

Claro! Vamos refatorar e aprimorar o documento "EveryFin - Tela de Relatórios", seguindo o padrão de detalhamento e excelência que aplicamos nos documentos anteriores.

Como de costume, o resultado será em **texto Markdown** e você precisará copiá-lo para gerar seu arquivo PDF.

EveryFin - Tela de Relatórios Financeiros (Documento Técnico e Negocial) Versão Detalhada e Aprimorada

Sumário

- 1. Visão Geral da Tela de Relatórios 1.1. Objetivo 1.2. Escopo e Valor de Negócio
- 2. Componentes da Tela de Relatórios 2.1. `FiltroRelatorios.tsx` (Componente de Filtros) 2.2. `TabelaRelatorio.tsx` (Componente de Tabela de Resultados) 2.3. `BotaoExportar.tsx` (Componente de Botão de Exportação)
- 3. Integração com API Backend 3.1. Endpoint Principal 3.2. Retorno Esperado da API 3.3. Formatação de Dados
- 4. Bibliotecas Utilizadas para Exportação e Manipulação de Dados 4.1. `jsPDF` para Geração de PDF 4.2. `SheetJS (xlsx)` para Exportação Excel 4.3. `date-fns` para Manipulação de Datas
- 5. Experiência do Usuário (UX) e Acessibilidade (A11y) 5.1. Feedback Visual Completo 5.2. Design Responsivo 5.3. Usabilidade da Tabela 5.4. Acessibilidade na Exportação
- 6. Testes Automatizados 6.1. Estratégia de Testes 6.2. Ferramentas de Teste 6.3. Cobertura de Testes
- 7. CI/CD 7.1. Validação do Build 7.2. Deploy Automático
- 8. Proposta Negocial Aprimorada 8.1. Valor Estratégico 8.2. Benefícios Chave
- 9. Prompt Reutilizável

1. Visão Geral da Tela de Relatórios

A tela de Relatórios do EveryFin é uma funcionalidade estratégica que capacita o usuário a extrair insights profundos de seus dados financeiros, oferecendo flexibilidade para personalizar e exportar informações.

1.1. Objetivo

O objetivo principal desta tela é permitir que o usuário gere relatórios financeiros com filtros personalizados, exporte os dados para formatos comuns como PDF e Excel, e acesse essas funcionalidades através de uma interface fluida e acessível.

1.2. Escopo e Valor de Negócio

- **Geração Profissional e Independente:** A capacidade de gerar relatórios de forma profissional e independente é um diferencial significativo, permitindo que o cliente audite e acompanhe suas movimentações financeiras.
- **Funcionalidade Valorizada:** Esta funcionalidade é altamente valorizada por empresas e na área de controladoria, pois fornece as ferramentas necessárias para análises aprofundadas.
- **Formatos Padronizados:** A exportação para PDF padronizado com logomarca e dados, juntamente com o Excel que oferece filtros, facilita a integração com sistemas externos e a manipulação dos dados.
- **Tomada de Decisão:** A tela de relatórios é crucial para apoiar a tomada de decisões financeiras estratégicas.

2. Componentes da Tela de Relatórios

A tela de Relatórios será construída com componentes modulares e reutilizáveis, garantindo organização e manutenibilidade.

2.1. `FiltroRelatorios.tsx` (Componente de Filtros)

- **Funcionalidade:** Este componente será responsável por coletar os critérios de filtro do usuário, incluindo datas de início e fim, e o tipo de relatório desejado (entradas, saídas, ou geral).
- **Controles:** Incluirá seletores de data intuitivos e um dropdown para o tipo de relatório. Poderá haver também um botão "Aplicar Filtros" para acionar a busca.
- **Validação:** Utilizará `zod` para validação dos campos de filtro, fornecendo feedback em tempo real ao usuário sobre entradas inválidas.

2.2. `TabelaRelatorio.tsx` (Componente de Tabela de Resultados)

- **Funcionalidade:** Exibirá os dados retornados da API em formato tabular, de forma clara e organizada. As colunas serão dinâmicas e adaptadas ao tipo de relatório selecionado.
- **Recursos da Tabela:**
 - **Ordenação:** Permitirá a ordenação dos dados clicando nos cabeçalhos das colunas.
 - **Cabeçalho Fixo:** O cabeçalho da tabela permanecerá fixo durante a rolagem para facilitar a leitura de grandes volumes de dados.
 - **Paginação:** Implementação de paginação para gerenciar grandes conjuntos de resultados e otimizar a performance.
- **Feedback Visual:** Exibirá mensagens de "carregando", "nenhum dado encontrado" ou "erro ao carregar" conforme o estado da requisição.

2.3. `BotaoExportar.tsx` (Componente de Botão de Exportação)

- **Funcionalidade:** Um botão que, ao ser clicado, apresentará opções de exportação para PDF e Excel.
- **Integração:** Acionará as lógicas de exportação que utilizam as bibliotecas `jsPDF` e `SheetJS` (`xlsx`).

- **Feedback:** Fornecerá feedback claro de operação bem-sucedida ou falha na exportação.

3. Integração com API Backend

A tela de Relatórios se comunicará com o backend para buscar e processar os dados financeiros.

3.1. Endpoint Principal

- **GET /relatorios:** Este será o endpoint principal da API para a geração de relatórios.
- **Parâmetros de Query:** Aceitará parâmetros de query para filtros, como `dataInicio=dd-mm-yyyy`, `dataFim=dd-mm-yyyy` e `tipo=entradas|saidas|geral`.

3.2. Retorno Esperado da API

- O backend retornará um JSON contendo as movimentações financeiras filtradas, prontas para serem exibidas na tabela do frontend. A estrutura do JSON será otimizada para facilitar a renderização da tabela e a exportação.

3.3. Formatação de Dados

- A API garantirá que os dados (especialmente valores e datas) sejam retornados em um formato consistente que possa ser facilmente formatado pelo frontend para exibição e exportação.

4. Bibliotecas Utilizadas para Exportação e Manipulação de Dados

Para garantir funcionalidades de exportação robustas e manipulação eficiente de dados, as seguintes bibliotecas serão empregadas:

4.1. jsPDF para Geração de PDF

- **Funcionalidade:** jsPDF será a biblioteca responsável pela geração de arquivos PDF diretamente no lado do cliente.
- **Recursos:** Permitirá a criação de PDFs padronizados com a logomarca do EveryFin, cabeçalhos, rodapés e a inclusão dinâmica dos dados do relatório.

4.2. SheetJS (xlsx) para Exportação Excel

- **Funcionalidade:** SheetJS (biblioteca xlsx) será utilizada para exportar os dados da tabela para o formato Excel (XLSX).
- **Benefícios:** A exportação em Excel com filtros facilita a análise de dados e a integração com outras ferramentas ou sistemas externos ao EveryFin.

4.3. `date-fns` para Manipulação de Datas

- **Funcionalidade:** Uma biblioteca moderna e leve para manipulação de datas no JavaScript.
- **Uso:** Será utilizada para formatar datas para exibição na UI, para preparar as datas dos filtros para o backend e para formatar datas nos arquivos de exportação.

5. Experiência do Usuário (UX) e Acessibilidade (A11y)

A usabilidade e acessibilidade da tela de relatórios serão prioridades para garantir uma experiência positiva para todos os usuários.

5.1. Feedback Visual Completo

- **Estados de Carregamento:** Será exibida uma interface de carregamento (spinner, skeleton) enquanto os dados do relatório estão sendo buscados da API.
- **Mensagens de Erro:** Em caso de falha na requisição ou no processamento, mensagens de erro claras e amigáveis serão exibidas.
- **Mensagens de Estado Vazio:** Se o relatório não retornar dados para os filtros selecionados, uma mensagem informativa como "Nenhum resultado encontrado para o período/filtros selecionados" será exibida.
- **Feedback de Operação Bem-sucedida:** Após uma exportação, uma notificação discreta (ex: toast message) confirmará que o arquivo foi gerado com sucesso.

5.2. Design Responsivo

- O layout da tela de relatórios será totalmente adaptável para diferentes tamanhos de tela (desktop, tablet, mobile), garantindo que os filtros e a tabela sejam utilizáveis em qualquer dispositivo. As colunas da tabela podem ser ajustadas ou tornarem-se roláveis horizontalmente em telas menores.

5.3. Usabilidade da Tabela

- **Tabela Ordenável:** Os cabeçalhos das colunas da tabela permitirão a ordenação dos dados em ordem crescente ou decrescente, melhorando a capacidade de análise do usuário.
- **Cabeçalho Fixo:** O cabeçalho da tabela permanecerá visível mesmo quando o usuário rolar para ver mais dados, facilitando a identificação das informações.

5.4. Acessibilidade na Exportação

- **Exportação Acessível por Teclado:** O processo de exportação (botões, seleção de formato) será totalmente navegável e acionável via teclado.
- **`aria-label` e Semântica:** Uso de atributos `aria-label` e HTML semântico para garantir que leitores de tela possam interpretar corretamente a funcionalidade dos filtros, tabelas e botões de exportação.

6. Testes Automatizados

Uma estratégia de testes robusta garantirá a qualidade e a confiabilidade da tela de Relatórios e suas funcionalidades.

6.1. Estratégia de Testes

- **Testes Unitários:** Foco em componentes isolados como `FiltroRelatorios.tsx` e `BotaoExportar.tsx`, verificando suas lógicas internas e estados.
- **Testes de Integração:** Para `TabelaRelatorio.tsx` e para a tela principal de relatórios, verificando a interação com a API (mockada) e a renderização correta dos dados e estados de carregamento/erro.
- **Testes End-to-End (E2E) com Cypress:**
 - Simulação de seleção de filtros e verificação do carregamento dos dados.
 - Testes do fluxo completo de exportação, com mocks de arquivos para verificar se a lógica de exportação é acionada corretamente.

6.2. Ferramentas de Teste

- **Jest + React Testing Library:** Para testes unitários e de integração de componentes React.
- **Cypress:** Para testes End-to-End, simulando a interação real do usuário.

6.3. Cobertura de Testes

- Uma meta de alta cobertura de testes será estabelecida para os componentes e lógicas críticas da tela de Relatórios, garantindo que as funcionalidades de filtragem e exportação sejam rigorosamente testadas.

7. CI/CD

A tela de Relatórios será integrada ao pipeline de Integração Contínua e Entrega Contínua (CI/CD) do frontend para garantir a qualidade e a automação do deployment.

7.1. Validação do Build

- O pipeline de CI/CD (GitHub Actions) garantirá que o build do frontend só seja gerado e liberado para deployment se todos os testes automatizados (unitários, de integração e E2E) passarem com sucesso, incluindo os testes específicos para as funcionalidades de exportação.

7.2. Deploy Automático

- O deploy da aplicação, incluindo a tela de Relatórios, será automatizado via Vercel (ou plataforma similar) com integração direta ao GitHub Actions, garantindo entregas rápidas e consistentes em ambientes de staging e produção.

8. Proposta Negocial Aprimorada

A implementação da tela de Relatórios representa um valor agregado significativo para o EveryFin, consolidando-o como uma ferramenta financeira completa.

8.1. Valor Estratégico

- **Análise Aprofundada:** Permite que os usuários realizem análises financeiras mais aprofundadas e personalizadas, algo essencial para o controle e planejamento.
- **Ferramenta de Auditoria e Acompanhamento:** Essencial para o cliente poder auditar suas movimentações e acompanhar seu progresso financeiro de forma independente.
- **Atração para Empresas:** A funcionalidade de relatórios é altamente valorizada em contextos empresariais e de controladoria, o que pode atrair um público mais amplo de usuários.

8.2. Benefícios Chave

- **Profissionalismo:** Geração de relatórios com um aspecto profissional, incluindo a logomarca, reforça a imagem do EveryFin.
- **Flexibilidade de Exportação:** A possibilidade de exportar para PDF e Excel oferece flexibilidade para diferentes necessidades de uso e análise.
- **Integração Facilitada:** O formato Excel com filtros facilita a integração dos dados financeiros com outros sistemas externos que o usuário possa utilizar.
- **Empoderamento do Usuário:** Coloca o poder de análise e gestão nas mãos do usuário, tornando-o mais autônomo em suas decisões financeiras.

9. Prompt Reutilizável

Este prompt pode ser utilizado para solicitar a criação de telas de relatórios financeiros semelhantes em outros contextos, mantendo o padrão de detalhamento e qualidade.

"Olá, [Nome do Modelo]!

Implemente uma tela de relatórios financeiros completa e detalhada com os seguintes requisitos:

1. **Filtros Personalizados:** Inclusão de filtros para data inicial, data final e tipo de relatório (entradas, saídas, geral).
2. **Tabela de Dados:** Exibição dos dados retornados da API em uma tabela com recursos de ordenação e cabeçalho fixo.
3. **Exportação:** Funcionalidade de exportação em PDF (utilizando `jspdf`) e Excel (utilizando `SheetJS` (`xlsx`)).
4. **Validação e Mensagens:** Validação de campos de filtro e exibição de mensagens claras de carregamento, erro ou estado vazio.
5. **Layout e Acessibilidade:** Layout acessível e responsivo para diferentes dispositivos e suporte completo à navegação por teclado.

6. **Testes Automatizados:** Inclusão de testes unitários de componentes (filtros, botões) e testes E2E (seleção de filtros, exportação simulada com mocks).
7. **CI/CD:** Integração com CI/CD, onde a exportação é testada automaticamente e o deploy é feito via Vercel com GitHub Actions.