
Desenvolvimento WEB

-

Full Stack Completo: Java + React

Controle de Acesso

Front-end



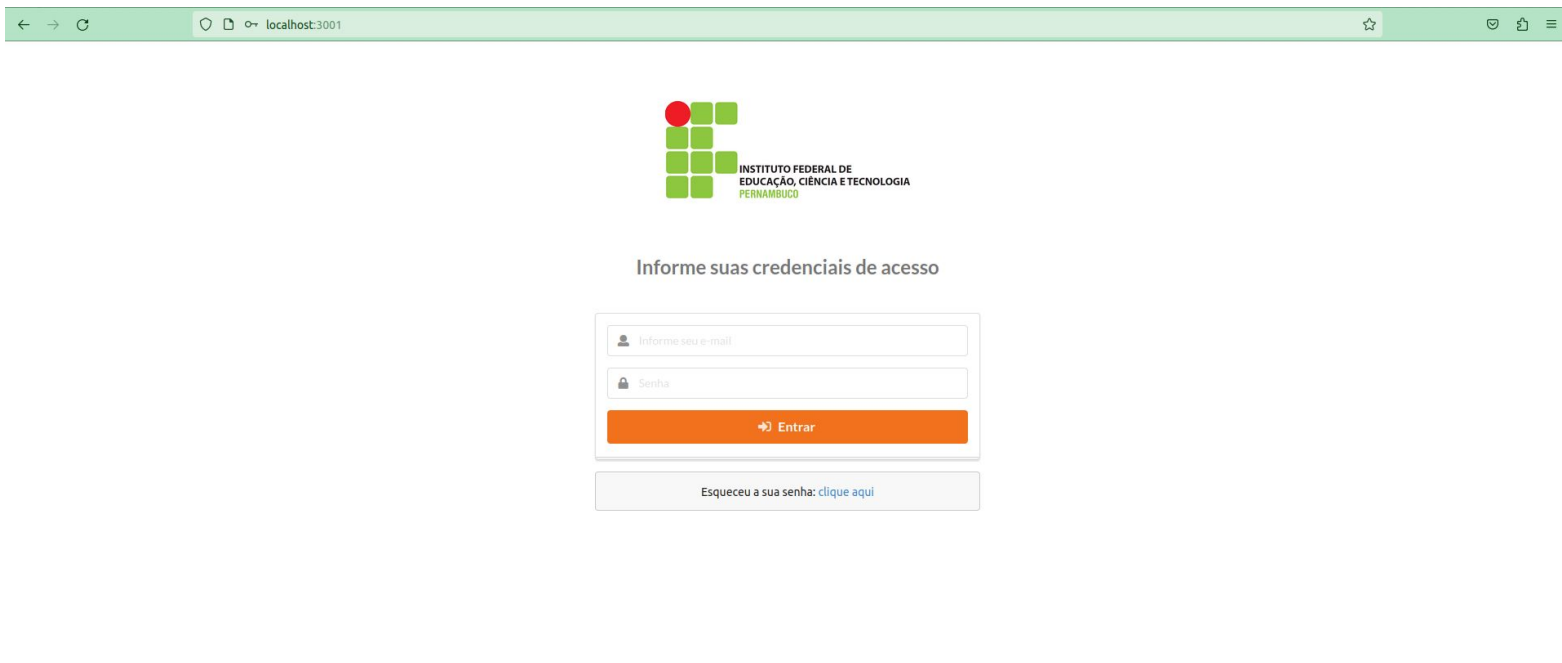
Informe suas credenciais de acesso

Entrar

Esqueceu a sua senha: [clique aqui](#)

Controle de Acesso :: Front-end

- **Objetivo:** implementar o controle de acesso no projeto do front-end




← → ↻

localhost:3001

☆

🔒 🏠 ☰



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PERNAMBUCO

Informe suas credenciais de acesso

Informe seu e-mail

Senha

➔ Entrar

Esqueceu a sua senha: [clique aqui](#)

Controle de Acesso :: Front-end

- Dentro da pasta `views`, crie uma nova pasta chamada `login`, e dentro da pasta `login` crie um novo arquivo chamado `FormLogin.jsx`. Código deste arquivo:

```
import axios from 'axios';
import React, { useState } from 'react';
import { Link, useNavigate } from 'react-router-dom';
import { Button, Form, Grid, Header, Image, Message, Segment } from 'semantic-ui-react';
import { notifyError } from '../../views/util/Util';
import { registerSuccessfulLoginForJwt } from '../../util/AuthenticationService';

export default function FormLogin () {

  const navigate = useNavigate();

  const [username, setUsername] = useState('');
  const [senha, setSenha] = useState('');

  function entrar() {

    if (username !== '' && senha !== '') {

      let authenticationRequest = {
        username: username,
        password: senha,
      }

      axios.post("http://localhost:8085/api/auth", authenticationRequest)
        .then((response) => {

          registerSuccessfulLoginForJwt(response.data.token, response.data.expiration)
          navigate("/home");

        })
        .catch((error) => {

          notifyError('Usuário não encontrado')

        })

    }

  }

}
```

```
return(

  <div>

    <Grid textAlign='center' style={{ height: '70vh' }} verticalAlign='middle'>

      <Grid.Column style={{ maxWidth: 500 }}>

        <center> <Image src='/logo-IFPE.png' size='medium' /> </center>

        <div style={{marginTop: '15%'}}>
          <Header as='h2' color='grey' textAlign='center'>
            Informe suas credenciais de acesso
          </Header>
        </div> <br/> <br/>

        <Form>
          <Segment stacked>

            <Form.Input
              fluid
              value={username}
              onChange={e => setUsername(e.target.value)}
              icon='user'
              iconPosition='left'
              placeholder='Informe seu e-mail'
              required
              maxLength="100"
            />

            <Form.Input
              fluid
              value={senha}
              onChange={e => setSenha(e.target.value)}
              icon='lock'
              iconPosition='left'
              type='password'
              placeholder='Senha'
              required
              maxLength="100"
            />

            <Button
              fluid
              size='large'
              color='orange'
              icon='sign in alternate'
              content='Entrar'
              onClick={() => entrar()} />

          </Segment>
        </Form>

        <Message>
          Esqueceu a sua senha: <Link>clique aqui</Link>
        </Message>

      </Grid.Column>
    </Grid>
  </div>

)
```

Controle de Acesso :: Front-end

- Dentro da pasta `util`, crie um novo arquivo chamado `AuthenticationService.js`. Código deste arquivo:

```
import axios from 'axios';

export const TOKEN_SESSION_ATTRIBUTE_NAME = 'token'
export const EXPIRATION_SESSION_ATTRIBUTE_NAME = 'expiration'

export const registerSuccessfulLoginForJwt = (token, expiration) => {

  localStorage.setItem(TOKEN_SESSION_ATTRIBUTE_NAME, token)
  localStorage.setItem(EXPIRATION_SESSION_ATTRIBUTE_NAME, expiration)

  setupAxiosInterceptors()

}

export const setupAxiosInterceptors = () => {

  let token = createJWTToken(localStorage.getItem(TOKEN_SESSION_ATTRIBUTE_NAME))

  if (isUserLoggedIn()) {
    axios.defaults.headers.common['Authorization'] = token;
  } else {
    delete axios.defaults.headers.common['Authorization'];
  }
}

export const createJWTToken = (token) => {
  return 'Bearer ' + token
}
```

```
export const logout = () => {

  localStorage.clear()
  delete axios.defaults.headers.common['Authorization'];
}

export const isTokenExpired = () => {

  let expiration = localStorage.getItem(EXPIRATION_SESSION_ATTRIBUTE_NAME)
  return expiration === null || expiration < new Date().getTime()
}

export const isUserLoggedIn = () => {

  let user = localStorage.getItem(TOKEN_SESSION_ATTRIBUTE_NAME)

  if (user === null) {
    return false
  } else {
    return true
  }
}

export const getToken = () => {

  let token = localStorage.getItem(TOKEN_SESSION_ATTRIBUTE_NAME)
  if (token === null) return ''
  return token
}
```

Controle de Acesso :: Front-end

- Ainda na pasta `util`, crie um novo componente chamado `ProtectedRoute.js`.
Código deste arquivo:

```
import { Navigate } from "react-router-dom";
import { isUserLoggedIn } from '../util/AuthenticationService';

export const ProtectedRoute = ({ children }) => {

  if (!isUserLoggedIn()) {
    // user is not authenticated
    return <Navigate to="/" />;
  }

  return children;
};
```

Controle de Acesso :: Front-end

- Refatore o componente `Rotas.jsx` para utilizar o componente que protege as rotas do acesso não autenticado.
- Exemplo:

```
import React from 'react';
import { Route, Routes } from "react-router-dom";
import { ProtectedRoute } from './views/util/ProtectedRoute';
...
import ListCliente from './views/cliente/ListCliente';
import Home from './views/home/Home';
...
function Rotas() {
  return (
    <>
      <Routes>

        <Route path="/" element={ <FormLogin/> } />

        <Route
          path="/home"
          element={
            <ProtectedRoute>
              <Home />
            </ProtectedRoute>
          }
        />

        <Route
          path="/list-cliente"
          element={
            <ProtectedRoute>
              <ListCliente />
            </ProtectedRoute>
          }
        />

      </>
    )
  }
}
```

Controle de Acesso :: Front-end

Acrescente um item de menu no `MenuSistema.jsx` para o logout do sistema:

```
...
import { logout } from '../views/util/AuthenticationService';
...

return(
  <>
    <Menu inverted>
      ...
      <Menu.Item
        className='navbar__item--mobile'
        onClick={logout}
        content='Sair'
        as={Link}
        to='/'
      />
      ...
    </Menu>
  </>
  ...
)
```


Dúvidas



Exercício



Implemente o controle de acesso no seu **projeto do front-end**.

A screenshot of a web browser window. The address bar shows 'localhost:3001'. The page features a logo for 'INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA PERNAMBUCO' consisting of a 3x3 grid of green squares with a red square in the top-left corner. Below the logo, the text 'Informe suas credenciais de acesso' is displayed. The login form has two input fields: 'Informe seu e-mail' and 'Senha'. Below these fields is an orange button with a right-pointing arrow and the text 'Entrar'. At the bottom of the form, there is a link that says 'Esqueceu a sua senha: [clique aqui](#)'.

The background features two large, solid green abstract shapes. One is a semi-circle on the left side, and the other is a more complex, organic shape on the right side.

Obrigado !