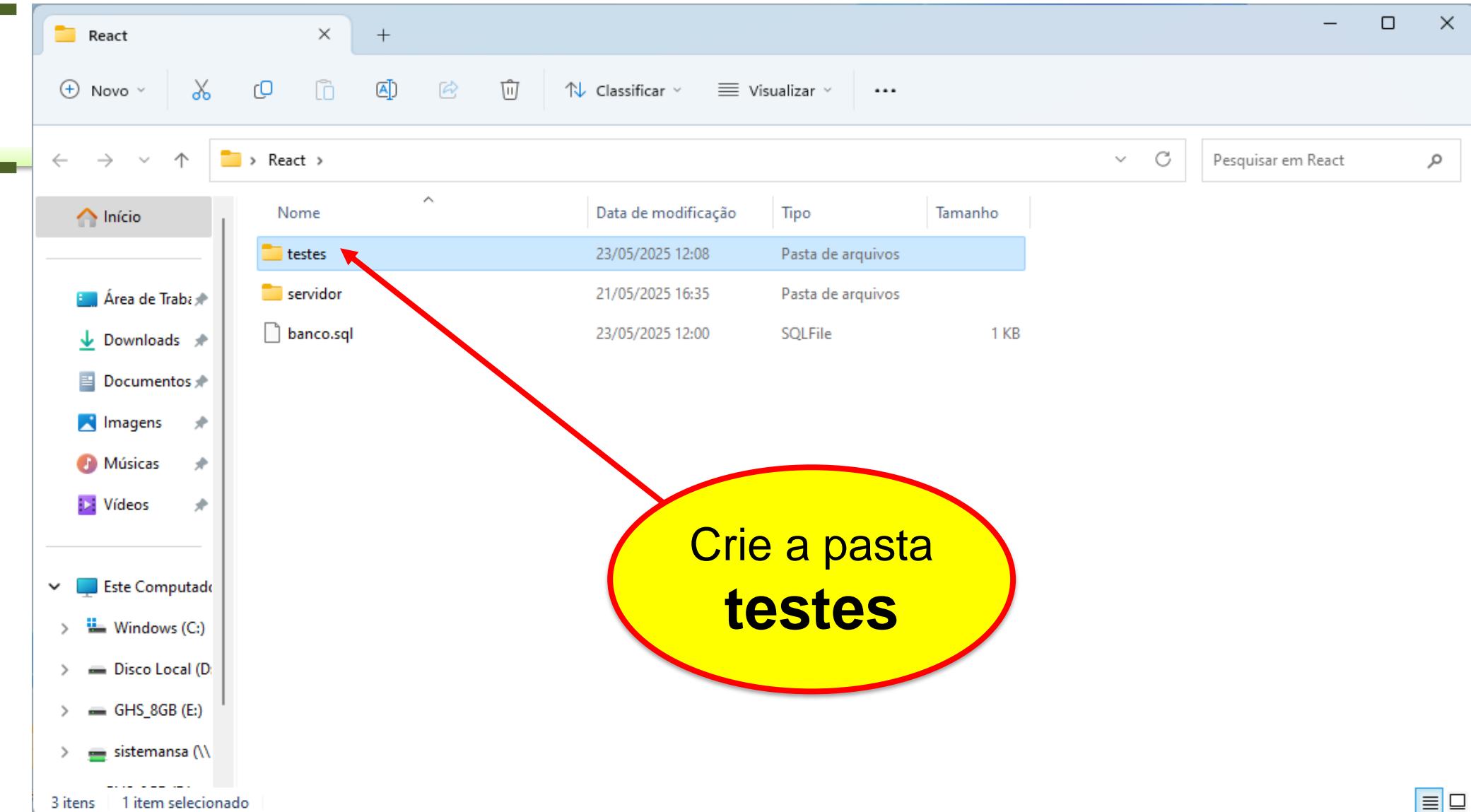


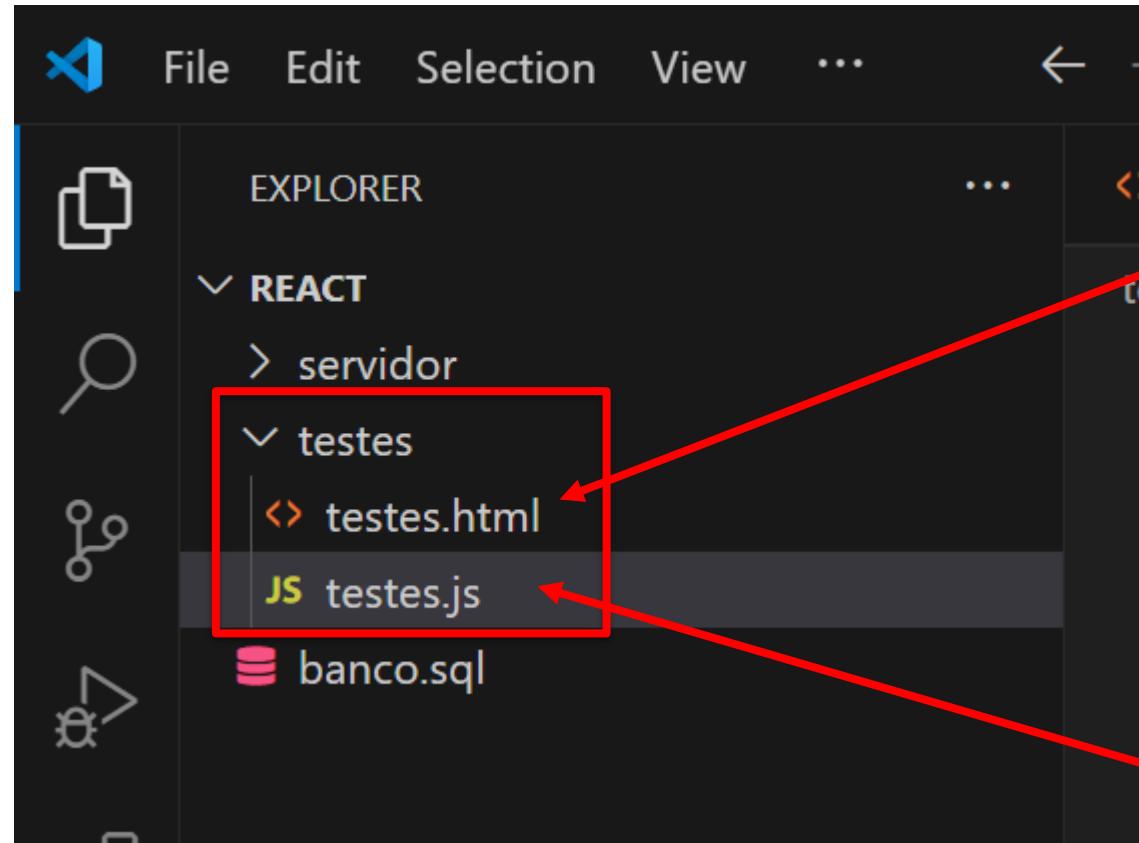
## 10-2 - Criação dos Testes em HTML e JS

**Guilherme Henrique de Souza**

[guilherme.souza@etec.sp.gov.br](mailto:guilherme.souza@etec.sp.gov.br)

[guilherme.souza183@fatec.sp.gov.br](mailto:guilherme.souza183@fatec.sp.gov.br)





Crie o arquivo  
**testes.html**

Crie o arquivo  
**testes.js**

# Funções do arquivo **testes.js**

**Guilherme Henrique de Souza**

guilherme.souza@etec.sp.gov.br

guilherme.souza183@fatec.sp.gov.br

The image shows a screenshot of the Visual Studio Code (VS Code) interface. On the left, the Explorer pane displays a file structure for a React project. The 'testes' folder contains 'testes.html' and 'JS testes.js'. A red box highlights 'JS testes.js'. A large yellow arrow points from the Explorer pane towards the Editor pane. In the Editor pane, a file named 'testes.js' is open, showing the following code:

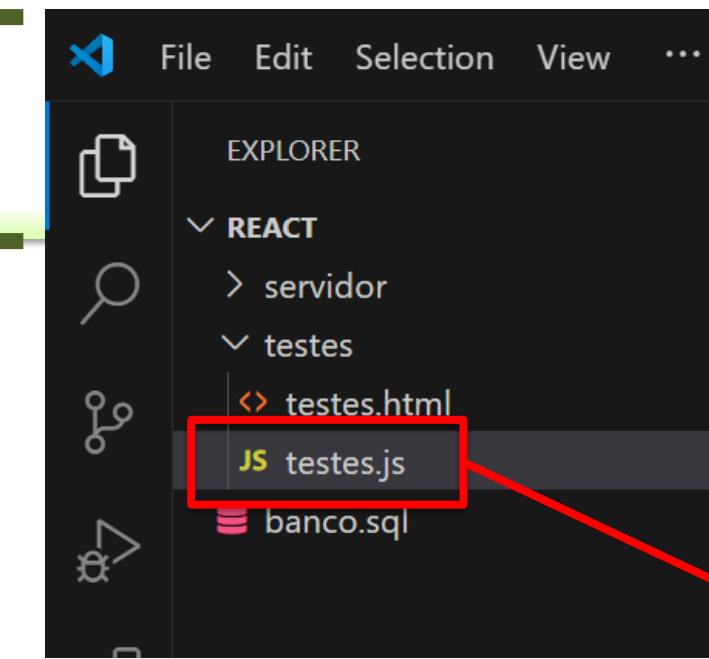
```
1  tester > JS testes.js > cadastrar
2  function consultar()
3  {
4  }
5
6  function consultarPorCodigo(codigo)
7  {
8  }
9
10 function cadastrar()
11 {
12 }
13
14 function alterar(codigo)
15 {
16 }
17
18 function excluir(codigo)
19 {
20 }
21
22 }
```

The image shows a screenshot of the Visual Studio Code (VS Code) interface. On the left, the Explorer sidebar displays a file tree for a 'REACT' project. The 'testes' folder contains 'testes.html' and 'JS testes.js'. A red box highlights 'JS testes.js'. A red arrow points from this highlighted file to the code editor on the right. The code editor shows the contents of 'JS testes.js'. The title bar indicates the file is 'JS testes.js'. The code itself consists of two functions: 'consultar()' and 'consultarPorCodigo(codigo)'. Both functions are currently empty, as indicated by the curly braces {}.

```
JS testes.js
testes > JS testes.js > cadastrar

function consultar()
{
}

function consultarPorCodigo(codigo)
{
}
```



```
10
11 function cadastrar() |
12 {
13 }
14
15
16 function alterar(codigo)
17 {
18 }
19
20
21 function excluir(codigo)
22 {
23 }
24 }
```

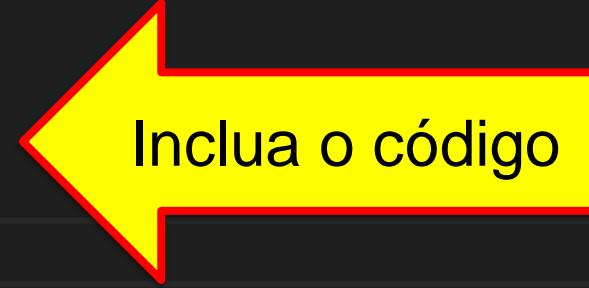
# Arquivo **testes.html**

**Guilherme Henrique de Souza**

[guilherme.souza@etec.sp.gov.br](mailto:guilherme.souza@etec.sp.gov.br)

[guilherme.souza183@fatec.sp.gov.br](mailto:guilherme.souza183@fatec.sp.gov.br)

```
<> testes.html X  
testes > <> testes.html > html > body  
1   <!DOCTYPE html>  
2   <html lang="en">  
3   <head>  
4       <meta charset="UTF-8">  
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">  
6       <title>Document</title>  
7   </head>  
8   <body>  
9       <h1>Testes com a API</h1>  
10  
11  
12  
13   </body>  
14   </html>
```



Inclua o código

```
<body>
  <h1>Testes com a API</h1>

  <p>
    <input type="button" value="Consultar" onclick="consultar()">
    <input type="button" value="Consultar Por Código" onclick="consultarPorCodigo(1)">

    <input type="button" value="Cadastrar" onclick="cadastrar()">
    <input type="button" value="Alterar" onclick="alterar(1)">
    <input type="button" value="Excluir" onclick="excluir(1)">
  </p>
```

Inclua o código acima para exibir 5 botões, veja que o evento onclick já está associado com a respectiva função no JavaScript

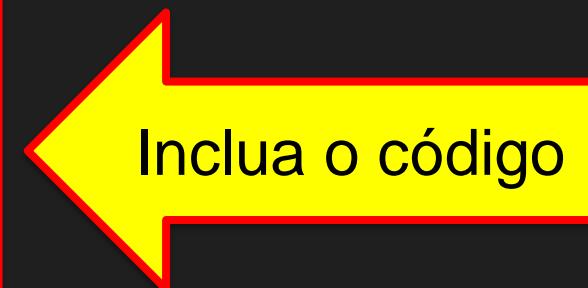
```
<body>
  <h1>Testes com a API</h1>

  <p>
    <input type="button" value="Consultar" onclick="consultar()">
    <input type="button" value="Consultar Por Código" onclick="consultarPorCodigo(1)">

    <input type="button" value="Cadastrar" onclick="cadastrar()">
    <input type="button" value="Alterar" onclick="alterar(1)">
    <input type="button" value="Excluir" onclick="excluir(1)">
  </p>

  <p id="resultado">
  </p>

  <script src="testes.js"></script>
</body>
</html>
```



Inclua o código

# Método Consultar

**Guilherme Henrique de Souza**

[guilherme.souza@etec.sp.gov.br](mailto:guilherme.souza@etec.sp.gov.br)

[guilherme.souza183@fatec.sp.gov.br](mailto:guilherme.souza183@fatec.sp.gov.br)

```
[function consultar()
{
    fetch("http://localhost:3001/alunos")
        .then(res => res.json())
}
]
```

## ■ **fetch**

- É uma função do JavaScript usada para fazer requisições HTTP.

## ■ **.then(res => res.json())**

- Essa linha espera pela resposta da requisição e a transforma em JSON.
  - **res.json()** é uma promessa que, quando resolvida, retorna o corpo da resposta no formato de objeto JavaScript.

```
function consultar()
{
    fetch("http://localhost:3001/alunos")
        .then(res => res.json())
        .then(data => {
            console.log("Alunos:", data);
            document.getElementById("resultado").innerHTML = JSON.stringify(data, null, 2);
        })
}
```

## ■ Quando os dados são recebidos:

- São impressos no console com console.log.
- São convertidos em uma string JSON e exibidos no console e no paragrafo.

```
function consultar()
{
    fetch("http://localhost:3001/alunos")
        .then(res => res.json())
        .then(data => {
            console.log("Alunos:", data);
            document.getElementById("resultado").innerHTML = JSON.stringify(data, null, 2);
        })
        .catch(err => {
            document.getElementById("resultado").innerHTML = "Erro ao obter dados da API !";
        });
}
```

- Se houver algum erro na requisição
  - O erro é mostrado no console e no paragrafo



# Método Consultar

```
function consultar()
{
    fetch("http://localhost:3001/alunos")
        .then(res => res.json())
        .then(data => {
            console.log("Alunos:", data);
            document.getElementById("resultado").innerHTML = JSON.stringify(data, null, 2);
        })
        .catch(err => {
            document.getElementById("resultado").innerHTML = "Erro ao obter dados da API !";
        });
}
```

# Testes com a API

Consultar Consultar Por Codigo Cadastrar Alterar Excluir

```
[ { "codigo": 1, "nome": "Guilherme", "cidade": "Mococa", "estado": "SP" }, { "codigo": 2, "nome": "Gustavo", "cidade": "Guaxupé", "estado": "MG" }, { "codigo": 3, "nome": "Felipe", "cidade": "Curitiba", "estado": "PR" } ]
```

# JSON.stringify()

Função utilizada do JavaScript é usada para converter objetos JavaScript em strings JSON

```
[const aluno = { nome: "Guilherme", cidade: "Mococa", estado : "SP" };

console.log( aluno );

console.log( JSON.stringify(aluno) );

console.log( JSON.stringify(aluno, null, 2) );]
```

```
{ nome: 'Guilherme', cidade: 'Mococa', estado: 'SP' }
```

```
{"nome": "Guilherme", "cidade": "Mococa", "estado": "SP"}
```

```
{
  "nome": "Guilherme",
  "cidade": "Mococa",
  "estado": "SP"
}
```



# Método Consultar Por Código

**Guilherme Henrique de Souza**

guilherme.souza@etec.sp.gov.br

guilherme.souza183@fatec.sp.gov.br



# Método Consultar Por Código

código

```
function consultarPorCodigo(código)
{
    fetch(`http://localhost:3001/alunos/${código}`)
        .then(res => res.json())
        .then(data => {
            console.log("Aluno:", data);
            document.getElementById("resultado").innerHTML = JSON.stringify(data, null, 2);
        })
        .catch(err => {
            document.getElementById("resultado").innerHTML = "Erro ao obter dados da API!";
        });
}
```

# Método Consultar Por Código



# Método Cadastrar

**Guilherme Henrique de Souza**

[guilherme.souza@etec.sp.gov.br](mailto:guilherme.souza@etec.sp.gov.br)

[guilherme.souza183@fatec.sp.gov.br](mailto:guilherme.souza183@fatec.sp.gov.br)



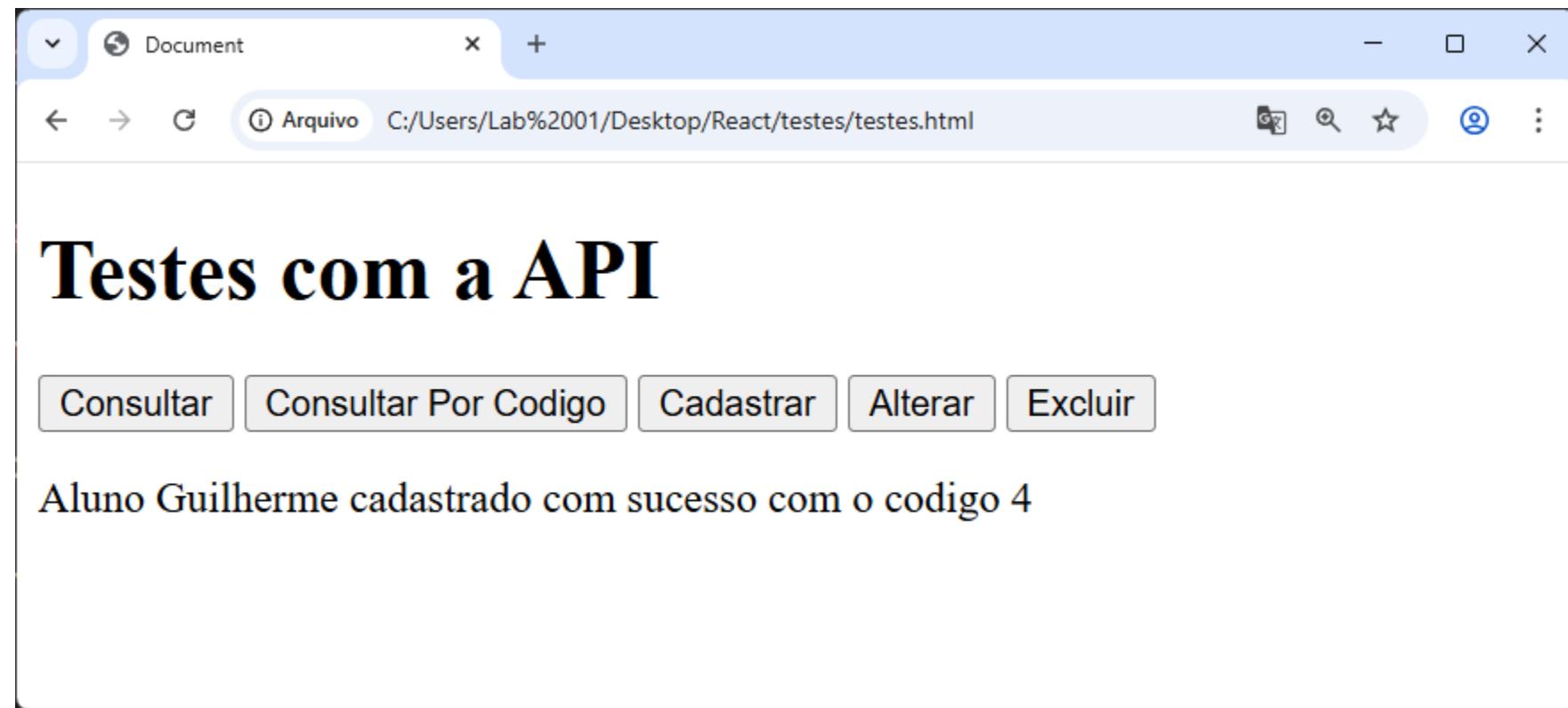
# Método Cadastrar

```
function cadastrar()  
{  
    const aluno = {  
        nome : "Guilherme",  
        cidade: "Mococa",  
        estado: "SP"  
    };  
}  
}
```

Variável para o  
aluno

```
[function cadastrar()
{
    const aluno = {
        nome : "Guilherme",
        cidade: "Mococa",
        estado: "SP"
    };

    fetch("http://localhost:3001/alunos", {
        method: "POST",
        headers: {
            "Content-Type": "application/json"
        },
        body: JSON.stringify(aluno)
    })
    .then(res => res.json())
    .then(data => {
        document.getElementById("resultado").innerHTML = data.message || data.error;
    })
    .catch(err => {
        document.getElementById("resultado").innerHTML = "Erro ao obter dados da API !";
    });
}
```



15 • select \* from alunos;

Result Grid | Filter Rows:

	codigo	nome	cidade	estado
▶	1	Guilherme	Mococa	SP
	2	Gustavo	Guaxupé	MG
	3	Felipe	Curitiba	PR
	4	Guilherme	Mococa	SP

# Testes com a API

Consultar Consultar Por Codigo Cadastrar Alterar Excluir

```
[ { "codigo": 1, "nome": "Guilherme", "cidade": "Mococa", "estado": "SP" }, { "codigo": 2, "nome": "Gustavo", "cidade": "Guaxupé", "estado": "MG" }, { "codigo": 3, "nome": "Felipe", "cidade": "Curitiba", "estado": "PR" }, { "codigo": 4, "nome": "Guilherme", "cidade": "Mococa", "estado": "SP" } ]
```

# Método Alterar

**Guilherme Henrique de Souza**

[guilherme.souza@etec.sp.gov.br](mailto:guilherme.souza@etec.sp.gov.br)

[guilherme.souza183@fatec.sp.gov.br](mailto:guilherme.souza183@fatec.sp.gov.br)

```
function alterar(codigo)
{
    const alunoAtualizado = {
        nome: "Guilherme Henrique de Souza",
        cidade: "São Paulo",
        estado: "SP"
    };

    fetch(`http://localhost:3001/alunos/${codigo}`, {
        method: "PUT", ←
        headers: {
            "Content-Type": "application/json"
        },
        body: JSON.stringify(alunoAtualizado)
    })
    .then(res => res.json())
    .then(data => {
        document.getElementById("resultado").innerHTML = data.message || data.error;
    })
    .catch(err => {
        document.getElementById("resultado").innerHTML = "Erro ao obter dados da API !";
    });
}
```





# Método Excluir

**Guilherme Henrique de Souza**

[guilherme.souza@etec.sp.gov.br](mailto:guilherme.souza@etec.sp.gov.br)

[guilherme.souza183@fatec.sp.gov.br](mailto:guilherme.souza183@fatec.sp.gov.br)

```
[function excluir(codigo)
{
    fetch(`http://localhost:3001/alunos/${codigo}`, {
        method: "DELETE"←
    })
    .then(res => res.json())
    .then(data => {
        document.getElementById("resultado").innerHTML = data.message || data.error;
    })
    .catch(err => {
        document.getElementById("resultado").innerHTML = "Erro ao obter dados da API !";
    });
}
```

