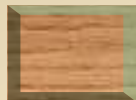
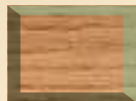




Introdução



router-dom



axios



Exemplo



Exercício



**React + router-dom
+ Axios**

Prof. Enzo
Seraphim

Profa. Bárbara
Pimenta Caetano



Agenda

Introdução

- ✓ Carregar Backend (ajuste cors)
- ✓ Carregar Frontend
- ✓ Rout
- ✓ Axios com JWT
- ✓ Entrega Final





Introdução

```
//download sigaa pblc01-aula12c.zip
//mover e descompactar em c:\users\aluno\vscode\
//abrir pblc01-aula12c no vscode
c:\users\aluno\vscode\pblc01-aula12c
//instalar dependência cors
npm install cors @types/cors
//inicializar postgres no docker
docker compose up -d
//gera Prisma Client
npx prisma generate
//migrar banco de dados para ambiente dev
npx prisma migrate dev --name init
//compila e executa seed.ts
npx prisma db seed
//Visualizar dados
npx prisma studio
```





Introdução

server.ts (BACKEND)

```
...  
import cors from 'cors';  
const app = express();  
app.use(cors({  
  origin: ['http://localhost:5173', 'http://localhost:3000'],  
  credentials: true  
}));  
app.use(express.json());  
...
```

//carregar o backend

npx run dev

//abrir browser

http://localhost:3000/docs/





Introdução

(FRONTEND) Novo VSCODE

```
//download sigaa pblc01-aula12c.zip  
//mover e descompactar em  
c:\users\aluno\vscode\  
//abrir pblc01-aula21b no vscode  
c:\users\aluno\vscode\pblc01-aula21b  
//instalar dependência cors  
npm install axios react-router-dom @types/react-router-dom  
//iniciar o servidor react  
npm run dev  
//abrir browser  
http://localhost:5173
```





Ambiente

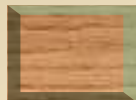
Introdução

```
//abrir pasta no vscode  
c:\users\aluno\vscode\pblc01-aula22  
//instalar mui  
npm install @mui/material @mui/icons-material  
@emotion/react @emotion/styled @types/react-router-dom  
axios react-router-dom  
  
//iniciar o servidor react  
npm run dev  
//abrir browser  
http://localhost:5173
```

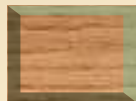




Introdução



router-dom



axios



Exemplo



Exercício



**React + router-dom
+ Axios**

Prof. Enzo
Seraphim

Profa. Bárbara
Pimenta Caetano



Rotas

Route

- ✓ Para navegar entre as páginas de uma aplicação React.JS precisaremos criar rota
- ✓ Cada rota representa uma funcionalidade que pode renderizar vários componentes
- ✓ Exemplo de rota para o site:

Home

Sobre





Route

Componentes

- ✓ `<Routes>` → seção de declaração de rotas
- ✓ `<Route>` → associa rota ao componente, parâmetros:
 - component:** que será exibido ao acessar a rota.
 - path:** URL que precisa ser acessado para mostrar o componente, definido pelo parâmetro **component**.
 - exact:** Garante que a rota só seja renderizada quando a URL for exatamente `/`.



Criar ./Rota.tsx

```
import React from 'react';
import {Route, Routes} from 'react-router-dom';
import Home from './Home';
import Sobre from './Sobre';
import NotFound from './NotFound';
const Rota: React.FC = () => {
  return (
    <Routes>
      <Route path="/" element={<Home />} />
      <Route path="/sobre" element={<Sobre />} />
      <Route path="*" element={<NotFound />} />
    </Routes>
  );
};
export default Rota;
```



Componentes

Route

- ✓ `<BrowserRouter>` → responsável por informar a nossa aplicação que teremos um roteamento de componentes.
- ✓ `<Link>` → substituir a nossa tag `<a>` do HTML para acessar as páginas do próprio projeto, parâmetro: **to**: nome da rota que será acessada pela URL.
- ✓ Entre as tags (abertura e fechamento) será inserido o conteúdo que ficará disponível para ser clicado
- ✓ `App.tsx` → ajudado com **BrowserRouter e Link**





Route



```
import { ThemeProvider } from './theme/ThemeContext';
import Header from './components/Header';
import { BrowserRouter } from 'react-router-dom';
import { Link } from 'react-router-dom';
import Rota from './Rota';
const App = () => {
  return (
    <ThemeProvider>
      <Header />
      <BrowserRouter>
        <nav>
          <ul>
            <li><Link to="/">Início</Link></li>
            <li><Link to="/sobre">Sobre</Link></li>
          </ul>
        </nav>
        <Rota />
      </BrowserRouter>
    </ThemeProvider>
  );
};
export default App;
```

App.tsx



Componentes

Route

- ✓ `<BrowserRouter>` → responsável por informar a nossa aplicação que teremos um roteamento de componentes.
- ✓ `<Link>` → substituir a nossa tag `<a>` do HTML para acessar as páginas do próprio projeto, parâmetro: **to**: nome da rota que será acessada pela URL.
- ✓ Entre as tags (abertura e fechamento) será inserido o conteúdo que ficará disponível para ser clicado
- ✓ `App.tsx` → ajudado com **BrowserRouter e Link**





Route

Home.tsx

```
import React from 'react';
const Home: React.FC = () => {
  return <h2>Bem-vindo à Home Page!</h2>;
};
export default Home;
```

Sobre.tsx

```
import React from 'react';
import { Link } from 'react-router-dom';
const Sobre: React.FC = () => {
  return(
    <>
      <h2>Sobre nossa aplicação</h2>
      <Link to="/">retornar a página inicial</Link>
    </>
  );
};
export default Sobre;
```





Route

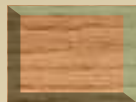
NotFound.tsx

```
import React from 'react';
import { Link } from 'react-router-dom';

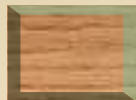
const NotFound: React.FC = () => {
  return(
    <>
      <h2>Página não encontrada!</h2>
      <Link to="/">retornar a página inicial</Link>
    </>
  );
};

export default NotFound;
```





Introdução



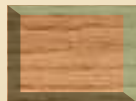
router-dom



axios



Exemplo



Exercício



**React + router-dom
+ Axios**

Prof. Enzo
Seraphim

Profa. Bárbara
Pimenta Caetano



Axios

Axios

- ✓ Cliente HTTP baseado-em-promessas para o node.js e para o navegador.
- ✓ É isomórfico (pode rodar no navegador e no node.js com a mesma base de código).
- ✓ Servidor usa código nativo do node.js (módulo http)
- ✓ Cliente (navegador) usa XMLHttpRequests.





Arquivos

Axios



src/auth/AuthContext.tsx



src/api/axios.tsx



```
import { createContext, useContext, useState, useEffect } from 'react';
import type { ReactNode } from 'react';
interface AuthContextType {
  token: string | null;
  setToken: (token: string | null) => void;
  logout: () => void;
}
const AuthContext = createContext<AuthContextType | undefined>(undefined);
export function AuthProvider({ children }: { children: ReactNode }) {
  const [token, setToken] = useState<string | null>(() => {
    return localStorage.getItem('token');
  });
  useEffect(() => {
    if (token) {
      localStorage.setItem('token', token);
    } else {
      localStorage.removeItem('token');
    }
  }, [token]);
  const logout = () => {
    setToken(null);
    localStorage.removeItem('token');
  };
};
```

src/auth/AuthContext.tsx

```
return (  
  <AuthContext.Provider value={{ token, setToken, logout }}>  
    {children}  
  </AuthContext.Provider>  
);  
}  
export function useAuth() {  
  const context = useContext(AuthContext);  
  if (!context) {  
    throw new Error('useAuth deve ser usado dentro de AuthProvider');  
  }  
  return context;  
}
```

src/auth/AuthContext.tsx

```
import axios from 'axios';

const api = axios.create({
  baseURL: 'http://localhost:3000',
});

api.interceptors.request.use((config) => {
  const token = localStorage.getItem('token');
  if (token) {
    config.headers.Authorization = `Bearer ${token}`;
  }
  return config;
});

export default api;
```

src/api/axios.tsx

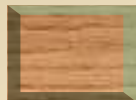
```
import { Navigate } from 'react-router-dom';
import { useAuth } from '../auth/AuthContext';
import type { JSX } from 'react';

export default function PrivateRoute({ children }: {
  children: JSX.Element }) {
  const { token } = useAuth();
  return token ? children : <Navigate to="/login" />;
}
```

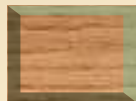
src/routes/PrivateRoute.tsx



Introdução



Route



Axios



Exemplo



Exercício



**React + router-dom
+ Axios**

Prof. Enzo
Seraphim

Profa. Bárbara
Pimenta Caetano



Exemplo

Exemplo

Login e Logout




```

import { useState } from 'react';
import {
  Box, Button, TextField, Typography, Snackbar, Alert,
} from '@mui/material';
import { useNavigate } from 'react-router-dom';
import { useAuth } from '../auth/AuthContext';
import axios from '../api/axios';
export default function Login() {
  const { setToken } = useAuth();
  const navigate = useNavigate();
  const [nome, setEmail] = useState('');
  const [senha, setSenha] = useState('');
  const [erro, setErro] = useState('');
  const handleLogin = async () => {
    try {
      const response = await axios.post('/auth/login', {
        nome,
        senha,
      }, {
        withCredentials: true // <--- apenas se back usa cookie auth
      })
      const { token } = response.data;
      setToken(token); // atualiza o contexto e salva no localStorage
      navigate('/');
    }
  }
}
//Continua...

```

src/pages/Login.tsx

```

    catch (err: any) {
      setErro(err.response?.data?.message || 'Erro ao fazer login');
    }
  }
  return (
    <Box display="flex" flexDirection="column" alignItems="center"
      justifyContent="center" height="100vh" px={2} gap={2} >
      <Typography variant="h4">Login</Typography>
      <TextField label="Nome de usuário" value={nome}
        onChange={(e) => setEmail(e.target.value)} fullWidth />
      <TextField label="Senha" type="password" value={senha}
        onChange={(e) => setSenha(e.target.value)} fullWidth />
      <Button variant="contained" fullWidth onClick={handleLogin}>
        Entrar
      </Button>
      <Snackbar open={!!erro} autoHideDuration={4000} onClose={() =>
setErro('')}>
        <Alert severity="error">{erro}</Alert>
      </Snackbar>
    </Box>
  );
}

```

src/pages/Login.tsx

```

import { AppBar, Toolbar, Typography, IconButton, Button, Box } from '@mui/material';
import LightModeIcon from '@mui/icons-material/LightMode';
import DarkModeIcon from '@mui/icons-material/DarkMode';
import { useThemeMode } from '../theme/ThemeContext';
import { useNavigate } from 'react-router-dom';
import { useAuth } from '../auth/AuthContext';
const Header = () => {
  const { isDark, toggleTheme } = useThemeMode();
  const { logout } = useAuth();
  const navigate = useNavigate();
  const handleLogout = () => {
    logout();
    navigate('/login');
  };
  return (
    <AppBar position="static" color="primary" elevation={2}>
      <Toolbar sx={{ display: 'flex', justifyContent: 'space-between' }}>
        <Typography variant="h6">Meu App</Typography>
        <Box sx={{ display: 'flex', alignItems: 'center', gap: 2 }}>
          <IconButton color="inherit" onClick={toggleTheme}>
            {isDark ? <LightModeIcon /> : <DarkModeIcon />}
          </IconButton>
          <Button color="inherit" onClick={handleLogout}>Sair</Button>
        </Box></Toolbar></AppBar>
      </>
    );
};
export default Header;

```

src/components/Header.tsx

```

import { useEffect, useState } from 'react';
import {
  Box, Typography, CircularProgress, Alert, Card, CardContent, CardHeader, Container,
} from '@mui/material';
import { Grid } from '@mui/material';
import axios from '../api/axios';
import Header from '../components/Header';
interface Poder {
  id: string;
  nome: string;
  descricao: string;
}
export default function Dashboard() {
  const [poderes, setPoderes] = useState<Poder[]>([]);
  const [carregando, setCarregando] = useState(true);
  const [erro, setErro] = useState('');
  useEffect(() => {
    const carregarPoderes = async () => {
      try {
        const response = await axios.get('/poderes');
        setPoderes(response.data);
      } catch (err: any) { setErro('Erro ao carregar os poderes'); }
      finally { setCarregando(false); }
    };
    carregarPoderes();
  }, []); //Continua ...

```

src/pages/Dashboard.tsx

```

return (
  <> <Header />
  <Container maxWidth="lg" sx={{ py: 4 }}>
    <Typography variant="h4" gutterBottom textAlign="center">
      Lista de Poderes
    </Typography>
    {carregando ? (
      <Box display="flex" justifyContent="center" mt={4}>
        <CircularProgress />
      </Box>
    ) : erro ? ( <Alert severity="error">{erro}</Alert>
    ) : ( <Grid container spacing={3}>
      {poderes.map((poder) => (
        <Grid size={12}>
          <Card elevation={3}>
            <CardHeader title={poder.nome} />
            <CardContent>
              <Typography variant="body2" color="text.secondary">
                {poder.descricao}
              </Typography>
            </CardContent>
          </Card> </Grid> ) )}
        </Grid> )} </Container> </> ); }

```

src/pages/Dashboard.tsx

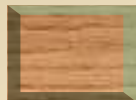
```
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import { AuthProvider } from '../auth/AuthContext';
import { ThemeProvider } from '../theme/ThemeContext';
import PrivateRoute from '../routes/PrivateRoute';
import Login from '../pages/Login';
import Dashboard from '../pages/Dashboard';

function App() {
  return (
    <ThemeProvider>
      <AuthProvider>
        <Router> <Routes>
          <Route path="/login" element={<Login />} />
          <Route
            path="/"
            element={
              <PrivateRoute>
                <Dashboard />
              </PrivateRoute>
            } />
        </Routes></Router></AuthProvider></ThemeProvider>
      </>
    )
  }
export default App;
```

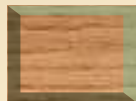
src/App.tsx



Introdução



Route



Axios



Exemplo



Exercício



**React + Route
+ Axios**

Prof. Enzo
Seraphim

Profa. Bárbara
Pimenta Caetano



Frontend

- ✓ Zip do Frontend consumindo a API Backend desenvolvida na primeira parte da disciplina contendo:

Exercício

- ✓ login da sua aplicação
- ✓ tela de listagem



**Prof. Enzo
Seraphim**

**Profa. Bárbara
Pimenta Caetano**

Os logotipos, marcas comerciais e nomes de produtos citados nesta publicação tem apenas o propósito de identificação e podem ser marcas registradas de suas respectivas companhias.



**React + Route
+ Axios**