

Projeto: Sistema de Autenticação com React Context

PROPOSTA:

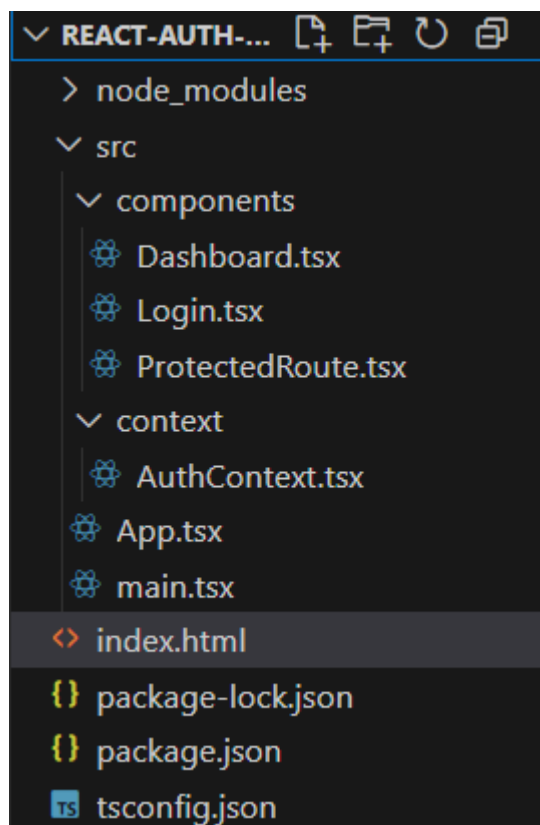
Desenvolver um sistema simples de autenticação de usuário utilizando React Context API para gerenciar o estado global de autenticação entre componentes. O sistema deve conter uma tela de login e uma área protegida que só pode ser acessada por usuários autenticados.

OBJETIVOS:

- Compreender o funcionamento do React Context.
- Aplicar Provider e Consumer ou useContext para compartilhar dados entre componentes.
- Simular a autenticação sem necessidade de backend, apenas manipulando estados locais.

Estrutura do Projeto

Sugestão para a estrutura do projeto.



Funcionalidades Esperadas

1. Login:

- Um formulário com campo de e-mail e senha.
- Ao clicar em "Entrar", o usuário será considerado autenticado (não precisa validar com backend).

2. Contexto de Autenticação:

- Criar AuthContext com os dados:
 - user: string ou null.
 - login(email, password): função para autenticar.
 - logout(): função para sair da conta.

3. Dashboard:

- Página protegida, acessível apenas se o usuário estiver autenticado.
- Deve exibir a mensagem: "Bem-vindo, {user}!" e um botão "Sair".

4. Rota Protegida:

- Criar um componente ProtectedRoute que verifica se o usuário está autenticado e redireciona para o login se não estiver.

5. Orientações para Desenvolvimento

- Criar o AuthContext com createContext e useState.
- Fornecer o contexto com AuthProvider envolvendo <App />.
- Usar useContext(AuthContext) nos componentes Login e Dashboard para acessar o estado global.
- Implementar ProtectedRoute verificando se user existe, se não, redirecionar para /login.
- Aplicar o conceito de lifting state up, centralizando o estado global de autenticação no Context.

6. Dicas Importantes

- Use localStorage ou sessionStorage se quiser manter a autenticação após recarregar a página.
- Trabalhe com rotas para separar /login e /dashboard utilizando react-router-dom.
- Use useNavigate() ou Navigate para redirecionar após login/logout.
- Use componentes funcionais e hooks (useContext, useState).