

## REACT ROUTER

Para criar uma página com o React Router no React, você precisará seguir alguns passos básicos. O React Router é uma biblioteca que permite a navegação entre diferentes componentes em uma aplicação React.

Certifique-se de ter o React Router instalado em seu projeto. Você pode instalá-lo usando o npm ou o yarn. Abra o terminal e execute um dos seguintes comandos:

**npm install react-router-dom**

Documentação: <https://v5.reactrouter.com/web/guides/quick-start>

Na pasta src, crie uma pasta de nome routes e um arquivo auth.routes.jsx  
Endereço: src/routes/auth.routes.jsx

```
import { Routes, Route } from 'react-router-dom'

import { Home } from '../pages/Home'
import { Destiny } from '../pages/Destiny'

export function AuthRoutes() {
  return (
    <Routes>
      <Route path="/" element={<Home />} />
      <Route path="/destino" element={<Destiny />} />
    </Routes>
  )
}
```

Vamos analisar o código:

- Importa os componentes Routes e Route do pacote 'react-router-dom'. Estes são componentes fundamentais para a definição de rotas em uma aplicação React.
- Importa os componentes **Home** e **Destiny** dos arquivos correspondentes em '../pages/Home' e '../pages/Destiny'.

- Define um componente chamado AuthRoutes que será responsável por renderizar as rotas específicas da parte autenticada da aplicação.
- Usa o componente Routes para envolver as rotas da aplicação. O Routes é usado como um container para agrupar várias rotas.
- Usa o componente Route para definir as rotas individuais. Cada Route possui propriedades, como path e element. O path especifica o caminho da URL associado à rota, e element especifica o componente React que deve ser renderizado quando a rota é correspondida.
- A primeira rota (<Route path="/" element={<Home />} />) é correspondente ao caminho '/' e renderizará o componente Home quando a URL corresponder.
- A segunda rota (<Route path="/destino" element={<Destiny />} />) é correspondente ao caminho '/destino' e renderizará o componente Destiny quando a URL corresponder.

Crie o arquivo index.jsx, dentro da pasta src/routes

Endereço: src/routes/index.jsx

```
import { BrowserRouter } from "react-router-dom";

import { AuthRoutes } from "../auth.routes";

export function Routes() {
  return (
    <BrowserRouter>
      <AuthRoutes />
    </BrowserRouter>
  );
}
```

Inicialmente é Importado o componente BrowserRouter do pacote 'react-router-dom'. Este componente é necessário para envolver a aplicação com o contexto do roteador, permitindo o uso do histórico de navegação do navegador e o roteamento do React Router.

Em seguida é importado o componente AppRoutes do arquivo "../app.routes". Presumivelmente, AppRoutes contém a definição das rotas principais da aplicação.

Em export function, é definido um componente chamado **Routes**. O componente BrowserRouter é utilizado como componente de nível superior. Isso é essencial para fornecer a funcionalidade de roteamento para a aplicação.

Por fim, renderiza o componente **AppRoutes** dentro do **BrowserRouter**. Isso significa que as rotas definidas em AppRoutes estarão ativas quando a aplicação estiver sendo renderizada dentro do contexto do roteador.

No arquivo **main.jsx** é importante realizar algumas configurações.

```
import React from 'react'
import ReactDOM from 'react-dom/client'

import GlobalStyles from './styles/global'
import { ThemeProvider } from 'styled-components'

import theme from './styles/theme'

import { Routes } from './routes'

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <ThemeProvider theme={theme}>
      <GlobalStyles/>
      <Routes />
    </ThemeProvider>
  </React.StrictMode>
)
```

Foi importado o ReactDOM e o Routes. Por fim, adicionado o componentes Routes dentro do ThemeProvider.