

Requisitos funcionais e Não funcionais

Funcionais	Não Funcionais
Controle de Gastos Pessoais de forma interativa <ul style="list-style-type: none"> - Inserção de gastos - Inserção de ganhos - Categorização das receitas e despesas 	<ul style="list-style-type: none"> • Uso do banco de dados • Uso de cores e formas interativas
Edição e exclusão de registros financeiros e usuários	<ul style="list-style-type: none"> • CRUD em laravel • Uso do BD
Geração de relatórios mensais de receitas e despesas	<ul style="list-style-type: none"> • Por meio do back-end gerar relatórios personalizados intuitivos
Exibição de gráficos de distribuição de gastos	<ul style="list-style-type: none"> • Cálculos feitos no back-end • Gráficos interativos e coloridos
Sistema de notificações: <ul style="list-style-type: none"> - Lembretes para registrar despesas - Notificações inteligentes - Alertas sobre progresso de metas 	<ul style="list-style-type: none"> • Notificações feitas dentro do próprio app • Cálculo de metas a partir dos resultados diários, mensais e anuais
Tela de resumo financeiro	<ul style="list-style-type: none"> • Gráficos coloridos para demonstrar os valores
Cadastro e login de usuário	<ul style="list-style-type: none"> • Login e Cadastro realizado a partir da inserção de dados utilizando backend laravel • Os dados são inseridos no banco de dados e salvos
Sugestões e dicas financeiras (educação financeira)	<ul style="list-style-type: none"> • Exibição de conteúdos curtos com dicas, baseados no comportamento do usuário
Histórico financeiro completo	<ul style="list-style-type: none"> • Dados armazenados de forma organizada e acessível ao usuário
Cadastro de metas financeiras com prazo e valor objetivo	<ul style="list-style-type: none"> • Exibição de conteúdos curtos com dicas, baseados no comportamento do usuário

Responsividade	<ul style="list-style-type: none">• Interface adaptável a diferentes tamanhos de tela por meio do CSS
Privacidade e segurança	<ul style="list-style-type: none">• Dados do usuário armazenados com criptografia no BD
Interface amigável com acessibilidade	<ul style="list-style-type: none">• Cores, fontes e elementos acessíveis no React
Baixo consumo de bateria e dados	<ul style="list-style-type: none">• Otimizações no código para rodar bem em aparelhos modestos
Escalabilidade	<ul style="list-style-type: none">• Estrutura do backend preparada para futura integração com web ou outros serviços
Gamificação	<ul style="list-style-type: none">• Recompensas visuais e incentivos com base nas metas atingidas
Integração com calendário	<ul style="list-style-type: none">• Sincronização com calendário do dispositivo para lembretes e metas
Busca por transações	<ul style="list-style-type: none">• Sistema de filtro e pesquisa rápida por data, categoria e valor
Backup do projeto	<ul style="list-style-type: none">• Salvamento do projeto via github

1. Cadastro de Usuário

- **Ator:** Usuário
- **Descrição:** Permite que novos usuários se cadastrem no sistema.
- **Fluxo Principal:**
 1. Usuário acessa a tela de cadastro.
 2. Insere nome, e-mail e senha.
 3. Sistema valida os dados.
 4. Dados são salvos no banco via Laravel.
- **Pré-condições:** Não estar logado.

2. Login de Usuário

- **Ator:** Usuário
- **Descrição:** Permite ao usuário acessar sua conta.
- **Fluxo Principal:**
 1. Usuário insere e-mail e senha.
 2. Sistema autentica os dados.
 3. Acesso ao painel financeiro é liberado.
- **Pré-condições:** Conta já cadastrada.

3. Inserção de Despesa ou Receita

- **Ator:** Usuário
- **Descrição:** Adiciona um novo gasto ou ganho.
- **Fluxo Principal:**

1. Usuário acessa a tela de inserção.
 2. Preenche dados (valor, categoria, data).
 3. Sistema salva no banco.
- **Pré-condições:** Usuário logado.

4. Edição ou Exclusão de Transações

- **Ator:** Usuário
- **Descrição:** Permite alterar ou remover lançamentos.
- **Fluxo Principal:**
 1. Usuário visualiza lista de transações.
 2. Seleciona uma e edita ou exclui.
 3. Sistema atualiza banco de dados.
- **Pré-condições:** Estar logado.

5. Visualização de Relatórios

- **Ator:** Usuário
- **Descrição:** Exibe relatórios personalizados de receitas e despesas.
- **Fluxo Principal:**
 1. Usuário acessa relatórios.
 2. Sistema processa dados.
 3. Gráficos e tabelas são exibidos.
- **Pré-condições:** Ter dados registrados.

6. Sistema de Notificações

- **Ator:** Sistema
- **Descrição:** Envia lembretes, dicas e alertas.
- **Fluxo Principal:**
 1. Sistema verifica comportamento e metas.
 2. Gera notificações contextuais.
 3. Exibe dentro do app.
- **Pré-condições:** Sistema com acesso às configurações do usuário.

7. Cadastro de Metas Financeiras

- **Ator:** Usuário
- **Descrição:** Permite definir metas com valores e prazos.
- **Fluxo Principal:**
 1. Usuário preenche valor-alvo e data.
 2. Sistema registra e calcula progresso.
- **Pré-condições:** Estar logado.

8. Exibição de Dicas Financeiras

- **Ator:** Sistema
- **Descrição:** Sugere conteúdos educativos com base nos hábitos do usuário.
- **Fluxo Principal:**
 1. Sistema analisa padrão de gastos.
 2. Exibe dicas curtas na interface.

- **Pré-condições:** Ter interações anteriores salvas.

9. Histórico Financeiro

- **Ator:** Usuário
- **Descrição:** Exibe lista completa de transações.
- **Fluxo Principal:**
 1. Usuário acessa tela de histórico.
 2. Visualiza dados organizados por período ou categoria.
- **Pré-condições:** Ter registros financeiros.

10. Busca por Transações

- **Ator:** Usuário
- **Descrição:** Permite localizar transações específicas.
- **Fluxo Principal:**
 1. Usuário insere termo de busca (data, valor ou categoria).
 2. Sistema retorna resultados filtrados.
- **Pré-condições:** Estar logado.

11. Responsividade da Interface

- **Ator:** Sistema
- **Descrição:** Adapta interface a diferentes tamanhos de tela automaticamente.
- **Fluxo Principal:** Aplicado via CSS com media queries.
- **Pré-condições:** Acesso via diferentes dispositivos.

12. Segurança e Criptografia de Dados

- **Ator:** Sistema
- **Descrição:** Garante privacidade dos dados com criptografia no banco.
- **Fluxo Principal:** Senhas e dados sensíveis são criptografados ao salvar.
- **Pré-condições:** Uso de métodos seguros no Laravel (ex: bcrypt).

13. Integração com Calendário

- **Ator:** Sistema
- **Descrição:** Sincroniza lembretes com calendário do usuário.
- **Fluxo Principal:** Sistema envia eventos com base nas metas.
- **Pré-condições:** Permissão concedida pelo usuário.

14. Gamificação

- **Ator:** Sistema
- **Descrição:** Exibe recompensas visuais ao atingir metas.
- **Fluxo Principal:** Sistema compara resultados com metas.
- **Pré-condições:** Metas ativas no sistema.

15. Geração de Gráficos Interativos

- **Ator:** Sistema
- **Descrição:** Cria gráficos visuais para distribuição de gastos e metas.
- **Fluxo Principal:**
 1. Sistema processa os dados do usuário.
 2. Gera visualizações coloridas e interativas.

- **Pré-condições:** Ter dados registrados.

16. Tela de Resumo Financeiro

- **Ator:** Usuário
- **Descrição:** Visualiza um painel com balanço geral dos ganhos e gastos.
- **Fluxo Principal:**
 1. Usuário acessa o painel principal.
 2. Sistema exibe totais, metas e gráficos.
- **Pré-condições:** Ter registros e metas cadastradas.

17. Cálculo Automático de Metas

- **Ator:** Sistema
- **Descrição:** Calcula o ritmo necessário para atingir metas diárias, mensais e anuais.
- **Fluxo Principal:**
 1. Usuário define metas.
 2. Sistema faz os cálculos e atualiza automaticamente.
- **Pré-condições:** Metas registradas.

18. Backup do Projeto via GitHub

- **Ator:** Desenvolvedor
- **Descrição:** Permite salvar e versionar o código no GitHub.
- **Fluxo Principal:**
 1. Desenvolvedor realiza commit e push do código.

2. Código é armazenado no repositório remoto.
- **Pré-condições:** Acesso ao GitHub e repositório criado.

19. Otimização para Baixo Consumo de Dados e Bateria

- **Ator:** Sistema
- **Descrição:** Executa de forma eficiente, mesmo em aparelhos modestos.
- **Fluxo Principal:**
 1. Código é escrito com foco em performance.
 2. Recursos pesados são minimizados.
- **Pré-condições:** Nenhuma específica; se aplica à arquitetura geral.



