

Componentes

1. BuscarAluno

O componente **BuscarAluno** permite que o usuário busque um aluno pela matrícula e exiba as informações do aluno ou uma mensagem de erro caso não o encontre.

BuscarAluno.jsx

```
import React, { useState } from 'react';
import { listarPorMatricula } from '../../services/alunoService'; //
Importa a função do serviço para buscar o aluno por matrícula
import styles from './BuscarAluno.module.css'; // Importa o CSS Module para
estilização

function BuscarAluno() {
  const [matricula, setMatricula] = useState(''); // Estado para
armazenar o valor da matrícula
  const [aluno, setAluno] = useState(null); // Estado para armazenar os
dados do aluno
  const [errorMsg, setErrorMsg] = useState(''); // Estado para armazenar
a mensagem de erro

  // Função para buscar o aluno utilizando a matrícula
  const buscarAluno = async () => {
    try {
      const response = await listarPorMatricula(matricula); // Chama
o serviço para buscar o aluno
      setAluno(response.data); // Atualiza o estado com os dados do
aluno
      setErrorMsg(''); // Limpa a mensagem de erro
    } catch (error) {
      setErrorMsg('Aluno não encontrado!'); // Exibe mensagem de erro
caso o aluno não seja encontrado
      setAluno(null); // Limpa os dados do aluno
    }
  };

  return (
    <div className={styles.container}> {/* Contêiner principal do
componente */}
      <h2 className={styles.titulo}>Buscar Aluno</h2> {/* Título */}
      <input
        type="text"
        value={matricula} // Valor do input é o estado "matricula"
        onChange={(e) => setMatricula(e.target.value)} // Atualiza
o estado conforme o valor digitado
        placeholder="Digite a matrícula"
        className={styles.input} // Estilos do input
      />
      <button onClick={buscarAluno} className=
```

```
{styles.botoao}>Buscar</button> {/* Botão para disparar a busca */}
  {errorMsg && <p className={styles.error}>{errorMsg}</p>} {/*
Exibe mensagem de erro caso exista */}
  {aluno && (
    <div className={styles.dadosAluno}>
      <h3>Dados do Aluno</h3>
      <p><strong>Nome:</strong> {aluno.nome}</p>
      <p><strong>Email:</strong> {aluno.email}</p>
      <p><strong>Matrícula:</strong> {aluno.matricula}</p>
    </div>
  )}
</div>
);
}

export default BuscarAluno; // Exporta o componente para ser utilizado em
outras partes do app
```

BuscarAluno.module.css

```
.container {
  display: flex;
  flex-direction: column;
  align-items: center;
  padding: 2rem;
}

.titulo {
  font-size: 1.5rem;
  margin-bottom: 1rem;
}

.input {
  padding: 0.5rem;
  font-size: 1rem;
  width: 200px;
  margin-bottom: 1rem;
}

.botao {
  padding: 0.5rem;
  font-size: 1rem;
  cursor: pointer;
}

.error {
  color: red;
  margin-top: 1rem;
}

.dadosAluno {
  margin-top: 1rem;
  font-size: 1.2rem;
}
```

2. ListarAlunos

O componente **ListarAlunos** exibe todos os alunos cadastrados no sistema, com a possibilidade de editar ou excluir um aluno.

ListarAlunos.jsx

```
import React, { useState, useEffect } from 'react';
import { listarTodos } from '../../services/alunoService'; // Importa a
função do serviço para listar todos os alunos
import styles from './ListarAlunos.module.css'; // Importa o CSS Module
para estilização

function ListarAlunos() {
  const [alunos, setAlunos] = useState([]); // Estado para armazenar a
lista de alunos
  const [errorMsg, setErrorMsg] = useState(''); // Estado para armazenar
a mensagem de erro

  // Função que é chamada ao carregar o componente, para buscar os alunos
  useEffect(() => {
    async function fetchAlunos() {
      try {
        const response = await listarTodos(); // Chama o serviço
para listar todos os alunos
        setAlunos(response.data); // Atualiza o estado com os dados
dos alunos
      } catch (error) {
        setErrorMsg('Erro ao carregar os alunos'); // Exibe
mensagem de erro caso não consiga carregar os alunos
      }
    }
    fetchAlunos(); // Chama a função para buscar os alunos
  }, []); // O array vazio garante que a função seja chamada apenas uma
vez quando o componente for montado

  return (
    <div className={styles.container}>
      <h2 className={styles.titulo}>Lista de Alunos</h2>
      {errorMsg && <p className={styles.error}>{errorMsg}</p>} {/*
Exibe mensagem de erro se houver */}
      <ul className={styles.lista}>
        {alunos.map((aluno) => (
          <li key={aluno.matricula} className={styles.item}>
            {aluno.nome} - {aluno.email} - Matrícula:
{aluno.matricula}
          </li>
        ))}
      </ul>
    </div>
  );
}
```

export default ListarAlunos; // Exporta o componente para ser utilizado em
outras partes do app

ListarAlunos.module.css

```
.container {
  display: flex;
  flex-direction: column;
  align-items: center;
  padding: 2rem;
}

.titulo {
  font-size: 1.5rem;
  margin-bottom: 1rem;
}

.lista {
  list-style-type: none;
  padding: 0;
  width: 100%;
}

.item {
  padding: 0.5rem;
  border-bottom: 1px solid #ddd;
  font-size: 1.1rem;
}

.error {
  color: red;
  margin-bottom: 1rem;
}
```

3. CadastroAluno

O componente **CadastroAluno** permite que o usuário cadastre um novo aluno no sistema.

CadastroAluno.jsx

```
import React, { useState } from 'react';
import { criar } from '../../services/alunoService'; // Importa a função do
serviço para cadastrar um aluno
import styles from './CadastroAluno.module.css'; // Importa o CSS Module
para estilização

function CadastroAluno() {
  const [nome, setNome] = useState(''); // Estado para armazenar o nome
do aluno
  const [email, setEmail] = useState(''); // Estado para armazenar o
email do aluno
  const [matricula, setMatricula] = useState(''); // Estado para
armazenar a matrícula do aluno
  const [errorMsg, setErrorMsg] = useState(''); // Estado para armazenar
a mensagem de erro
  const [successMsg, setSuccessMsg] = useState(''); // Estado para
armazenar a mensagem de sucesso
```

```

// Função para cadastrar o aluno
const cadastrarAluno = async () => {
  try {
    await criar({ nome, email, matricula }); // Chama o serviço
    para cadastrar o aluno
    setSuccessMsg('Aluno cadastrado com sucesso!'); // Exibe
    mensagem de sucesso
    setErrorMsg(''); // Limpa a mensagem de erro
  } catch (error) {
    setErrorMsg('Erro ao cadastrar aluno!'); // Exibe mensagem de
    erro caso falhe
    setSuccessMsg(''); // Limpa a mensagem de sucesso
  }
};

return (
  <div className={styles.container}>
    <h2 className={styles.titulo}>Cadastrar Aluno</h2>
    <input
      type="text"
      value={nome}
      onChange={(e) => setNome(e.target.value)} // Atualiza o
estado conforme o valor digitado
      placeholder="Nome"
      className={styles.input} // Estilos do input
    />
    <input
      type="email"
      value={email}
      onChange={(e) => setEmail(e.target.value)} // Atualiza o
estado conforme o valor digitado
      placeholder="Email"
      className={styles.input} // Estilos do input
    />
    <input
      type="text"
      value={matricula}
      onChange={(e) => setMatricula(e.target.value)} // Atualiza
o estado conforme o valor digitado
      placeholder="Matrícula"
      className={styles.input} // Estilos do input
    />
    <button onClick={cadastrarAluno} className=
{styles.botao}>Cadastrar</button> {/* Botão para disparar o cadastro */}
    {errorMsg && <p className={styles.error}>{errorMsg}</p>} {/*
Exibe mensagem de erro caso exista */}
    {successMsg && <p className={styles.success}>{successMsg}</p>}
    {/* Exibe mensagem de sucesso caso exista */}
  </div>
);
}

export default CadastroAluno; // Exporta o componente para ser utilizado em
outras partes do app

```

CadastroAluno.module.css

```
.container {
  display: flex;
  flex-direction: column;
  align-items: center;
  padding: 2rem;
}

.titulo {
  font-size: 1.5rem;
  margin-bottom: 1rem;
}

.input {
  padding: 0.5rem;
  font-size: 1rem;
  width: 200px;
  margin
- bottom: 1rem;
}

.botao {
  padding: 0.5rem;
  font-size: 1rem;
  cursor: pointer;
}

.error {
  color: red;
  margin-top: 1rem;
}

.success {
  color: green;
  margin-top: 1rem;
}
```