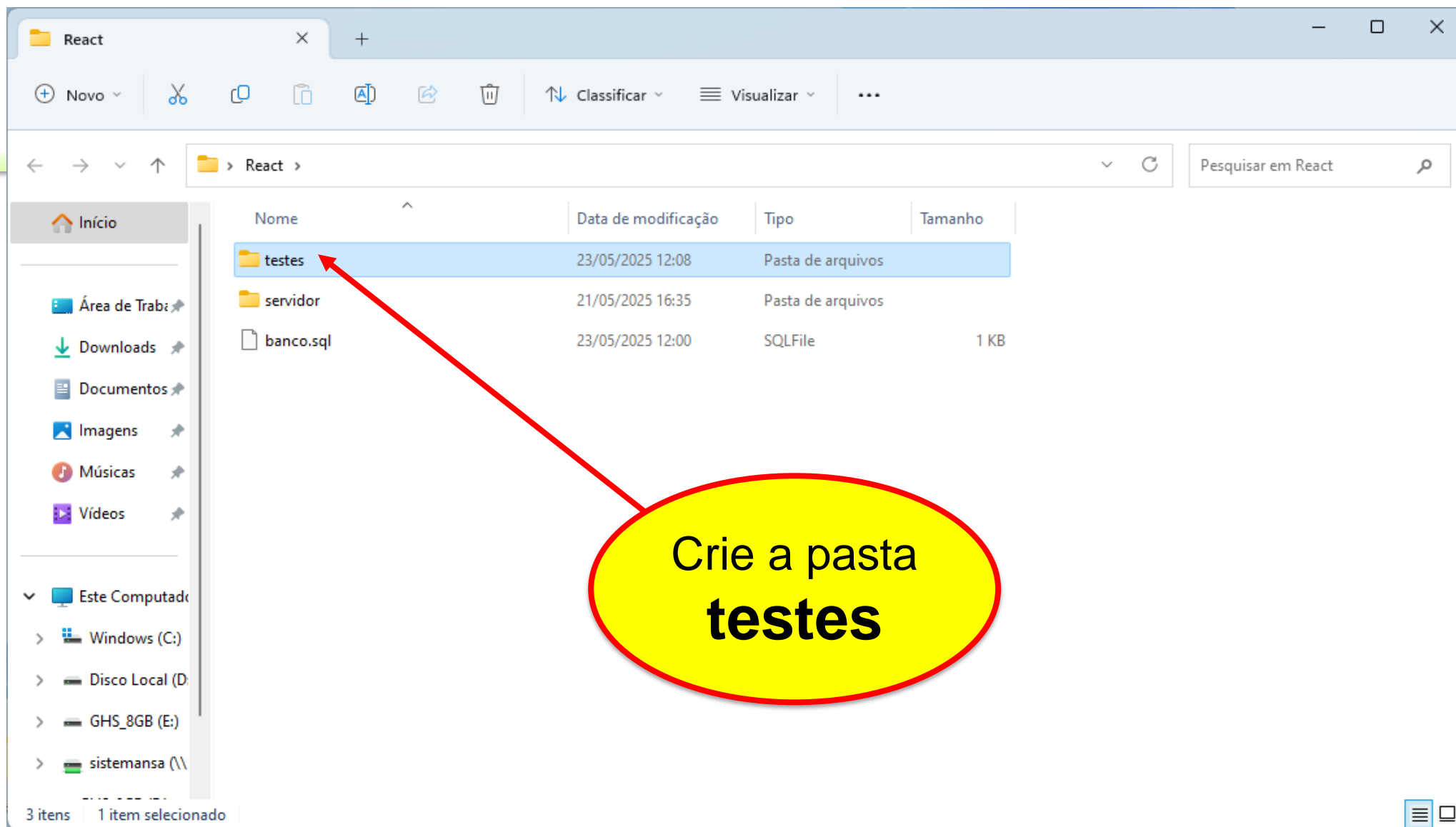


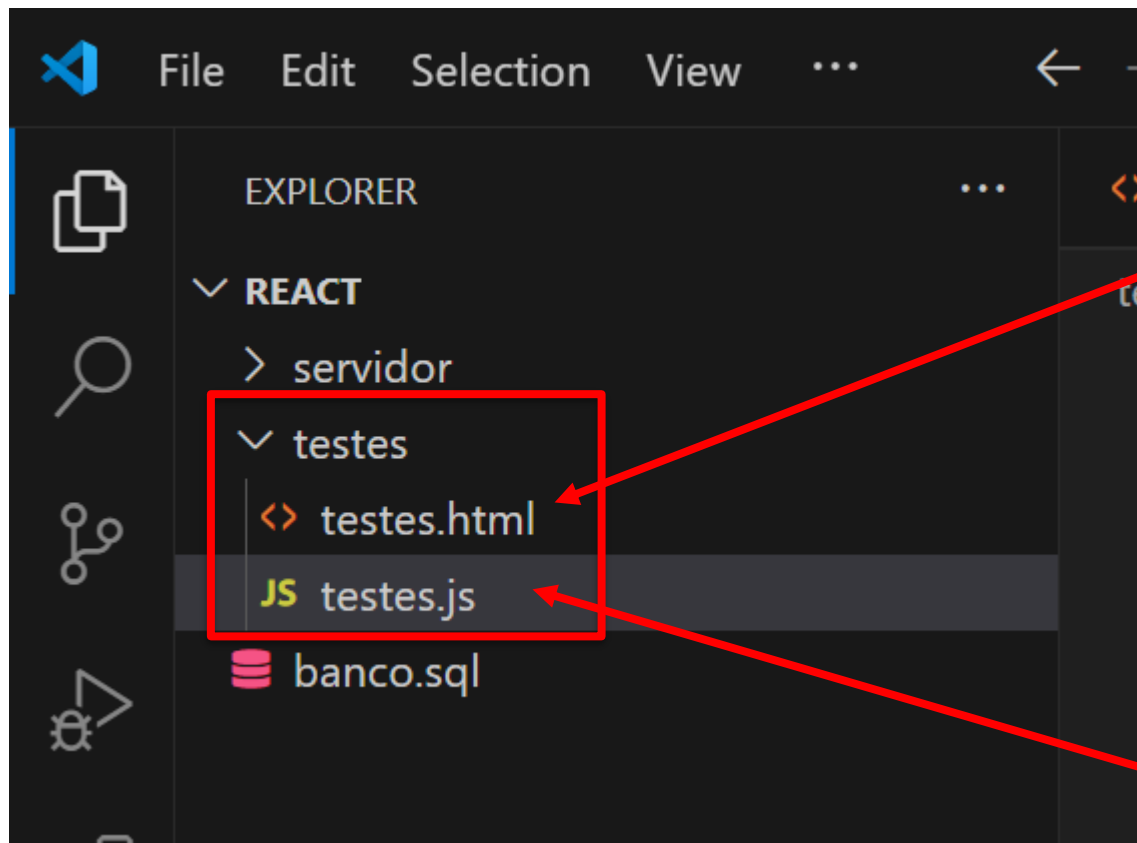
## 10-2 - Criação dos Testes em HTML e JS

**Guilherme Henrique de Souza**

guilherme.souza@etec.sp.gov.br

guilherme.souza183@fatec.sp.gov.br





Crie o arquivo  
**testes.html**

Crie o arquivo  
**testes.js**

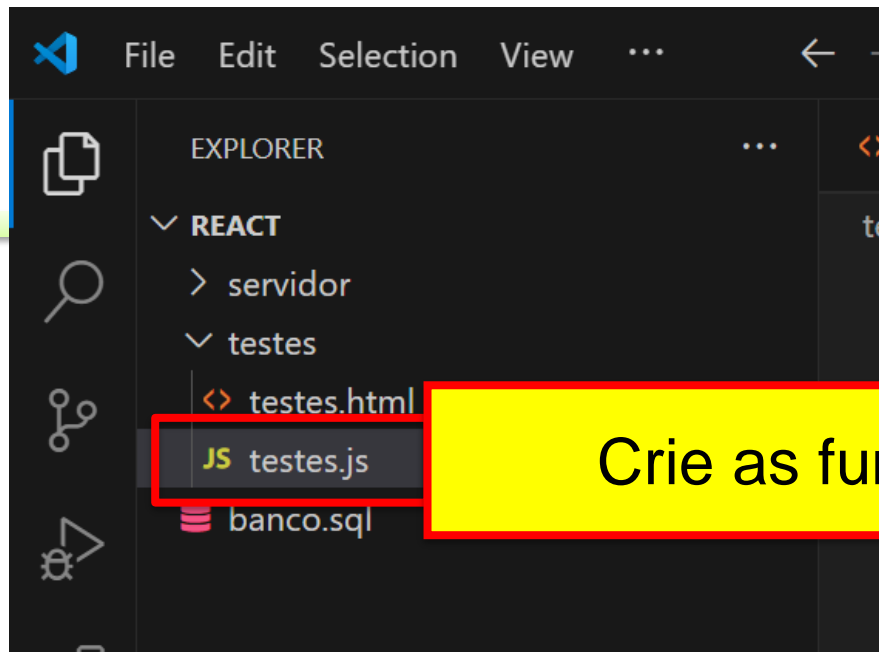
A decorative graphic on the left side of the slide, consisting of a thin green circle and a thick green bracket-like shape that frames the title.

# Funções do arquivo **testes.js**

**Guilherme Henrique de Souza**

guilherme.souza@etec.sp.gov.br

guilherme.souza183@fatec.sp.gov.br

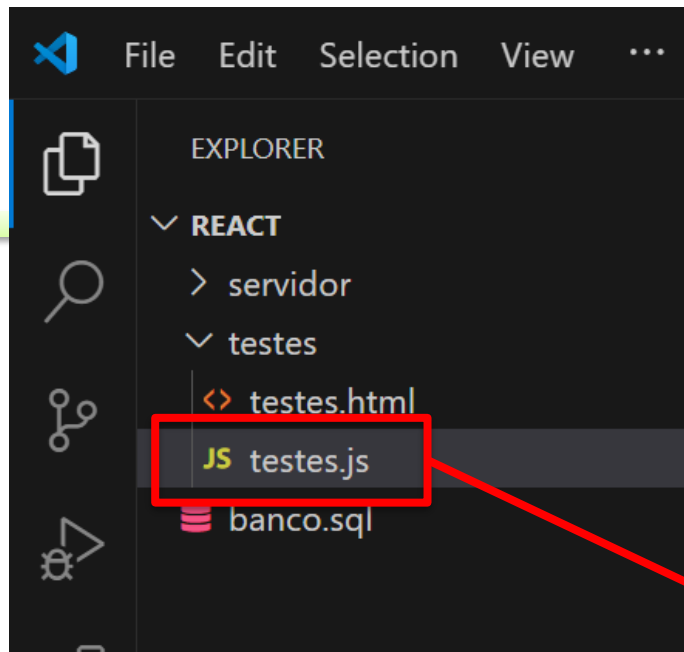


Crie as funções

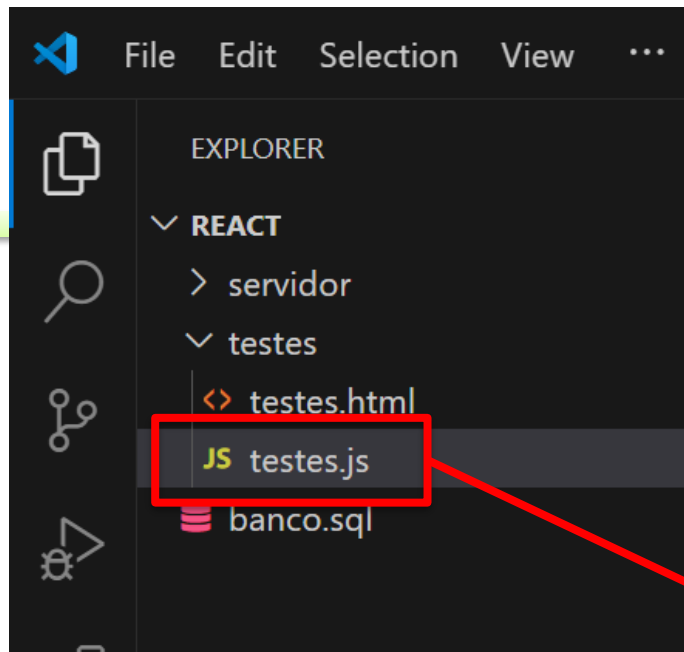
```

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

1  function consultar()
2  {
3
4  }
5
6  function consultarPorCodigo(codigo)
7  {
8
9  }
10
11 function cadastrar() |
12 {
13
14 }
15
16 function alterar(codigo)
17 {
18
19 }
20
21 function excluir(codigo)
22 {
23
24 }
    
```



```
JS testes.js X
testes > JS testes.js > cadastrar
1 function consultar()
2 {
3
4 }
5
6 function consultarPorCodigo(codigo)
7 {
8
9 }
```



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

A decorative graphic on the left side of the slide. It features a thin green circle that is partially obscured by a thick green bracket. The bracket is composed of two vertical lines and two horizontal lines at the top and bottom, forming a large '[' shape. The text 'Arquivo testes.html' is positioned to the right of the bracket, within a light green horizontal bar.

# Arquivo **testes.html**

**Guilherme Henrique de Souza**

guilherme.souza@etec.sp.gov.br

guilherme.souza183@fatec.sp.gov.br



&lt;&gt; testes.html X

testes &gt; &lt;&gt; testes.html &gt; html &gt; 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 PorCodigo" 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 PorCodigo" 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

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 parágrafo.

```
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 !";
  });
}
```

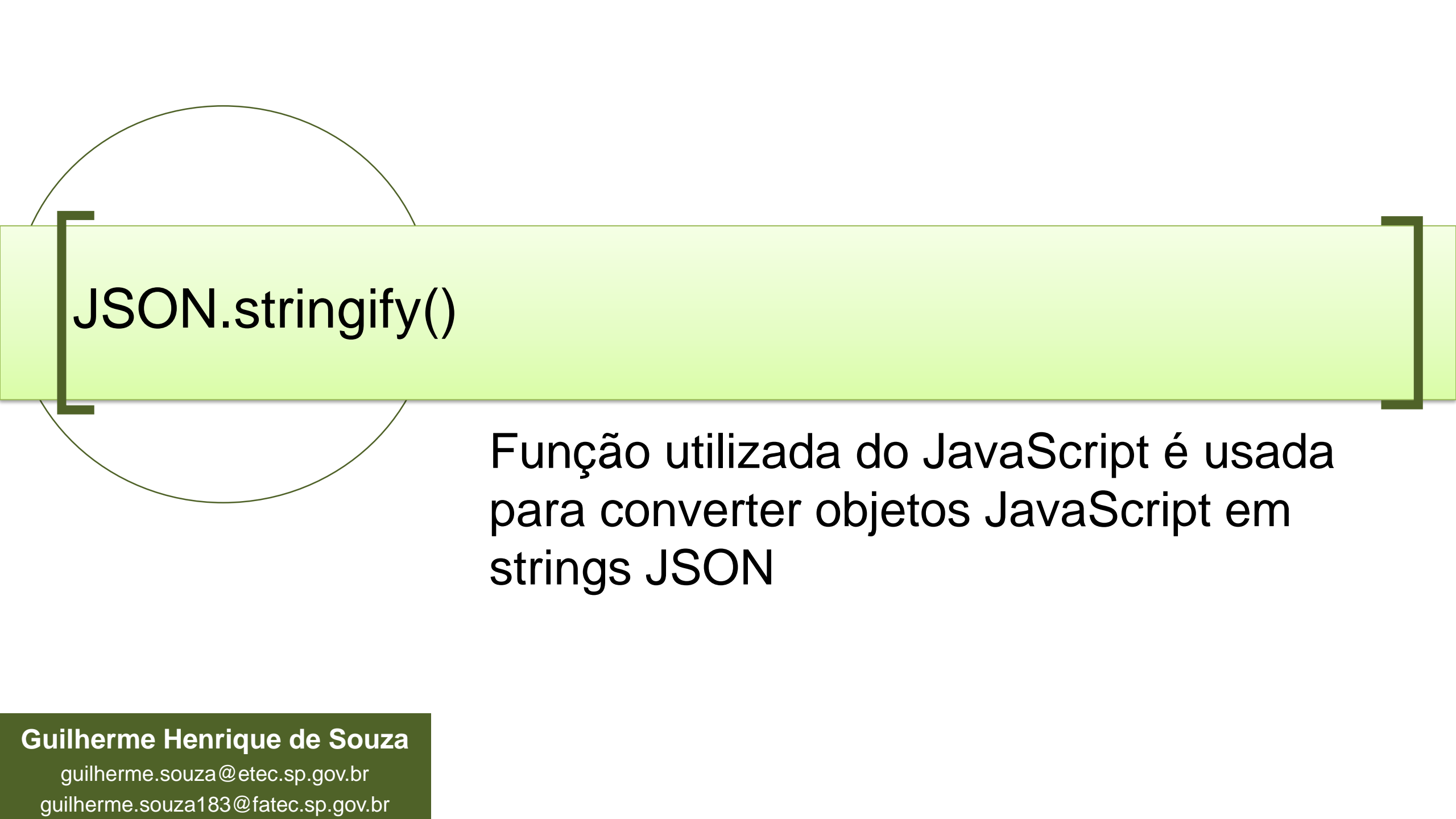
- Se houver algum erro na requisição
  - O erro é mostrado no console e no paragrafo

# 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 !";  
    });  
}
```







# JSON.stringify()

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


**Guilherme Henrique de Souza**

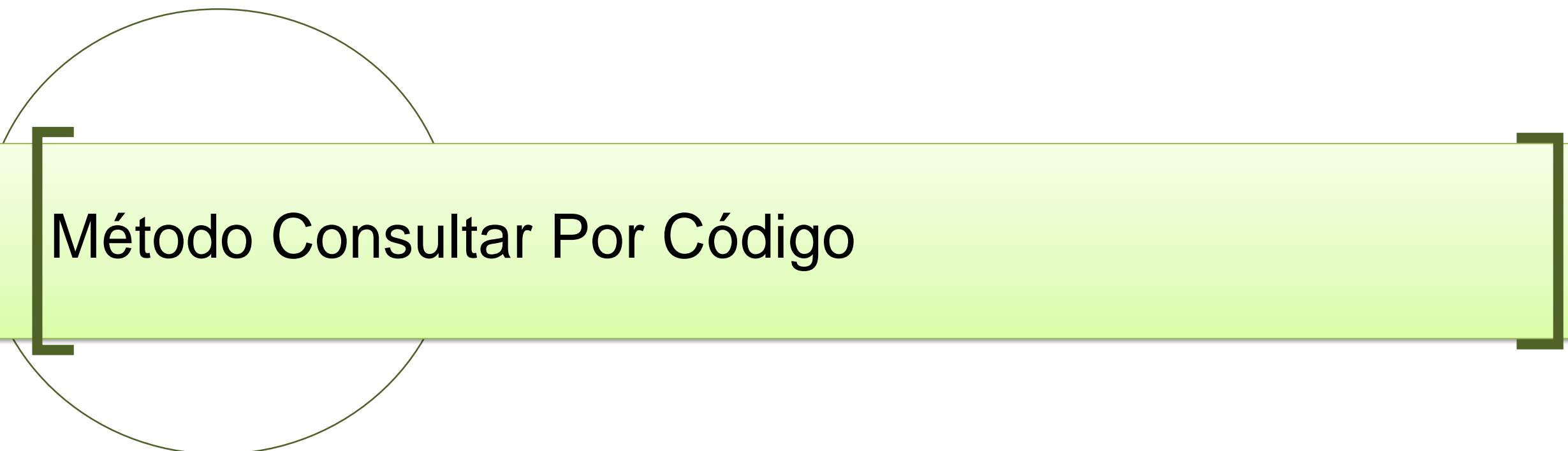
guilherme.souza@etec.sp.gov.br

guilherme.souza183@fatec.sp.gov.br

```
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"  
}
```



A decorative graphic on the left side of the slide, featuring a thin green circle and a thick green bracket-like shape that frames the title.

# 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(codigo)
{
  fetch(`http://localhost:3001/alunos/${codigo}`)
  .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

guilherme.souza183@fatec.sp.gov.br

# Método Cadastrar

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

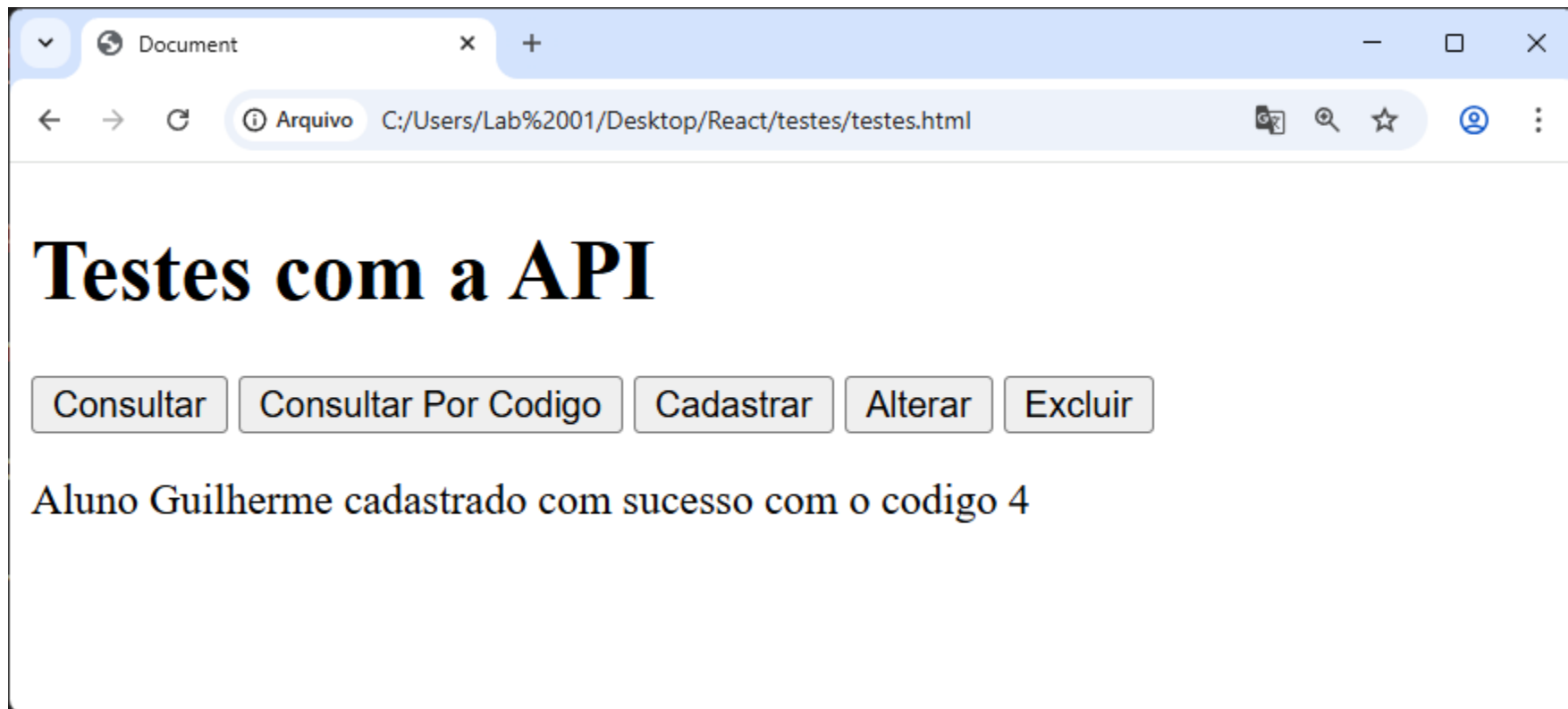


Variável para o  
aluno



```
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;`

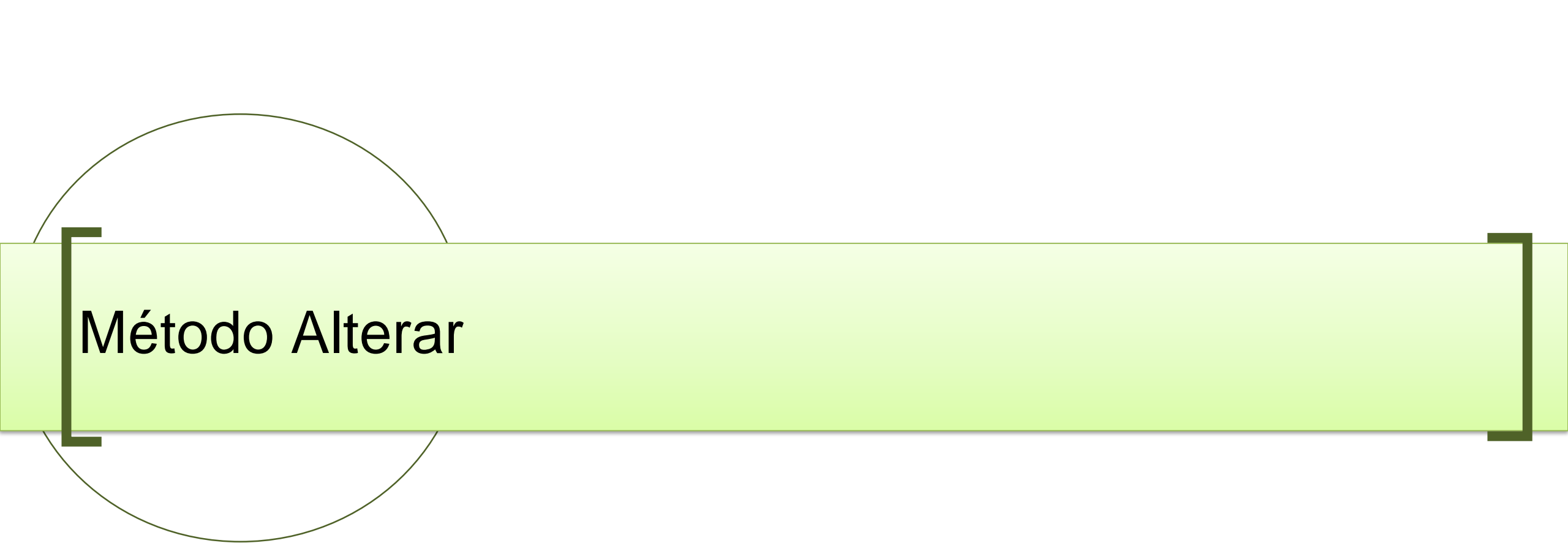
Result Grid



Filter Rows:

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





# Método Alterar

**Guilherme Henrique de Souza**

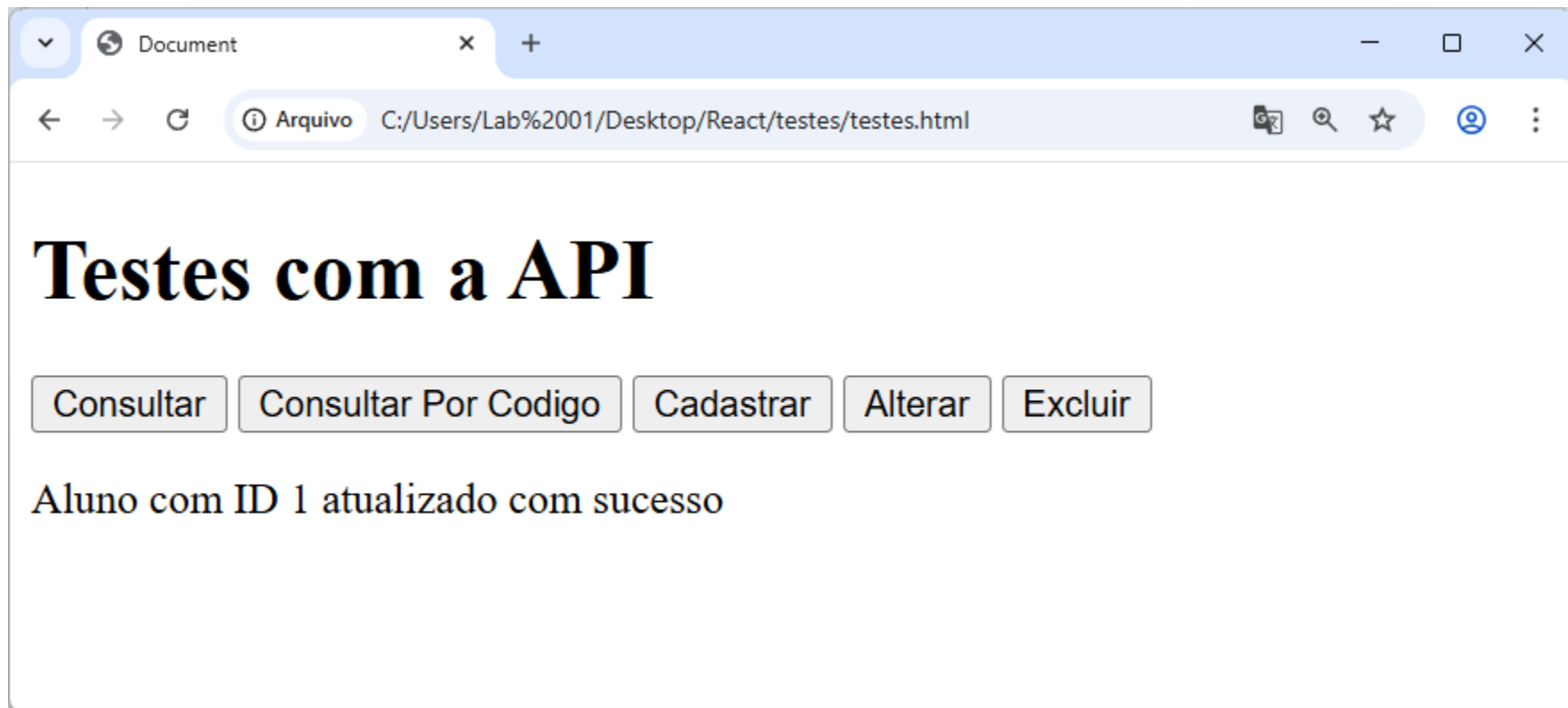
guilherme.souza@etec.sp.gov.br

guilherme.souza183@fatec.sp.gov.br

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

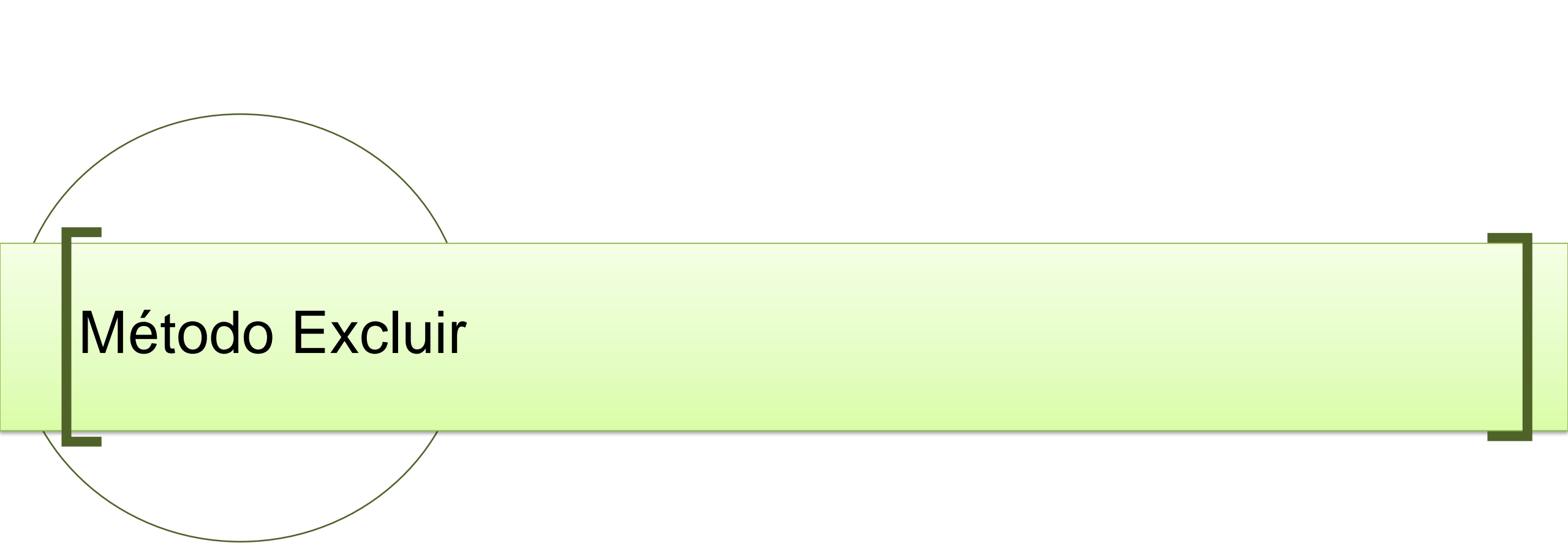
  fetch(`http://localhost:3001/alunos/${codigo}`, {
    method: "PUT",
    headers: {
      "Content-Type": "application/json"
    },
    body: JSON.stringify(alunoAtualizado)
  })
  .then(res => res.json())
  .then(data => {
    document.getElementById("resultado").innerHTML = data.message || data.error;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











# Método Excluir

**Guilherme Henrique de Souza**

guilherme.souza@etec.sp.gov.br

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 !";
  });
}
```

