

Projeto - WEB Controle Financeiro

Prof.: Jonatas Edward Dias de Oliveira

Objetivo

Desenvolver a interface do usuário para a aplicação de controle financeiro, que permitirá aos usuários cadastrar pessoas, grupos, metas, criar e editar lançamentos, e gerar relatórios. A aplicação será desenvolvida utilizando Angular 17 e PrimeNG.

Tecnologias

- **Framework:** Angular 17
- **Biblioteca de Componentes:** PrimeNG
- **HTTP Client:** Angular HttpClient
- **Gerenciamento de Estado:** NgRx (opcional, se necessário)
- **Estilo e Layout:** CSS e PrimeNG Themes

Requisitos

1. Cadastro de Pessoas

Funcionalidades

- **Listagem de Pessoas:** Exibir uma lista de todas as pessoas cadastradas.
- **Cadastro de Pessoas:** Permitir o cadastro de novas pessoas.
- **Edição de Pessoas:** Permitir a edição das informações de uma pessoa existente.
- **Exclusão de Pessoas:** Permitir a exclusão de uma pessoa.

Regras

- **Cadastro Único:** Cada pessoa deve ter um nome único.

- **Validação:** O nome é obrigatório e deve ser uma string não vazia.

Componentes

- `PessoaService` : Serviço para comunicação com a API.
- `PessoaListComponent` : Componente para listar e gerenciar pessoas.
- `PessoaFormComponent` : Componente para cadastro e edição de pessoas.

2. Cadastro de Grupos

Funcionalidades

- **Listagem de Grupos:** Exibir uma lista de todos os grupos cadastrados.
- **Cadastro de Grupos:** Permitir o cadastro de novos grupos.
- **Edição de Grupos:** Permitir a edição das informações de um grupo existente.
- **Exclusão de Grupos:** Permitir a exclusão de um grupo.

Regras

- **Associação com Pessoa:** Cada grupo deve estar associado a uma pessoa.
- **Validação:** O nome do grupo é obrigatório e deve ser uma string não vazia.

Componentes

- `GrupoService` : Serviço para comunicação com a API.
- `GrupoListComponent` : Componente para listar e gerenciar grupos.
- `GrupoFormComponent` : Componente para cadastro e edição de grupos.

3. Cadastro de Metas

Funcionalidades

- **Listagem de Metas:** Exibir uma lista de todas as metas cadastradas.
- **Cadastro de Metas:** Permitir o cadastro de novas metas.
- **Edição de Metas:** Permitir a edição das informações de uma meta existente.
- **Exclusão de Metas:** Permitir a exclusão de uma meta.

Regras

- **Descrição e Valor:** A meta deve ter uma descrição e um valor associado.
- **Validação:** O valor deve ser um número positivo e a descrição uma string não vazia.

Componentes

- `MetaService` : Serviço para comunicação com a API.
- `MetaListComponent` : Componente para listar e gerenciar metas.
- `MetaFormComponent` : Componente para cadastro e edição de metas.

4. Criação e Edição de Lançamentos

Funcionalidades

- **Criação de Lançamentos:** Permitir a criação de novos lançamentos financeiros.
- **Edição de Lançamentos:** Permitir a edição das informações de um lançamento existente.
- **Validação do Saldo:** Verificar se o saldo do grupo é suficiente antes de permitir a inclusão de uma despesa.

Regras

- **Associação com Grupo e Categoria:** Cada lançamento deve estar associado a um grupo e uma categoria.
- **Validação:** O valor deve ser um número e a data deve ser válida.

Componentes

- `LancamentoService` : Serviço para comunicação com a API.
- `LancamentoFormComponent` : Componente para criação e edição de lançamentos.

5. Relatórios

Funcionalidades

- **Geração de Relatórios:** Permitir a geração de relatórios mensais, por grupo e por categoria.

- **Visualização de Relatórios:** Exibir os relatórios gerados.

Regras

- **Relatórios Detalhados:** Os relatórios devem ser agrupados e filtrados conforme necessário.

Componentes

- `RelatorioService` : Serviço para comunicação com a API.
- `RelatorioListComponent` : Componente para exibir relatórios.

Estrutura do Projeto

- **src/app**
 - **components**
 - **pessoa**
 - `pessoa-list.component.ts`
 - `pessoa-list.component.html`
 - `pessoa-form.component.ts`
 - `pessoa-form.component.html`
 - **grupo**
 - `grupo-list.component.ts`
 - `grupo-list.component.html`
 - `grupo-form.component.ts`
 - `grupo-form.component.html`
 - **meta**
 - `meta-list.component.ts`
 - `meta-list.component.html`
 - `meta-form.component.ts`
 - `meta-form.component.html`
 - **lancamento**
 - `lancamento-form.component.ts`

- `lancamento-form.component.html`
- **relatorio**
 - `relatorio-list.component.ts`
 - `relatorio-list.component.html`
- **services**
 - `pessoa.service.ts`
 - `grupo.service.ts`
 - `meta.service.ts`
 - `lancamento.service.ts`
 - `relatorio.service.ts`
- **models**
 - `pessoa.model.ts`
 - `grupo.model.ts`
 - `meta.model.ts`
 - `lancamento.model.ts`
 - `relatorio.model.ts`
- `app.component.css`
- `app.component.html`
- `app.component.spec.ts`
- `app.component.ts`
- **`app.config.ts`**
- **`app.routes.ts`**

Instruções de Implementação

1. Configuração do Ambiente

- Instalar Angular CLI e PrimeNG.
- Clonar projeto base disponivel no github
<https://github.com/jonatasEdward/angular-primeng-starter>

2. Desenvolvimento dos Componentes

- Implementar os serviços para cada entidade (`Pessoa` , `Grupo` , `Meta` , `Lancamento` , `Relatorio`).
- Criar e estilizar os componentes de listagem e formulários para cada entidade.
- Implementar validações e lógicas de negócios conforme as regras especificadas.

3. Testes e Validação

- Testar cada componente e serviço individualmente.
- Realizar testes de integração para garantir a comunicação entre frontend e backend.

4. Documentação e Comentários

- Adicionar comentários detalhados nos arquivos de código para explicar a funcionalidade e a lógica implementada.
- Atualizar a documentação do projeto com instruções de uso e configuração.

Links:

<https://primeng.org/>

<https://primeng.org/installation>

<https://primeng.org/theming>