

Especificação Suplementar

1. Introdução

1.1 Finalidade

Este documento tem como objetivo apresentar os requisitos não funcionais, restrições de design, padrões e regras de negócio que não foram especificados diretamente nos casos de uso. Ele complementa a especificação funcional do sistema, garantindo que aspectos como usabilidade, segurança e desempenho sejam considerados no desenvolvimento.

1.2 Escopo

O sistema tem como finalidade **facilitar o gerenciamento financeiro**, permitindo que usuários cadastrem e acompanhem receitas e despesas por meio de um **dashboard interativo**. A plataforma é destinada a **pequenos empresários, autônomos e profissionais liberais**, proporcionando uma solução simples e acessível para controle de fluxo de caixa.

1.3 Definições, Acrônimos e Abreviações

- **Fluxo de caixa:** Registro de todas as movimentações financeiras de um usuário, incluindo receitas e despesas.
- **Dashboard:** Painel interativo que apresenta gráficos e indicadores financeiros.
- **CRUD:** Acrônimo para Create, Read, Update e Delete, referente às operações de cadastro, consulta, atualização e remoção de registros.

1.4 Referências

- IEEE Std 830-1998 – Recommended Practice for Software Requirements Specifications.
- Rational Unified Process (RUP) – Best Practices for Software Engineering.
- Documentação oficial do Next.js: <https://nextjs.org/docs>.
- Documentação do Prisma ORM: <https://www.prisma.io/docs>.

2. Requisitos Suplementares

2.1 Requisitos de Usabilidade

- O sistema deve possuir uma **interface responsiva**, permitindo seu uso em dispositivos móveis e desktops.
- O layout deve ser **intuitivo**, com navegação simplificada e menus bem organizados.
- O cadastro de transações deve ser realizado em **no máximo 3 etapas**, garantindo rapidez no uso.
- Os usuários devem conseguir **filtrar e visualizar despesas e receitas de forma clara**, sem a necessidade de treinamento prévio.

2.2 Requisitos de Confiabilidade

- O sistema deve garantir **disponibilidade mínima de 99,5%**, evitando períodos de inatividade que prejudiquem o usuário.
- Deve haver um **mecanismo de backup automático** do banco de dados pelo menos uma vez por dia.
- Em caso de falha na API, o sistema deve exibir uma **mensagem de erro clara** e permitir uma nova tentativa.

2.3 Requisitos de Segurança

- As senhas dos usuários devem ser armazenadas utilizando **criptografia bcrypt**.
- O sistema deve utilizar **autenticação JWT (JSON Web Token)** para proteger as rotas autenticadas.
- Apenas usuários autenticados podem cadastrar, editar e visualizar suas transações financeiras.
- Todas as requisições devem ser protegidas contra **ataques CSRF e SQL Injection**.

2.4 Requisitos de Desempenho

- O sistema deve carregar o dashboard em **menos de 2 segundos**.
- Consultas no banco de dados devem ser otimizadas para **tempo de resposta inferior a 500ms**.
- O banco de dados deve ser escalável para suportar **até 10.000 usuários simultâneos no MVP**.

2.5 Requisitos de Portabilidade

- A aplicação deve ser **compatível com os principais navegadores modernos**, incluindo Google Chrome, Mozilla Firefox e Microsoft Edge.
- O sistema deve ser responsivo e funcionar adequadamente em dispositivos com resolução mínima de **360x640 pixels**.

3. Restrições do Projeto

3.1 Restrições de Tecnologia

- O **frontend** deve ser desenvolvido utilizando **Next.js com Tailwind CSS**.
- O **backend** deve ser implementado com **Node.js (FastAPI opcionalmente como microserviço futuro)**.
- O banco de dados deve utilizar **PostgreSQL (NeonDB como provedor na nuvem)**.
- As requisições ao banco devem ser feitas por meio do **ORM Prisma**.

3.2 Restrições Legais e Regulamentares

- O sistema deve seguir a **LGPD (Lei Geral de Proteção de Dados)**, garantindo que as informações dos usuários sejam protegidas e não compartilhadas sem consentimento.
- Dados financeiros devem ser armazenados de forma segura e não devem ser acessíveis por usuários não autorizados.

4. Regras de Negócio

4.1 Regras Gerais

- O usuário **não pode cadastrar despesas com valor negativo ou igual a zero**.
- O usuário **deve informar uma categoria válida** ao cadastrar uma despesa.
- O sistema **não permite a edição de despesas cadastradas há mais de 30 dias**.
- Apenas transações do próprio usuário devem ser visíveis no dashboard.

4.2 Regras de Cadastro e Exclusão

- O sistema **deve permitir a exclusão de transações**, mas o usuário deverá confirmar a ação antes da remoção definitiva.
- Caso uma transação seja removida, o sistema deve recalcular automaticamente o saldo disponível.

5. Conclusão

Este documento complementa a especificação funcional do sistema, estabelecendo os **requisitos não funcionais, restrições técnicas e regras de negócio** necessárias para o correto funcionamento da aplicação. Com a definição desses requisitos, o projeto poderá ser implementado de forma estruturada, garantindo **usabilidade, desempenho, segurança e confiabilidade**.