



FUNDAÇÃO EDSON QUEIROZ
UNIVERSIDADE DE FORTALEZA - UNIFOR

DOCUMENTAÇÃO DA ESTRUTURA DE ROTAS

EQUIPE:

Larissa Evangelista Moreira - 2225988
Breno Guimarães - 2322669
Bruno Aguiar Santana - 2223824
Daniel de Moura Mascarenhas - 2322690
Matheus Silva Matos - 2312654
Walther Oliveira Pires - 2322679

1. Objetivo

A finalidade desta documentação é apresentar como ocorre a separação entre rotas públicas e privadas no sistema de Gerenciamento de Estoque, garantindo a segurança e a padronização no fluxo de navegação dos usuários. Neste documento também será abordado sobre a implementação destas rotas seguindo o padrão atual.

2. Identificação e Entendimento sobre Rotas Públicas e Privadas

2.1 Definição:

Rotas são os caminhos que um usuário pode acessar dentro da aplicação. Elas podem ser divididas em dois tipos:

- Rotas Públicas: Acessíveis sem necessidade de autenticação. Exemplos incluem páginas de login, cadastro, ou outras páginas abertas.
- Rotas Privadas: Exigem que o usuário esteja autenticado para acessar. Normalmente, apresentam dados sensíveis ou funcionalidades específicas do sistema.

2.2 Critérios de Separação:

As rotas que envolvem dados pessoais, informações internas ou funcionalidades restritas são chamadas de *rotas Privadas*. Já as rotas de acesso aberto como página de login, registro ou quaisquer informações públicas são chamadas de *rotas Públicas*.

Dentro do nosso sistema, a separação das rotas é feita através de dois componentes “PrivateRoute.jsx” e “PublicRoute.jsx” onde o Private Route será para proteger as rotas privadas e Public Route será para controlar o acesso às rotas públicas.

3. Segurança

Para garantir a proteção dos dados sensíveis podem ser utilizados diversos mecanismos como a Autenticação baseada em Tokens, Middleware de Segurança e AuthContext.

No nosso sistema, está sendo utilizado o Contexto de Autenticação (AuthContext) para gerenciar a proteção de rotas:

- PrivateRoute.jsx: Verifica se o usuário está autenticado (isLoggedIn) antes de liberar o acesso às rotas privadas, se o usuário não estiver logado, ele é redirecionado automaticamente para a tela de login.
- PublicRoute.jsx: Impede que usuários autenticados acessem rotas públicas como login e cadastro, redirecionando-os para o dashboard.

Além disto, também devem-se adotar boas práticas para garantir a manutenção de segurança das rotas tais como validação de permissão, expiração de token, proteger as rotas no backend e frontend.

4. Implementação

4.1 Componente PublicRoute: Ao utilizar este componente, a rota ficará disponível sem exigir autenticação.

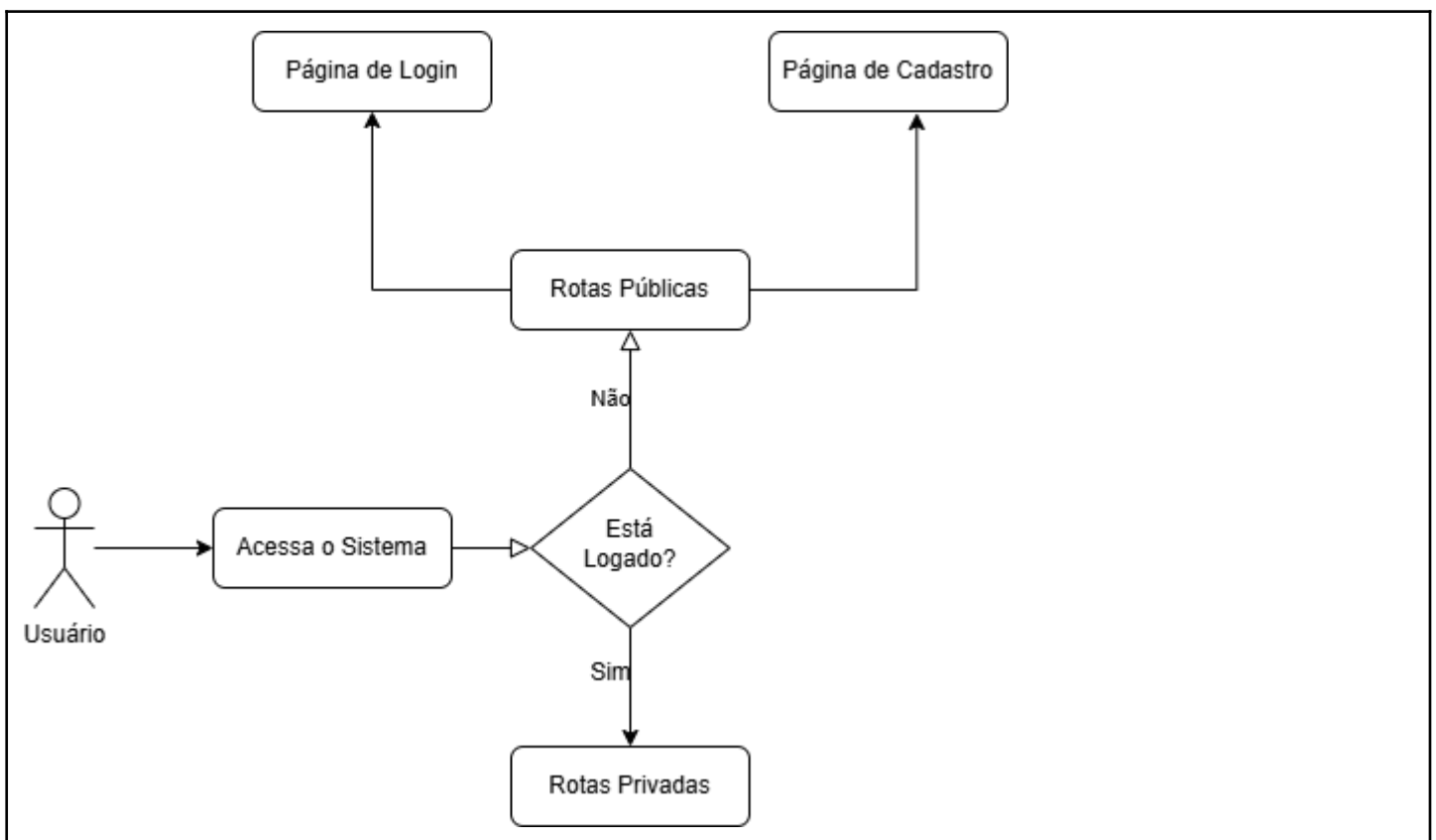
4.1.1 Código de implementação da PublicRoute:

```
<Route element={<PublicRoute />}>  
  <Route path="/login" element={<LoginPage /> } />  
</Route>
```

4.1.2 Fluxo de Caso - PublicRoute.jsx :

- Verifica se o usuário está logado (isLoggedIn).
- Se estiver logado, redireciona para /dashboard.
- Se não estiver logado, exibe o conteúdo da rota (<Outlet />).

4.1.3 Fluxograma:



4.2 Componente PrivateRoute: Ao utilizar este componente, a rota só ficará disponível se o usuário estiver autenticado.

4.2.1 Código de implementação da PrivateRoute:

```
<Route element={<PrivateRoute />}>
```

```
<Route path="/dashboard" element={<DashboardPage />} />
</Route>
```

4.2.2 Fluxo de Caso - PrivateRoute.jsx:

- Verifica se o usuário está logado (isLoggedIn).
- Se não estiver logado, redireciona para /login.
- Se estiver logado, exibe o conteúdo da rota (<Outlet />).

4.2.3 Fluxograma:

