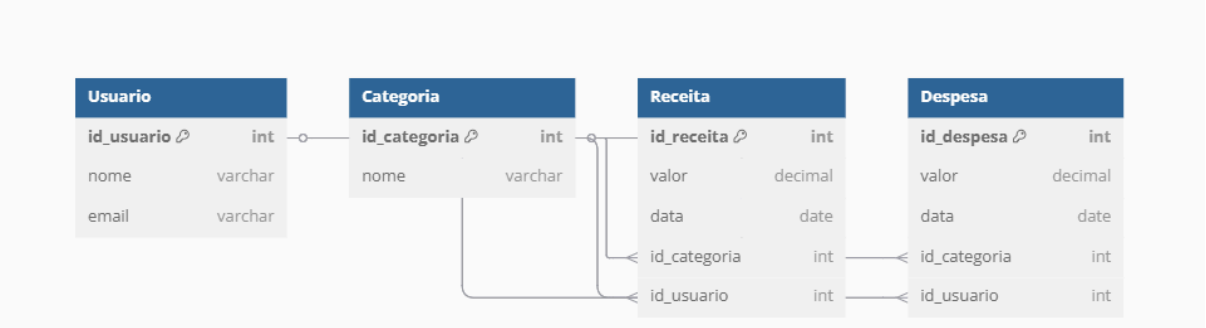


6. CASOS DE USO DETALHADOS

1. **Caso de Uso: Inserir Despesa**
    - **Ator:** Usuário
    - **Descrição:** Adiciona uma nova despesa ao sistema.
    - **Pré-condição:** O sistema está em execução.
    - **Fluxo Principal:**
        - Usuário clica em “Nova Despesa”.
        - Preenche os campos: categoria, valor, data.
        - Clica em “Adicionar”.
        - Sistema salva o registro e atualiza a tabela e os gráficos.
    - **Fluxo Alternativo:**
        - Se algum campo estiver vazio ou inválido, o sistema exibe uma mensagem de erro.
2. **Caso de Uso: Visualizar Gráficos**
    - **Ator:** Usuário
    - **Descrição:** Visualiza os gráficos de pizza e de barras.
    - **Pré-condição:** O sistema já ter dados cadastrados.
    - **Fluxo Principal:**
        1. Usuário consegue visualizar os gráficos de acordo com os dados que estão sendo adicionados no sistema
        2. Sistema gera e exibe os gráficos com base nos dados salvos.

7. MODELAGEM DE RELACIONAMENTO DAS CLASSES PRINCIPAIS

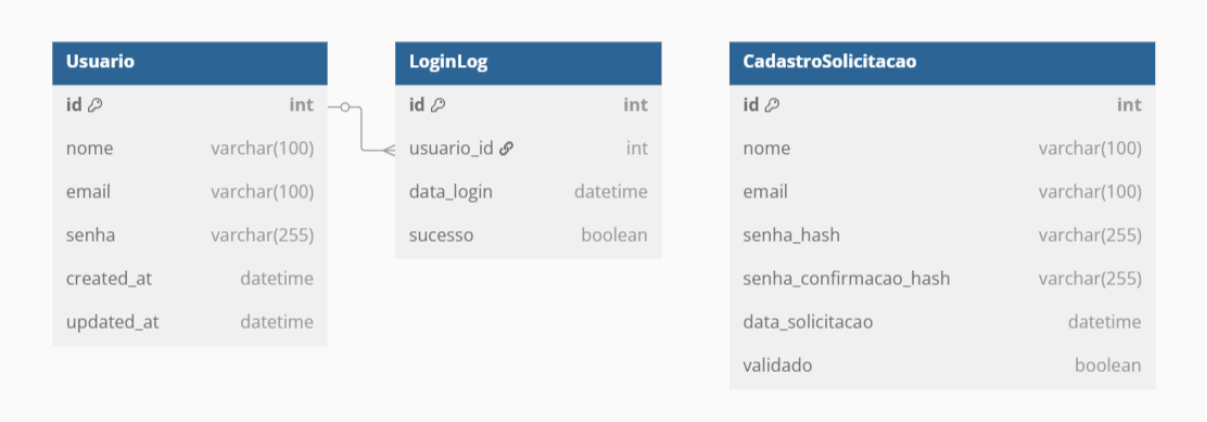
1. **Modelagem do dashboard:**



O diagrama apresenta a relação de quatro classes principais: Usuário, Categoria, Receita e Despesa.

- Usuário: representa quem vai utilizar o sistema. Cada usuário possui um id_usuario, nome e email.
    - Está relacionado com Receita e Despesa, indicando que cada lançamento pertence a um único usuário.
- Categoria: classifica os lançamentos (como “Alimentação” e “Transporte”, etc). Cada categoria possui um id_categoria e um nome.
    - Está ligada tanto à Receita quanto à Despesa, indicando que cada uma pertence a uma única categoria.
- Receita: representa os valores recebidos pelo usuário. Possui os campos id_receita, valor, data, id_categoria e id_usuario.
    - Cada receita é associada a um usuário e uma categoria.
- Despesa: Representa os valores gastos pelo usuário. Contém os campos id_despesa, valor, data, id_categoria e id_usuario.
    - Cada despesa também é associada a um usuário e uma categoria.

2. Modelagem do cadastro e login:



Nesse diagrama está representado a estrutura de dados necessária para implementar funcionalidades de login, cadastro de usuários e registro de acessos, conforme os requisitos RF01 a RF05.

Tabela: Usuario

Armazena os dados dos usuários cadastrados no sistema.
id: Identificador único (chave primária)
nome: Nome completo do usuário.
email: Endereço de e-mail único (usado para login).
senha: Senha criptografada.
created_at / updated_at: Datas para controle de criação e atualização.

Relacionamento:
Associada com LoginLog via usuario_id.

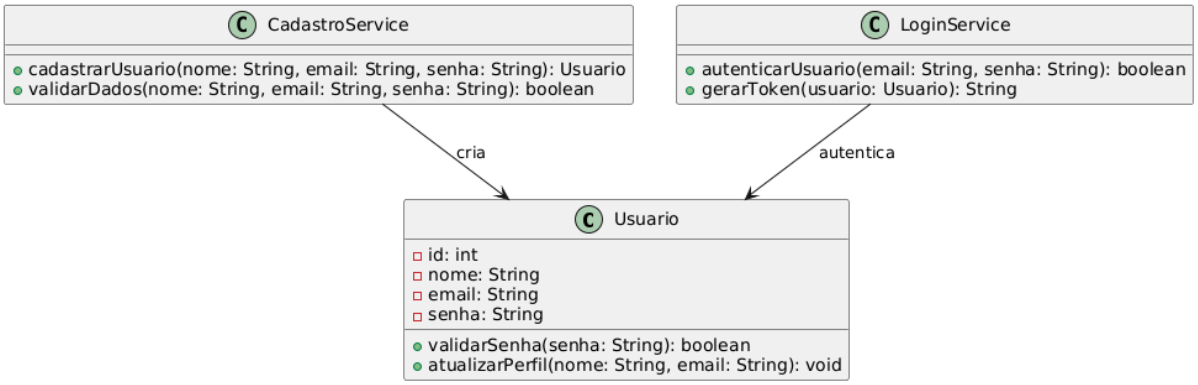
Tabela: LoginLog
Registra tentativas de login feitas no sistema.
id: Identificador da tentativa de login.
usuario_id: FK para Usuario.id, indicando quem tentou o acesso.
data_login: Data e hora da tentativa.
sucesso: Booleano indicando se o login foi bem-sucedido.

Objetivo:
Auxilia na auditoria e segurança, permitindo rastrear logins e falhas (RF01).

Tabela: CadastroSolicitacao
Opcional, usada para armazenar dados temporários de cadastro antes da criação efetiva de um usuário.
nome, email, senha_hash, senha_confirmacao_hash: Dados enviados pelo formulário de cadastro.
data_solicitacao: Registro de quando o pedido foi feito.

validado: Booleano indicando se a validação foi bem-sucedida (ex: senhas coincidem, e-mail válido etc.).
Relevância:
Suporta o processo de validação de campos e lógica de cadastro do RF04.

Relacionamentos
LoginLog.usuario_id → Usuario.id: Relacionamento de muitos para um, onde múltiplas tentativas de login podem ser atribuídas a um único usuário.



Descrição das principais classes:

Usuario: Entidade que armazena dados do usuário e ações relacionadas, como validar senha e atualizar perfil.

CadastroService: Serviço responsável pelo cadastro e validação dos dados do usuário.

LoginService: Serviço que gerencia a autenticação do usuário e geração de tokens de acesso.

8. PROTÓTIPO DE ALTA FIDELIDADE

Segue o link para o projeto no figma
https://www.figma.com/design/SNbwae5dRxa4F8iikiHnmK/Poupa%C3%AA?node-id=0-1&t=xB9uL4fsa5kVSxJZ-1