

# Sumário

- 1. Visão Geral da Tela de Recibos 1.1. Objetivo 1.2. Escopo e Valor de Negócio
- 2. Funcionalidades da Tela de Recibos 2.1. Filtros Disponíveis 2.2. Fontes de Dados 2.3. Geração de Recibo em PDF 2.4. Download Imediato do Arquivo
- 3. Componentes Frontend Dedicados 3.1. `FiltroRecibos.tsx` 3.2. `TabelaRecibos.tsx` 3.3. `BotaoGerarRecibo.tsx`
- 4. Integração com Backend (API RESTful) 4.1. Endpoints da API 4.2. Retorno Esperado da API
- 5. Bibliotecas Utilizadas para Geração de PDF 5.1. `jsPDF` 5.2. Personalização do Layout do Recibo
- 6. Experiência do Usuário (UX) e Acessibilidade (A11y) 6.1. Feedback Visual 6.2. Exibição Condicional do Botão 6.3. Navegação por Teclado 6.4. Design Responsivo
- 7. Testes Automatizados 7.1. Estratégia de Testes 7.2. Ferramentas de Teste 7.3. Cobertura de Testes
- 8. CI/CD 8.1. Inclusão no Pipeline
- 9. Proposta Negocial Aprimorada 9.1. Valor Estratégico 9.2. Benefícios Chave
- 10. Prompt Reutilizável

---

## 1. Visão Geral da Tela de Recibos

A tela de Recibos do EveryFin é uma funcionalidade crucial que permite aos usuários gerar documentos formais de comprovação de transações financeiras.

### 1.1. Objetivo

O objetivo principal é permitir a geração de recibos financeiros vinculados a lançamentos existentes, com suporte tanto para Recibos de Entrada (recebimentos) quanto para Recibos de Pagamento (despesas).

### 1.2. Escopo e Valor de Negócio

- **Recurso Essencial:** Este módulo é um recurso essencial para a prestação de contas e conformidade fiscal.
- **Documentos Legais e Formais:** Permite gerar documentos legais e formais com a identidade da empresa, reforçando o profissionalismo.
- **Versatilidade:** Pode ser utilizado para clientes, fornecedores ou parceiros.
- **Valorização do Sistema:** Contribui para a valorização do sistema e o profissionalismo nos registros financeiros.

## 2. Funcionalidades da Tela de Recibos

A tela de Recibos oferecerá um conjunto de filtros e uma funcionalidade robusta de geração de PDF.

### 2.1. Filtros Disponíveis

Os usuários poderão filtrar os lançamentos para encontrar a transação desejada para o recibo:

- **Tipo:** Seleção entre "Entrada" (recebimento) e "Pagamento" (despesa).
- **Data Inicial e Final:** Período em que a transação ocorreu.
- **Cliente/Fornecedor (autocomplete):** Permite buscar e selecionar a entidade associada à transação (cliente para entradas, fornecedor para pagamentos).

### 2.2. Fontes de Dados

Os dados para a geração dos recibos virão das tabelas financeiras existentes:

- **Recibos de Entrada:** Serão gerados a partir dos registros da tabela `Entradas`.
- **Recibos de Pagamento:** Serão gerados a partir dos registros da tabela `Saídas`.

### 2.3. Geração de Recibo em PDF

A funcionalidade principal da tela será a geração do recibo em formato PDF:

- **Dados Inclusos:** O recibo incluirá informações essenciais como nome do cliente/fornecedor, valor, descrição da transação, data da transação e CPF/CNPJ.
- **Layout Diferenciado por Tipo:** O layout do PDF será personalizado de acordo com o tipo de recibo:
  - **Entrada:** Com a frase "Recebemos de..."
  - **Pagamento:** Com a frase "Pagamos a..."

### 2.4. Download Imediato do Arquivo

Após a geração, o arquivo PDF será imediatamente baixado para o dispositivo do usuário, proporcionando uma experiência eficiente.

## 3. Componentes Frontend Dedicados

O frontend contará com componentes React específicos para a tela de Recibos, garantindo modularidade e reutilização.

### 3.1. `FiltroRecibos.tsx`

- Componente responsável pela UI e lógica dos filtros (tipo, datas, cliente/fornecedor).

### 3.2. TabelaRecibos.tsx

- Exibirá a listagem de lançamentos (entradas ou saídas) que correspondem aos filtros aplicados, permitindo que o usuário selecione qual transação irá gerar o recibo.

### 3.3. BotaoGerarRecibo.tsx

- O componente do botão que acionará a lógica de geração do PDF.
- Sua exibição poderá ser condicional, aparecendo apenas quando uma transação válida para recibo for selecionada.

## 4. Integração com Backend (API RESTful)

A tela de Recibos dependerá da API do backend para buscar os dados dos lançamentos financeiros.

### 4.1. Endpoints da API

- **GET /entradas?filtros:** Para buscar dados de entradas com base nos filtros.
- **GET /saidas?filtros:** Para buscar dados de saídas com base nos filtros.
- Pode ser considerado um endpoint unificado como **GET /lancamentos?tipo=entrada|saida&filtros** para simplificar a busca de transações para recibos.

### 4.2. Retorno Esperado da API

- O backend retornará um JSON contendo os dados das transações filtradas, incluindo todas as informações necessárias para preencher o recibo (descrição, valor, data, dados do cliente/fornecedor, etc.).

## 5. Bibliotecas Utilizadas para Geração de PDF

A geração do PDF será realizada no frontend para maior agilidade e menor carga no servidor.

### 5.1. jsPDF para Geração de PDF

- **jsPDF:** Será a biblioteca utilizada para a criação programática dos documentos PDF diretamente no navegador do usuário.

### 5.2. Personalização do Layout do Recibo

- A lógica de geração do PDF, utilizando **jsPDF**, será implementada para criar layouts diferenciados conforme o tipo de recibo (Entrada ou Pagamento).
- Será possível incluir logomarca da empresa, informações de cabeçalho e rodapé, e garantir a formatação correta dos valores monetários e datas.

## 6. Experiência do Usuário (UX) e Acessibilidade (A11y)

A tela de Recibos será projetada para ser intuitiva, eficiente e acessível a todos os usuários.

### 6.1. Feedback Visual

- O sistema fornecerá feedback claro ao usuário após a geração do recibo, como uma mensagem de sucesso.
- Indicadores de carregamento (ex: spinner no botão) serão exibidos durante o processo de geração do PDF.

### 6.2. Exibição Condicional do Botão

- O botão de geração de recibo (`BotaoGerarRecibo.tsx`) será exibido de forma condicional, apenas quando uma transação estiver selecionada e pronta para ter um recibo gerado.

### 6.3. Navegação por Teclado

- Todas as interações na tela (filtros, seleção de linhas da tabela, botão de gerar recibo) serão totalmente navegáveis e operáveis via teclado, com foco visual claro nos elementos.

### 6.4. Design Responsivo

- O layout da tela será responsivo, adaptando-se para funcionar bem em diferentes tamanhos de tela (desktop, tablet, mobile), garantindo a usabilidade dos filtros e da tabela.

## 7. Testes Automatizados

A qualidade e a confiabilidade da tela de Recibos serão asseguradas por uma estratégia de testes abrangente.

### 7.1. Estratégia de Testes

- **Unitários:** Testes focados nos componentes isolados (filtros, tabela, botão de geração) e nas lógicas de formatação de dados para o PDF.
- **E2E com Cypress:** Testes End-to-End para simular o fluxo completo do usuário:
  - Aplicação de filtros e seleção de um lançamento.
  - Simulação da ação de gerar o PDF e verificação do download do arquivo.
  - Mocks de arquivos para garantir que a lógica de exportação é acionada corretamente, mesmo sem a geração real de um arquivo em ambiente de teste.

## 7.2. Ferramentas de Teste

- **Jest + React Testing Library:** Para testes unitários e de integração de componentes frontend.
- **Cypress:** Para testes End-to-End.

## 7.3. Cobertura de Testes

- Será mantida uma alta cobertura de testes para as funcionalidades críticas do módulo de Recibos, garantindo a robustez do sistema.

# 8. CI/CD

O módulo de Recibos será integrado ao pipeline de CI/CD para automação de builds e deploys.

## 8.1. Inclusão no Pipeline

- O pipeline de CI/CD (GitHub Actions) incluirá etapas de linting, testes unitários, de integração e E2E, garantindo que o módulo de Recibos seja validado a cada push ou Pull Request.

# 9. Proposta Negocial Aprimorada

A implementação da tela de Recibos adiciona um valor significativo ao EveryFin, especialmente em termos de conformidade e profissionalismo.

## 9.1. Valor Estratégico

- **Prestação de Contas e Conformidade Fiscal:** O recurso é essencial para que os usuários possam gerar documentos formais para fins de prestação de contas e para cumprir requisitos fiscais.
- **Documentos Formais com Identidade da Empresa:** A capacidade de gerar recibos com logomarca e informações da empresa eleva o profissionalismo percebido da gestão financeira do usuário.

## 9.2. Benefícios Chave

- **Flexibilidade no Uso:** Os recibos podem ser gerados e utilizados para clientes, fornecedores ou parceiros, cobrindo diversas necessidades de comprovação.
- **Confiança e Credibilidade:** A disponibilidade de documentos formais aumenta a confiança e a credibilidade nas transações financeiras registradas no EveryFin.
- **Profissionalismo nos Registros:** Contribui para a organização e o profissionalismo geral dos registros financeiros do usuário.

# 10. Prompt Reutilizável

Este prompt pode ser utilizado para solicitar a implementação de telas de recibos semelhantes em outros sistemas, mantendo o padrão de excelência e detalhamento.

"Olá, [Nome do Modelo]!"

Implemente uma tela de recibos financeiros completa em um sistema de gestão, com os seguintes requisitos:

1. **Filtros Detalhados:** Inclua filtros por data (inicial e final), tipo de transação (Entrada/Pagamento) e cliente/fornecedor (com autocomplete) para buscar lançamentos.
2. **Listagem e Seleção:** Exiba uma listagem dos lançamentos filtrados (entradas ou saídas) de onde o usuário possa selecionar a transação para o recibo.
3. **Geração de PDF Personalizado:** Botão de geração de PDF utilizando `jsPDF`, com recibos personalizados por tipo (Entrada com 'Recebemos de...' e Pagamento com 'Pagamos a...'), incluindo dados da transação (nome, valor, descrição, data, CPF/CNPJ).
4. **UX e Acessibilidade:** Interface com feedback visual (após geração), exibição condicional do botão, download imediato, e total acessibilidade (navegação por teclado).
5. **Testes Automatizados:** Inclusão de testes unitários de componentes (filtros, botão) e testes E2E com Cypress (simulando filtros e geração de PDFs).
6. **CI/CD:** Garanta a inclusão do módulo no pipeline de testes e deploy contínuo.

Finalize com um documento técnico-negocial detalhado em PDF, incluindo visão geral, funcionalidades, componentes, integração, UX/A11y, testes, CI/CD e a proposta de valor.

*Documento expandido para excelência máxima e alinhamento com padrões UX e de negócios."*

Fontes

Criar Resumo em Áudio

Deep Research

Canvas

Vídeo

O Gemini pode cometer erros. Por is