

Título:

Autenticação com Formulário e Proteção de Rota usando Route Guard no Angular 19 Standalone

Motivação

A empresa deseja proteger o acesso à área administrativa do sistema. Apenas usuários autorizados devem conseguir acessar a tela de controle. Para isso, é necessário implementar uma autenticação simples baseada em formulário, validando os dados com um serviço interno, e restringindo o acesso via rota protegida (guard).

Situação

O sistema já possui componentes principais de navegação e visualização. Agora, foi solicitado que seja criada uma tela de login onde o usuário digite **usuário e senha**. Os dados serão validados diretamente por um service, sem conexão com backend. Caso estejam corretos, o usuário será redirecionado para uma **tela de painel de controle**. Essa tela só deve ser acessada por usuários autenticados.

Especificação

Implemente os seguintes requisitos:

1. Crie um componente **PainelComponent** contendo um **formulário com campos de login e senha**.
 2. Crie um service **AuthService** que:
 - Valide os dados digitados (usuário: "adm", senha: "123").
 - Mantenha o estado de autenticação.
 3. Crie um **guard de rota** chamado **authGuard** que:
 - Verifique se o usuário está autenticado.
 - Redirecione para /painel caso tente acessar o painel sem estar autenticado.
 4. Crie um componente **PainelControleComponent** que será exibido apenas **se o usuário estiver autenticado**.
 5. Configure o **app.routes.ts** com as seguintes rotas:
 - /painel → exibe o formulário de login
 - /painel-controle → exibe a área restrita, protegida pelo guard
 6. Estilize o formulário com CSS simples (input, botão, mensagens de erro).
 7. Comente cada linha importante dos arquivos TypeScript explicando o que o código está fazendo.
 8. Não utilize template inline — o layout deve estar no arquivo HTML externo.
-

Produto Esperado (P)

- Um **formulário funcional de login** com validação simples em um service.
- Um **painel de controle acessível apenas após login válido**.
- Uma rota protegida com guard (acesso negado se não estiver autenticado).
- Estilização básica do formulário e mensagens de erro.
- Códigos organizados, separados por arquivos .ts, .html e .css.

- Comentários no código explicando o funcionamento das funções e lógica de autenticação.

Obs: *Acesso ao componente painelComponente através da digitação da rota na URL*

logo após deves renderizar o componente com formulário na tela principal como no exemplo abaixo

Imagem:

URL	localhost:4200/painelcontrole	
Home - Sobre - Serviço - Contato		
<p>Tela de login:</p>		
LOGIN:		Se senha e login corretos, logar no componente: painelControle. Caso contrário redirecionar para home
<input type="text" value="adm"/>		
SENHA:		
<input type="text" value="123"/>		
<input type="button" value="LOGAR"/>		