CURSO DE ENGENHARIA DE COMPUTAÇÃO

André Luis de Sousa Santos

Rafael Reis Alves

São Paulo  
2025

## 1. Introdução

O presente relatório descreve o desenvolvimento do software Gestor de Despesas, utilizando o Processo Unificado (PU) para modelagem de sistemas. O objetivo do projeto é aplicar práticas integradas de análise e desenvolvimento de software, desde a pesquisa de mercado até a documentação e apresentação final do produto.  
  
 O software tem como finalidade auxiliar o usuário a controlar suas despesas e receitas, gerar relatórios financeiros e gráficos, facilitando a tomada de decisão financeira.

## 2. Objetivo

- Criar um sistema para gestão de despesas e receitas.  
- Aplicar conceitos do Processo Unificado para modelagem de software.  
- Produzir documentação completa, incluindo diagramas e casos de uso.

## 3. Etapa 1 – Pesquisa de Nicho de Mercado

### 3.1 Escopo do Projeto

O escopo do projeto é desenvolver um software de gestão financeira pessoal, chamado Gestor de Despesas, capaz de:

* Cadastre receitas e despesas.
* Edite ou exclua registros.
* Visualizar o saldo financeiro.
* Gere gráficos de análise financeira.
* Ter um balanço e uma análise de insights sobre suas despesa
* Envia alertas de despesas recorrentes no calendário

### 3.2 Pesquisa de Nicho

Público-alvo: Estudantes, trabalhadores e pequenas empresas.  
Produtos similares: Mobills, Finance, Organizze.  
Diferenciais do projeto: Interface simples, rápido cadastro, relatórios visuais, geração automática de gráficos.

### 3.3 Requisitos do Sistema

Requisitos Funcionais (RF):

1. Cadastro de receitas e despesas.  
2. Deleção de registros.  
3. Cálculo automático do saldo.  
4. Geração de gráficos de receitas e despesas.

Requisitos Não Funcionais (RNF):

1. Interface amigável e responsiva.  
2. Armazenamento seguro e persistente dos dados.

## 4. Etapa 2 – Projeto de Software

### 4.1 Análise de Requisitos

A análise de requisitos identificou as funcionalidades essenciais, atores do sistema e fluxos de dados. O Gestor de Despesas foi estruturado para atender às necessidades de controle financeiro de forma intuitiva e segura.

### 4.2 Diagramas

#### 4.2.1 Modelo Conceitual

Entidades principais e relacionamentos:  
Gestor de despesas  
Usuário  
Interface gráfica  
Gerenciador financeiro  
Receita  
Despesa  
Banco de dados  
- Despesa  
- Relatório  
Relacionamentos:  
- Usuário cadastra Receitas e Despesas.

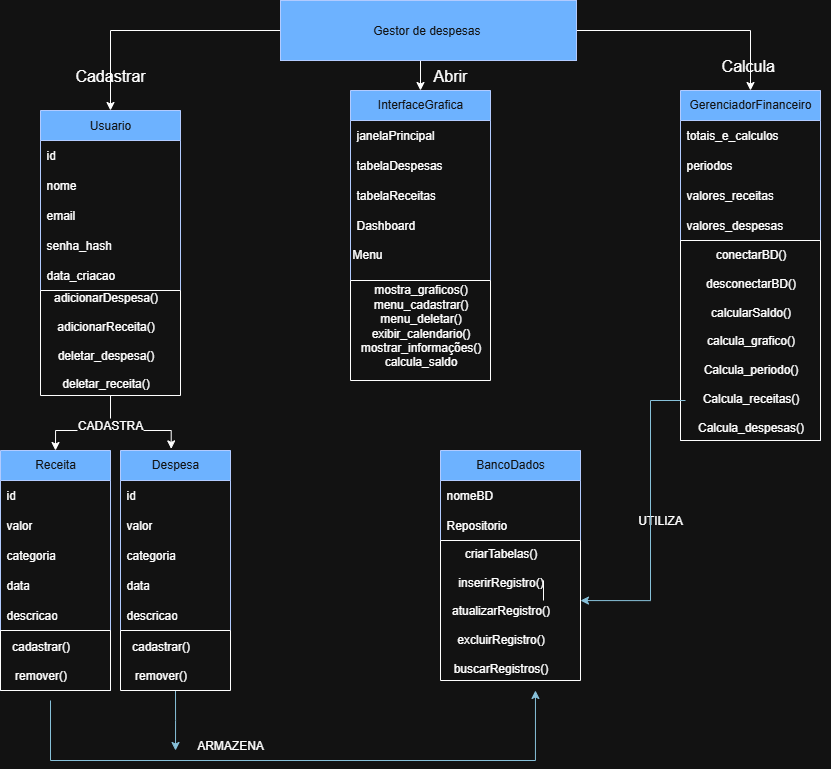


Figura 1 - Diagrama de modelo conceitual - Fonte: Autores

#### 4.2.2 Diagrama de Casos de Uso

Atores e casos de uso:  
- Usuário: Cadastrar receitas, cadastrar despesa, excluir receita, excluir despesa  
- Banco de dados: Salva receitas, salva despesas  
- Interface visual: Abra a janela do software, mostra os frames de cadastro e delete, mostra as tabelas, mostra o calendário, mostra os menus de interação  
- Dashboard: Visualizar Gráfico de despesa, visualizar gráfico de categorias, visualizar gráficos de saldo  
-Calendário: Exibir o calendário, mostrar as despesas e receitas recorrentes, mostrar alertas de despesas.

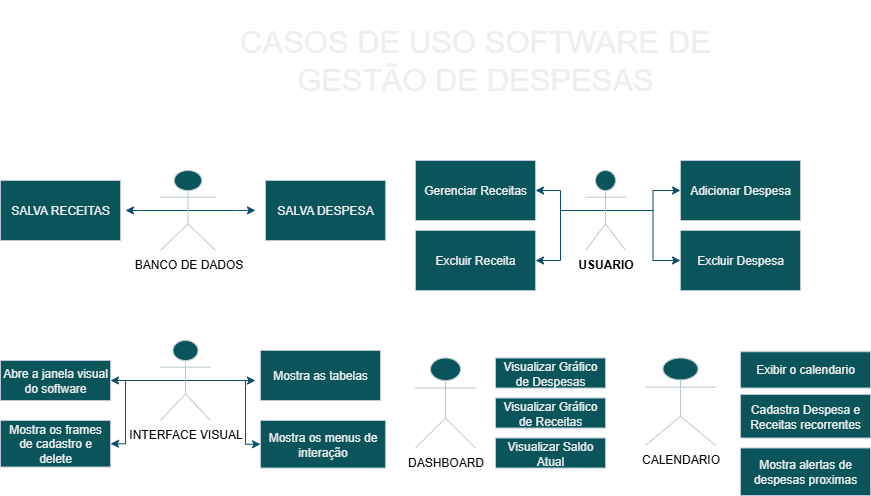


Figura 2 - Diagrama de casos de uso - Fonte: Autores

#### 4.2.3 Casos de Uso Textuais

\*\*Caso de Uso:\*\* Cadastrar Despesa  
- Ator: Usuário  
- Descrição: O usuário cadastra uma nova despesa no sistema.  
- Fluxo Principal:  
 1. Usuário acessa a tela de cadastro.  
 2. Insere os dados de despesa ou receita.  
 3. Salva os valores.  
 4. Salva no banco de dados   
 5. Retorna os valores na interface nas tabelas e nos gráficos  
 6. Calcula o saldo do balanço entre receitas e despesa e exibe no dashboard  
  
- Fluxo Alternativo:  
 - Dados incompletos → Sistema emite erro numa janela  
- Usuário não cadastrar despesa/receita → fim.   
- Se a despesa for cadastrada como recorrente → o calendário emite o alerta todo mês

#### 4.2.4 Diagrama de Classes

Classes principais e atributos/métodos:  
InterfaceGráfica  
Gerenciador  
- Janela principal   
- TreeDespesas  
- TreeReceitas  
Métodos: inicializar interface(), atualizar Treeviews()

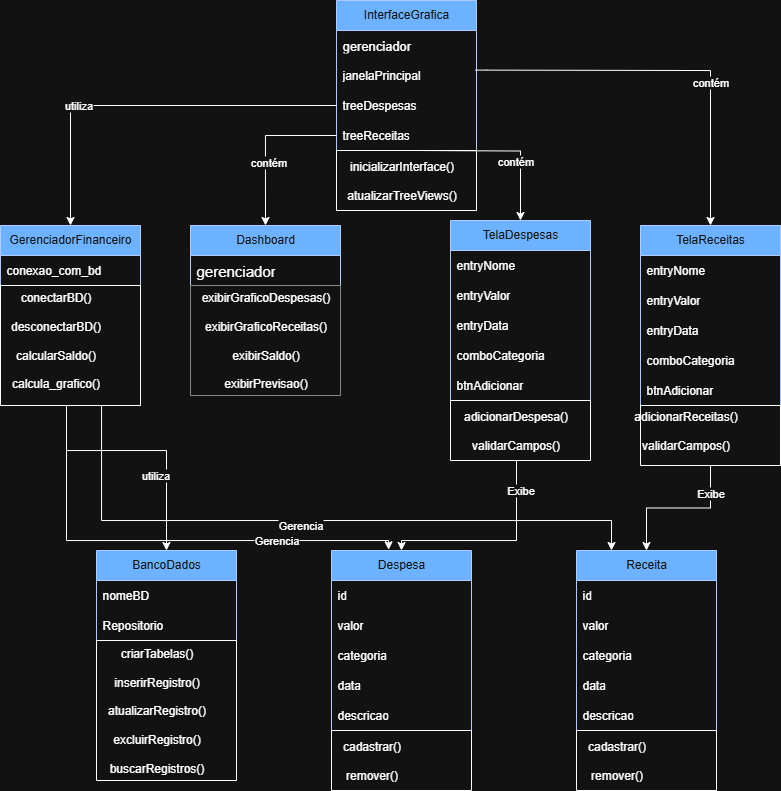


Figura 3 - Diagrama de classes - Fonte: Autores

#### 4.2.5 Diagrama de Sequência

Assim segue o fluxo:  
Usuário cadastra receita e despesas no software → Sistema salva no banco de dados → Sistema atualiza dados e tabela → Dashboard pega as informações e exibe gráficos → Calendário pega as despesas mostra as despesas recorrentes e manda alerta.

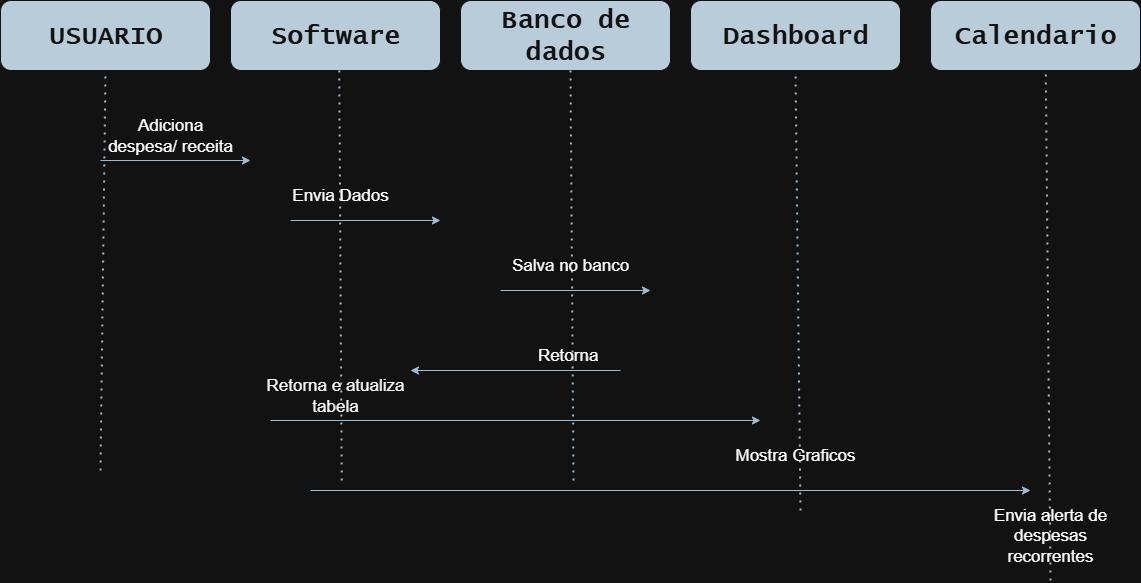


Figura 4 - Diagrama de sequência - Fonte: Autores

***4.2.6 MER - Modelo de entidade de relacionamento***Funciona da seguinte maneira os relacionamentos do software:

-Usuário Cadastra inúmeras despesas e receitas  
-Inúmeras despesas e receitas, todas elas pertencem a uma categoria  
-Inúmeras categorias são salvas no calendário para poder emitir o alerta.

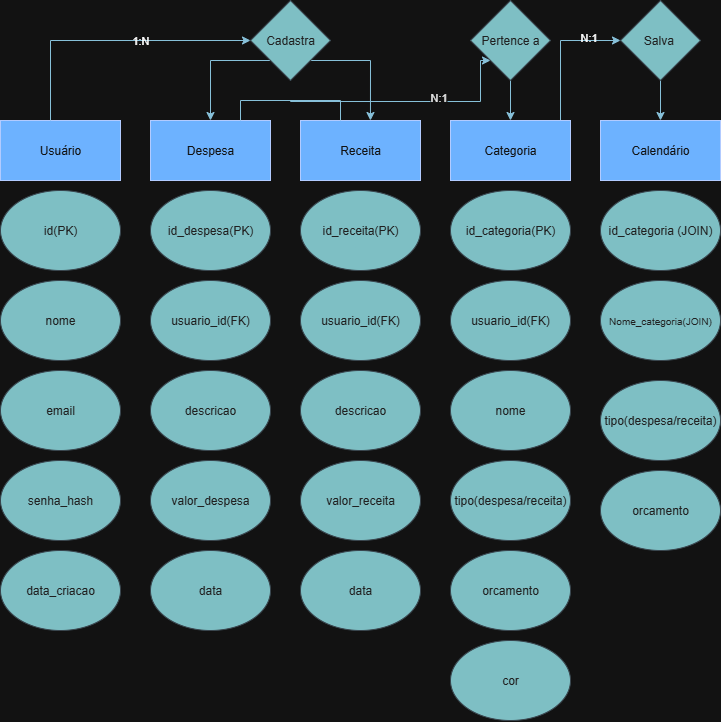


Figura 3 - Diagrama de modelo de entidade de relacionamento - Fonte: Autores

## 5. Considerações Finais

- O software atende aos requisitos funcionais e não funcionais definidos.  
- O uso do PU possibilitou uma abordagem estruturada e documentação organizada.  
- Próximos passos: Implementação da interface gráfica avançada e integração com dispositivos móveis.

## 6. Referências

- Pressman, Roger S. Engenharia de Software: Uma Abordagem Profissional.  
- Sommerville, Ian. Software Engineering.  
- Sites e tutoriais de desenvolvimento de sistemas de gestão financeira.