

Guia Passo a Passo - Lista de Tarefas (To-Do List) em React

1. Criando o Projeto

Para começar, crie um projeto React utilizando Vite:

```
npm create vite@latest minha-lista-tarefas --template react
cd minha-lista-tarefas
npm install
npm run dev
```

2. Estruturando os Componentes

Crie um arquivo ListaTarefas.jsx dentro da pasta src/components/ e adicione o código:

```
1 import { useState } from 'react';
2
3 function ListaTarefas() {
4   const [tarefas, setTarefas] = useState([]);
5   const [novaTarefa, setNovaTarefa] = useState('');
6
7   const adicionarTarefa = () => {
8     if (novaTarefa.trim() !== '') {
9       setTarefas([...tarefas, novaTarefa]);
10      setNovaTarefa('');
11    }
12  };
13
14  const removerTarefa = (indice) => {
15    setTarefas(tarefas.filter((_, i) => i !== indice));
16  };
17
18  return (
19    <div>
20      <h2>Lista de Tarefas</h2>
21      <input
22        type='text'
23        value={novaTarefa}
24        onChange={(e) => setNovaTarefa(e.target.value)}
25        placeholder='Digite uma nova tarefa'
26      />
27      <button onClick={adicionarTarefa}>Adicionar</button>
28      <ul>
29        {tarefas.map((tarefa, indice) => (
30          <li key={indice}>
31            {tarefa} <button onClick={() =>
32              removerTarefa(indice)}>Remover</button>
33          </li>
34        ))}
35      </ul>
36    </div>
37  );
38 }
39
40 export default ListaTarefas;
41
```

Explicação do Código:

1. **useState([])**: Gerencia o estado da lista de tarefas.
2. **adicionarTarefa**: Adiciona uma nova tarefa ao estado.
3. **removerTarefa**: Remove uma tarefa específica usando `filter()`.
4. **Renderização**: Exibe a lista de tarefas dinamicamente usando `map()`.

3. Integrando no App

No `App.jsx`, importe e utilize o componente:

```
1  import ListaTarefas from './components/ListaTarefas';
2
3  import './App.css';
4
5  function App() {
6
7    return (
8      <>
9        <h1>Gerenciador de Tarefas</h1>
10       <ListaTarefas />
11     </>
12   );
13 }
14
15 export default App;
```

4. Desafios para Melhorar o Código

1. **Estilização**: Utilize CSS para melhorar a aparência da aplicação, aplicando cores, espaçamentos e uma fonte mais amigável.
2. **Marcar tarefas como concluídas**: Adicione um botão ao lado de cada tarefa para marcar/desmarcar como concluída, alterando a estilização quando estiver concluída.
3. **Persistência dos dados**: Utilize `localStorage` para armazenar as tarefas e manter a lista mesmo após recarregar a página.
4. **Ordenação das tarefas**: Permita ao usuário reordenar as tarefas por ordem alfabética ou por data de adição.

5. Critérios de Avaliação

- **Funcionalidade (40%):** O código implementa corretamente os novos recursos?
- **Estilização (20%):** A interface está visualmente agradável e organizada?
- **Código limpo e comentado (20%):** O código está bem estruturado e explicado?
- **Criatividade e melhorias extras (20%):** Foram implementadas melhorias além das sugeridas?

6. Entrega da Atividade

- Você publicar o projeto no GitHub, garantindo que os arquivos não estejam compactados. O repositório deve conter um arquivo **README.md** com as seguintes informações:
 - ✓ Instruções detalhadas sobre como baixar ou clonar o projeto.
 - ✓ Um texto explicando o funcionamento e os objetivos do projeto.
 - ✓ Além disso, é necessário gravar um vídeo explicativo apresentando o projeto, destacando suas funcionalidades e a implementação dos conceitos aprendidos.
- Após concluir essas etapas, o link do repositório no GitHub deverá ser enviado pela plataforma **Microsoft Teams** até o dia **10/04 às 15h45**.