

HTML

Aula 5


Vamos continuar com o
exercício do currículo.
Veremos nesta aula:

```
<form>  
<input>  
<textarea>  
<select>  
<label>  
<button>
```

Curriculum Vitae

[Currículo Cadastro](#)

Dados Pessoais



Lula Molusco

Endereço: Rua Concha 122
Fenda do Biquini, *Oceano Pacífico*

[Fenda do Biquíni](#) é uma formação natural subaquática localizada no Oceano Pacífico.

Experiência Profissional

1. **Siri Cascudo - 2015**
Trabalho de caixa, na maior parte do tempo.
2. **Siri Cascudo - 2014**
Chef (ajudando Bob Esponja na cozinha)
3. **Fenda do Biquini - 2011**
Músico amador (clarinete)

"Lula Molusco, eu usei sua clarineta pra desentupir minha privada!"
Bob Esponja

Formação Acadêmica

Formulário

*Mas para que
serve um
formulário?*



1

Tudo começa no browser. É comum chamarmos essa camada de front-end.

Front-end





Front-end

Back-end

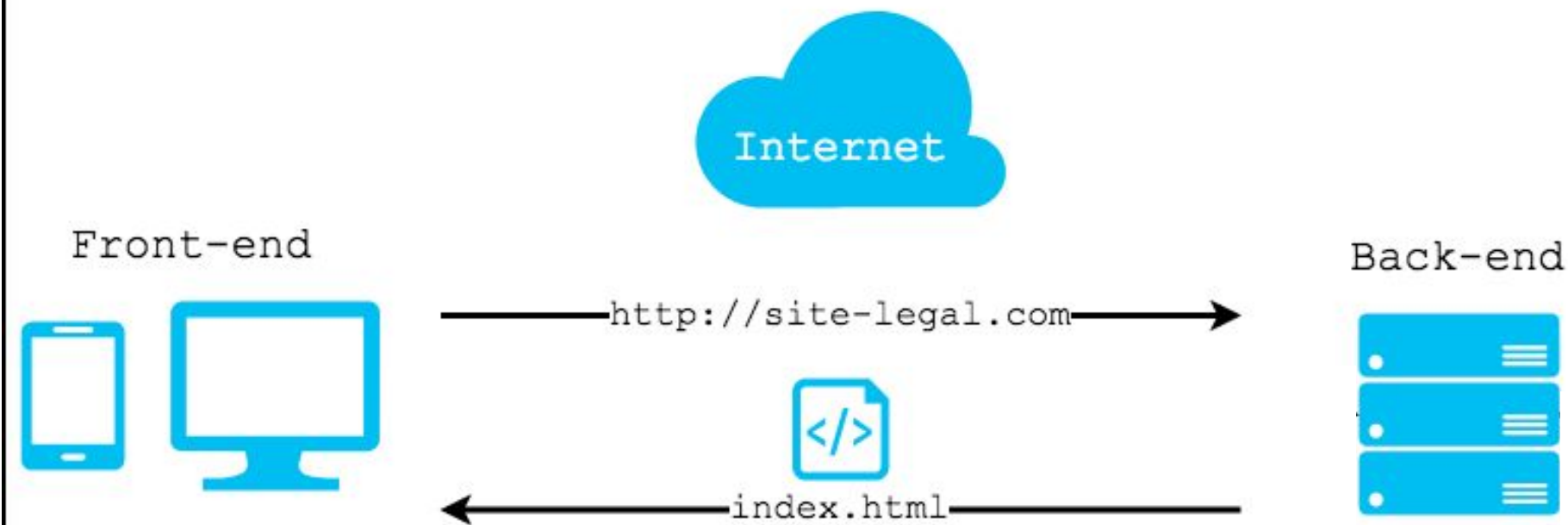


`http://site-legal.com` →



2

Ao digitar um endereço, o browser faz uma solicitação a um servidor HTTP, geralmente chamado de *back-end*.
Exemplos: Apache, Nginx, IIS



3

O servidor HTTP retorna um arquivo HTML, que é exibido pelo browser ao usuário.



Front-end

Back-end

`http://site-legal.com`



`index.html`



Formulário



Usuário

4

Às vezes essas páginas possuem campos de entrada que permitem o usuário enviar dados ao servidor. Este comportamento é realizado pelo formulário.



Front-end

Back-end



`http://site-legal.com`



`index.html`



`Form Data`



Formulário



Usuário

5

Após o preenchimento, o browser envia os dados do formulário para o back-end.

Dados recebidos no back-end em 05/08/2018 11:10:42

Nome: Guilherme
Sobrenome: Pontes

Veremos mais detalhes sobre isso ao longo da disciplina!

Elements Console Sources Network Performance Memory Application Security Audits

View: [Icons] Group by frame Preserve log Disable cache Offline Online

Filter [] Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other

Name x Headers Preview Response Timing

index.php

- General
- Response Headers (5)
- Request Headers (12)
- Form Data view source view URL encoded
 - nome: Guilherme
 - sobrenome: Pontes

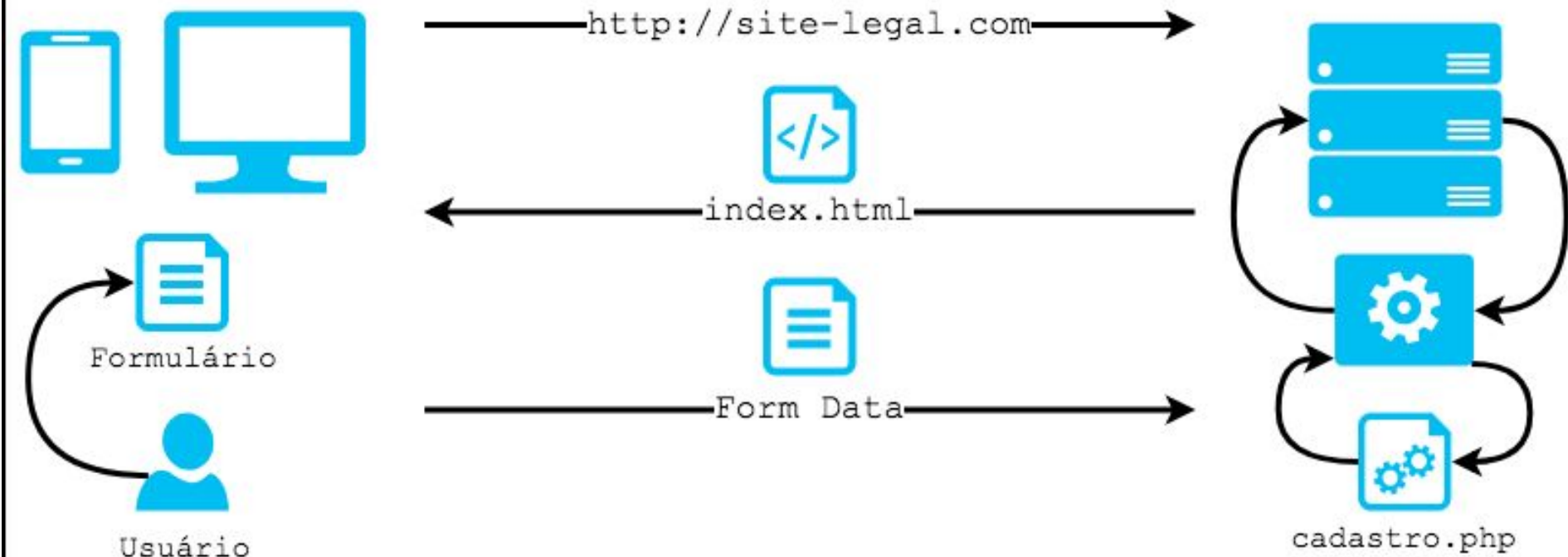
1 requests | 504 B transferred | Finish: 7 ms | D...

6

O servidor HTTP repassa os dados do formulário para um servidor de aplicação, que processa a requisição e retorna os dados. Exemplos de servidor de aplicação: PHP, Java (WebLogic, Tomcat), C# (IIS), Python (Gunicorn), NodeJS, entre outros.

Front-end

Back-end



7

Por fim, o servidor HTTP retorna o resultado para o browser.

Internet

Front-end

Back-end



`http://site-legal.com`



`index.html`



Formulário

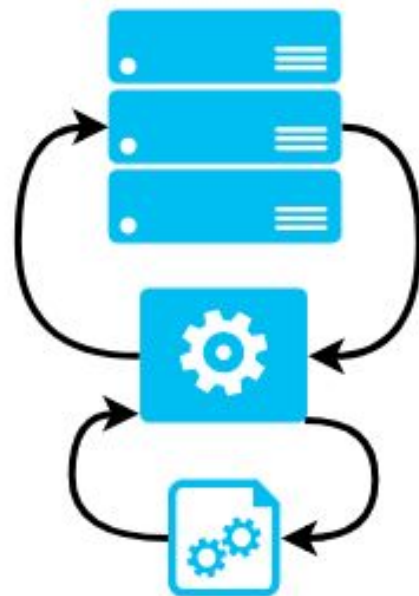


Form Data




Usuário

Resultado




`cadastro.php`

Tag <form> (exe1/index.php)

<form> 

<form>


Esta tag é responsável por agrupar todos os elementos pertencentes ao formulário. Além dessa responsabilidade semântica, é ele que organiza o envio dos dados.

</form> 

Tag <form> (exe1/index.php)

```
<form>
```

Nome:

```
<input type="text" /> 
```

```
</form>
```


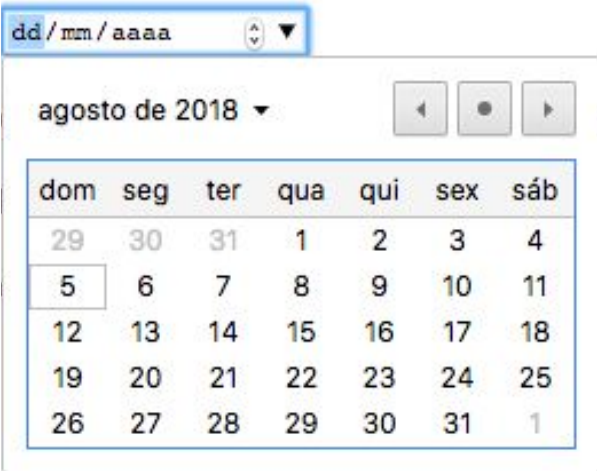
```
<input>
```

Esta tag define um campo de entrada para o usuário. Ela possui vários tipos. Neste exemplo estamos utilizando o tipo "text", que cria um campo de texto.


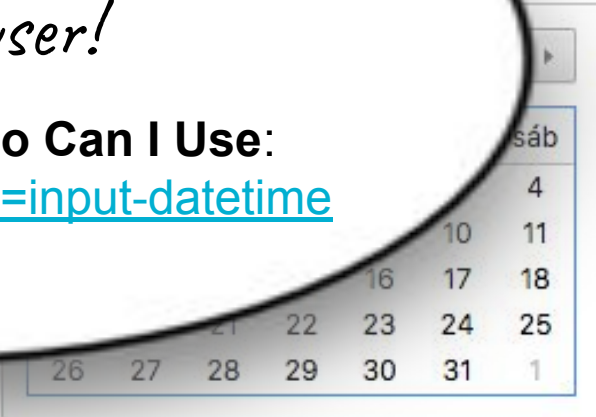
Tipos da tag <input> (exe1/tipos.html)

Tipo	Descrição	Exemplo
text	Utilizado para informar textos	<input type="text" value="Guilherme"/>
password	Utilizado para informar senha	<input type="password" value="....."/>
<div>HTML5</div> email	Próprio para informar um email	<div><input type="text" value="lgapontes@gmail.com"/><div>Configurações de preenchimento automático</div></div>
<div>HTML5</div> number	Para campos numéricos. É possível definir <i>ranges</i> (exemplos em breve)	<input type="text" value="10"/>
file	<div><input type="button" value="Escolher arquivo"/> Nenhum arquivo selecionado</div> Utilizado para selecionar arquivos para upload. Veremos no futuro outros detalhes para utilização deste campo.	

Tipos da tag <input> (exe1/tipos.html)

Tipo	Descrição	Exemplo
 date	Utilizado para informar datas	
checkbox	Útil para valores booleanos	<input type="checkbox"/> Campo checkbox
radio	Utilizado para indicar um conjunto de valores	<input type="radio"/> Campo radio 1 <input type="radio"/> Campo radio 2

Tipos da tag <input> (exe1/tipos.html)

Tipo	Descrição	Exemplo
 date	Utilizado para capturar uma data e hora	
checkbox	Campos booleanos	<input type="checkbox"/> Campo checkbox
radio	Campos para indicar um conjunto de opções	<input type="radio"/> Campo radio 1 <input type="radio"/> Campo radio 2

CUIDADO:


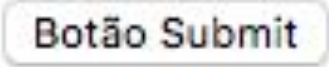
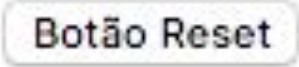
Nem todos os tipos do HTML 5 são cross-browser!

Na dúvida, procure no Can I Use:

<https://caniuse.com/#feat=input-datetime>



Tipos da tag <input> (exe1/tipos.html)

Tipo	Descrição	Exemplo
<code>button</code>	Botão comum, utilizado para ações que devem ser programadas via JavaScript. Veremos uso prático no futuro.	
<code>submit</code>	Botão mais importante do formulário. Este é o que realiza a ação de enviar os dados.	
<code>reset</code>	Botão para limpar todos os dados do formulário.	

Veremos outros exemplos adiante. Para ver outros campos, acesse:
https://www.w3schools.com/html/html_form_input_types.asp

Tag <form> (exe1/index.php)

```
<form>
```

Nome:

```
<input type="text" name="nome" />
```



Mas como o servidor vai identificar quais campos foram enviados?

A tag <input> possui um atributo chamado *name*. Através dele que o servidor vai identificar os campos.

```
</form>
```

Tag <form> (exe1/index.php)

```
<form>
  Nome:
  <input type="text" name="nome" />
  <br />
  Sobrenome:
  <input type="text" name="sobrenome" />
  <br />
  <input type="submit" value="Enviar" />
</form>
```

Neste exemplo estamos enviando dois campos de texto para serem processados no servidor. **Vamos enviar este formulário!**

Não se preocupe com o código PHP por enquanto. Veremos em detalhes no futuro.

127.0.0.1:8080/exe1/index.php

Nome:

Sobrenome:

Elements Console Sources **Network** Performance Memory Application Security Audits

☐ View: ☐ Group by frame ☐ Preserve log ☐ Disable cache ☐ Offline Online

Filter ☐ Hide data URLs **All** XHR JS CSS Img Media Font Doc WS Manifest Other

Recording network activity...

Perform a request or hit **R** to record the reload.

Dados recebidos no back-end em 05/08/2018 12:55:28

Nome: Guilherme
Sobrenome: Pontes

Elements | Console | Sources | Network | Performance | Memory | Application | Security | Audits

View: [Icons] ☐ Group by frame ☐ Preserve log ☐ Disable cache ☐ Offline Online

Filter ☐ Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other

Name	Headers	Preview	Response	Timing
index.php?nome=Guilherme&sobrenome=P...	<div><div>General</div><div>Response Headers (5)</div><div>Request Headers (8)</div><div>Query String Parameters<div>view sourceview URL encoded</div><div>nome: Guilherme</div><div>sobrenome: Pontes</div></div></div>			

1 requests | 504 B transferred | Finish: 7 ms | D...

Dados recebidos no back-end em 05/08/2018 12:55:28

Nome: Guilherme
Sobrenome: Pontes



Mas por que os dados estão sendo enviados pela URL?

index.php?nome=Guilherme&sobrenome=P...

General

Response Headers (5)

Request Headers (8)

Query String Parameters

view source

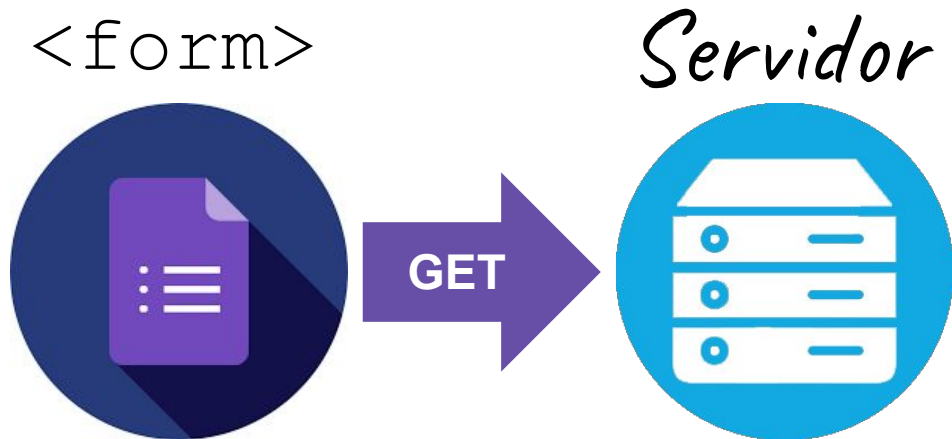
view URL encoded

nome: Guilherme

sobrenome: Pontes

GET

O método HTTP GET é utilizado para obter recursos do servidor. Os dados são enviados pela URL (Query String).



Notes on GET:



w3schools.com

- Appends form-data into the URL in name/value pairs
- The length of a URL is limited (about 3000 characters)
- Never use GET to send sensitive data! (will be visible in the URL)
- Useful for form submissions where a user want to bookmark the result
- GET is better for non-secure data, like query strings in Google

POST

O método HTTP POST é utilizado para enviar dados para o servidor. Ele é adequado para envio de formulários.

`<form>`



Servidor



Notes on POST:




w3schools.com

- POST has no size limitations, and can be used to send large amounts of data.
- Form submissions with POST cannot be bookmarked



ATENÇÃO: O método POST mantém os dados escondidos mas **NÃO** deve ser utilizado sem HTTPS para envio de dados sensíveis.



Tag <form> (exe2/index.php)

```
<form method="POST">  
  Nome:   
  <input type="text" name="nome" />  
  <br />  
  Sobrenome:  
  <input type="text" name="sobrenome" />  
  <br />  
  <input type="submit" value="Enviar" />  
</form>
```

Através do atributo *method* nós podemos definir se o <form> enviará seus dados por POST ou GET.

Nome:

Sobrenome:

Elements Console Sources Network Performance Memory Application Security Audits							⋮	✕
🔴 🔵 🎥 🔍 View: 📄 📄 <input type="checkbox"/> Group by frame <input type="checkbox"/> Preserve log <input type="checkbox"/> Disable cache <input type="checkbox"/> Offline Online ▼								
Filter <input type="checkbox"/> Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other								
Name	Status	Type	Initiator	Size	Time	Waterfall		
 index.php	200	document	Other	619 B	6 ms			

Dados recebidos no back-end em 05/08/2018 03:07:39

Nome: Guilherme
Sobrenome: Pontes

Elements Console Sources Network Performance Memory Application Security Audits

View: [Icons] [Icons] Group by frame Preserve log Disable cache Offline Online

Filter [] Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other

Name	Headers	Preview	Response	Timing
index.php	<div><div>General</div><div>Response Headers (5)</div><div>Request Headers (12)</div><div>Form Data view source view URL encoded</div></div>			

1 requests | 504 B transferred | Finish: 7 ms | D...

Tag <form> (exe2/index.php)

```
<form method="POST">  
  Nome:  
  <input type="text" name="nome" />  
  <br />  
  Sobrenome:  
  <input type="text" name="sobrenome" />  
  <br />  
  <input type="submit" value="Enviar" />  
</form>
```



Quando este formulário for enviado, qual é o arquivo no servidor responsável por recebê-lo?



Tag <form> (exe2/index.php)

```
<form method="POST">  
  Nome:  
  <input type="text" name="nome" />  
  <br />  
  Sobrenome:  
  <input type="text" name="sobrenome" />  
  <br />  
  <input type="text" name="email" />  
</form>
```

Quando não especificamos o "destino" do envio do formulário, os dados serão recebidos pelo próprio arquivo no servidor (neste caso, index.php)



Tag <form> (exe3/index.html)

```
<form method="POST" action="back-end.php">
```

Nome:

```
<input type="text" name="nome" />
```

```
<br />
```

Sobrenome:

```
<input type="text" name="sobrenome" />
```

```
<br />
```

```
<input type="submit" value="Enviar" />
```

```
</form>
```



Através do atributo *action* nós podemos definir o receptor do envio do formulário.

Neste caso, estamos enviando para um script chamado *back-end.php*

Exercício em Sala

Na pasta *exe-amigos* disponível nos arquivos da Aula 5, há um script PHP chamado *cadastro.php* capaz de receber os campos *nome* e *nascimento* através do método POST.

Construa a página *index.html* com um formulário capaz de enviar esses dados.

Não se preocupe em executar o servidor PHP por enquanto. Veremos isso em detalhes no futuro. Caso queira executar o exercício, recomendo publicá-lo no diretório disponível na Internet.

Exercício em Sala



A screenshot of a web browser window. The title bar shows a tab titled 'Cadastro de Amigos x' and a user profile 'Luis Guilherme...'. The address bar contains the URL '127.0.0.1:8080/exe-amigos...' with a star icon for bookmarks and a menu icon. The page content features a heading 'Cadastro de Amigos' followed by a horizontal line. Below the line, there are two input fields: 'Nome:' with the value 'Maria' and 'Data de nascimento:' with the value '01/01/2016'. A 'Salvar' button is positioned below the date field.

Cadastro de Amigos

Nome:

Data de nascimento:

Resolvendo o Exercício

```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3   <head>
4     <meta charset="utf-8" />
5     <title>Cadastro de Amigos</title>
6   </head>
7   <body>
8     <h3>Cadastro de Amigos</h3>
9     <hr>
10    <form method="POST" action="cadastro.php">
11      Nome:
12      <input type="text" name="nome" />
13      <br />
14      Data de nascimento:
15      <input type="date" name="nascimento" />
16      <br />
17      <input type="submit" value="Salvar" />
18    </form>
19  </body>
20 </html>
```

Resolvendo o Exercício



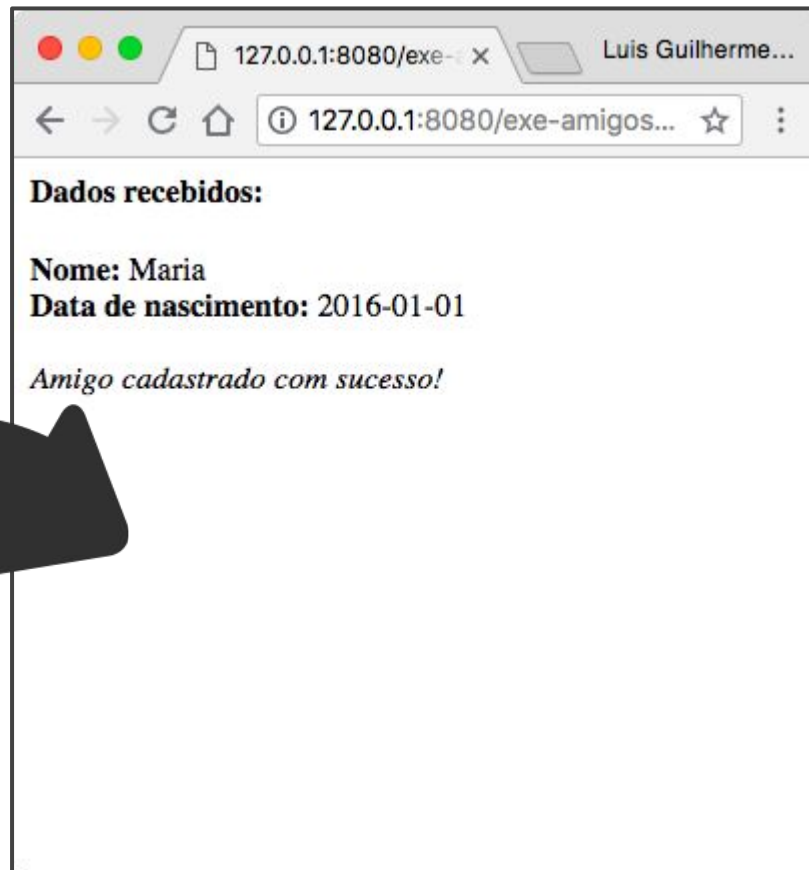
Cadastro de Amigos x Luis Guilherme...

127.0.0.1:8080/exe-amigos... ☆ ⋮

Cadastro de Amigos

Nome:

Data de nascimento:



127.0.0.1:8080/exe- x Luis Guilherme...

127.0.0.1:8080/exe-amigos... ☆ ⋮

Dados recebidos:

Nome: Maria

Data de nascimento: 2016-01-01

Amigo cadastrado com sucesso!

Validação realizada no servidor!



Cadastro de Amigos x Luis Guilherme...

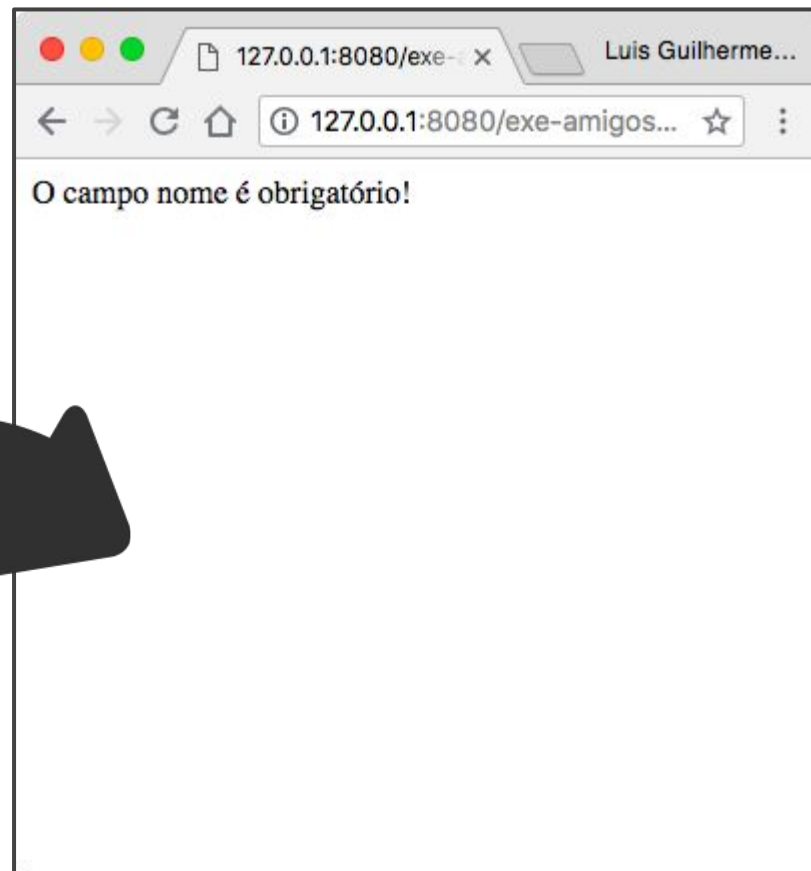
127.0.0.1:8080/exe-amigos... ☆ ⋮

Cadastro de Amigos

Nome:

Data de nascimento:

Salvar

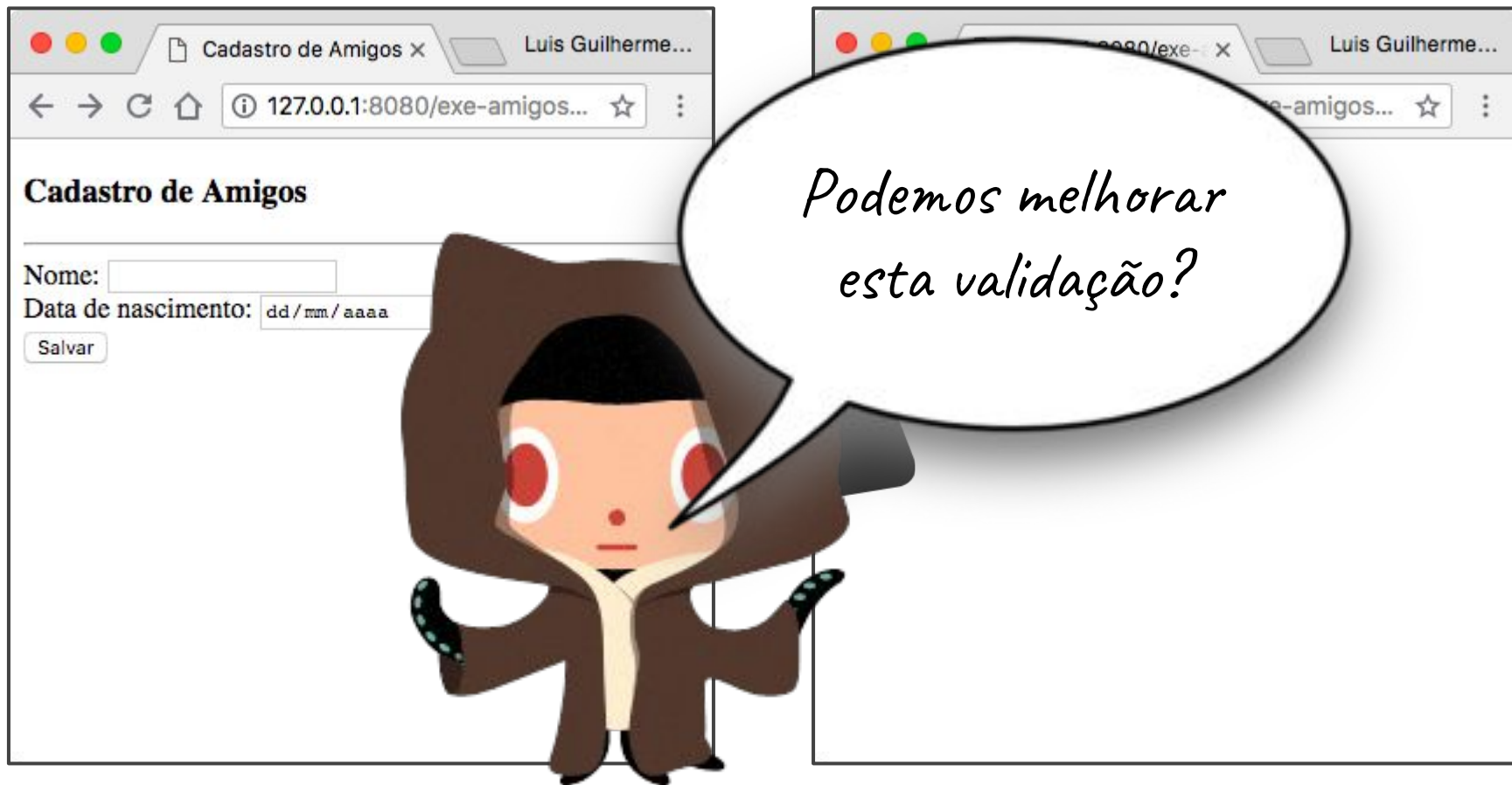


127.0.0.1:8080/exe-amigos... x Luis Guilherme...

127.0.0.1:8080/exe-amigos... ☆ ⋮

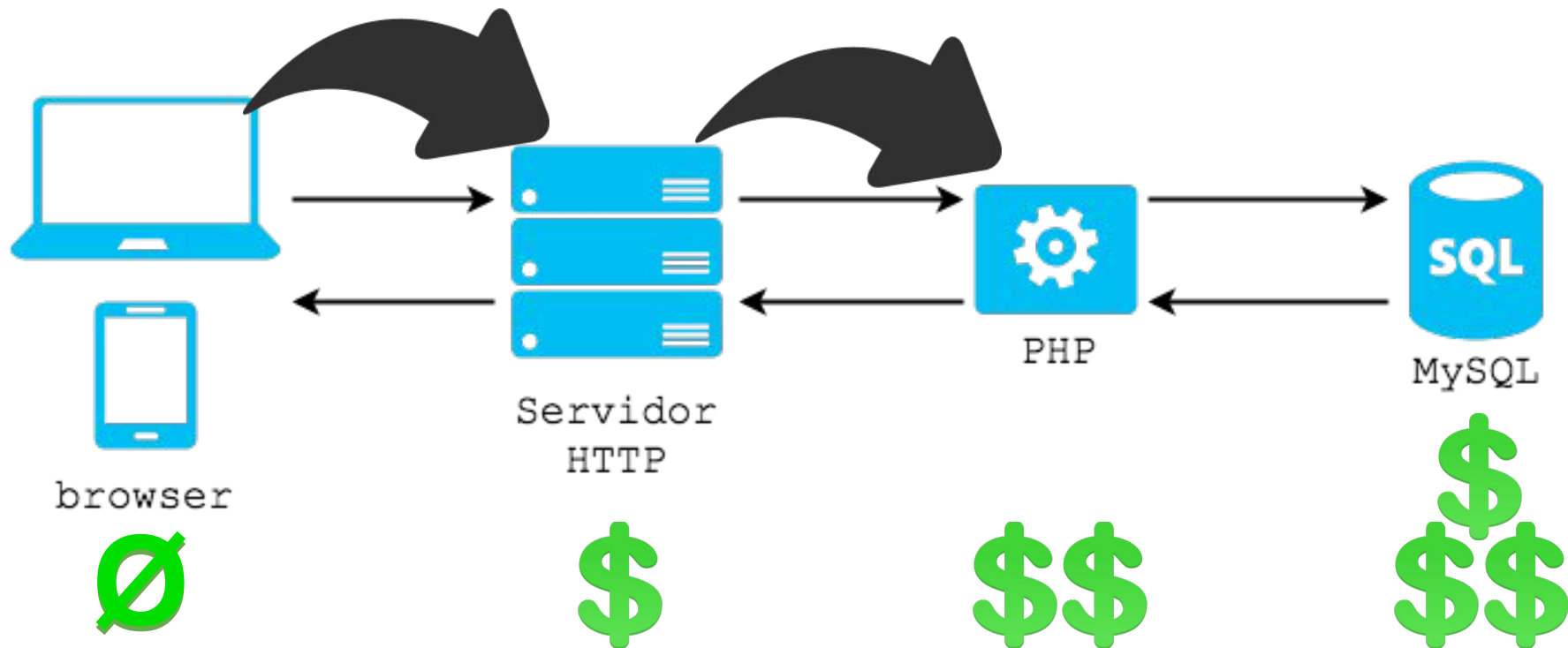
O campo nome é obrigatório!

Validação realizada no servidor!



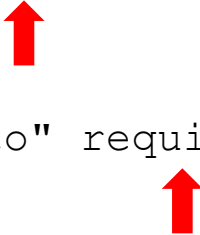
Validação realizada no servidor!

A validação de dados no servidor é necessária, porém existem boas práticas para evitar requisições desnecessárias ao servidor.



Atributo *required* (exe4/index.html)

```
<form method="POST" action="cadastro.php">  
  Nome:  
  <input type="text" name="nome" required />  
  <br />  
  Data de nascimento:  
  <input type="date" name="nascimento" required />  
  <br />  
  <input type="submit" value="Salvar" />  
</form>
```



O atributo *required* (do HTML 5) serve para indicar obrigatoriedade no preenchimento dos campos. Ao tentar enviar um campo requerido sem preencher, o navegador vai exibir uma mensagem.

Validação realizada no cliente!



Cadastro de Amigos x Luis Guilherme...

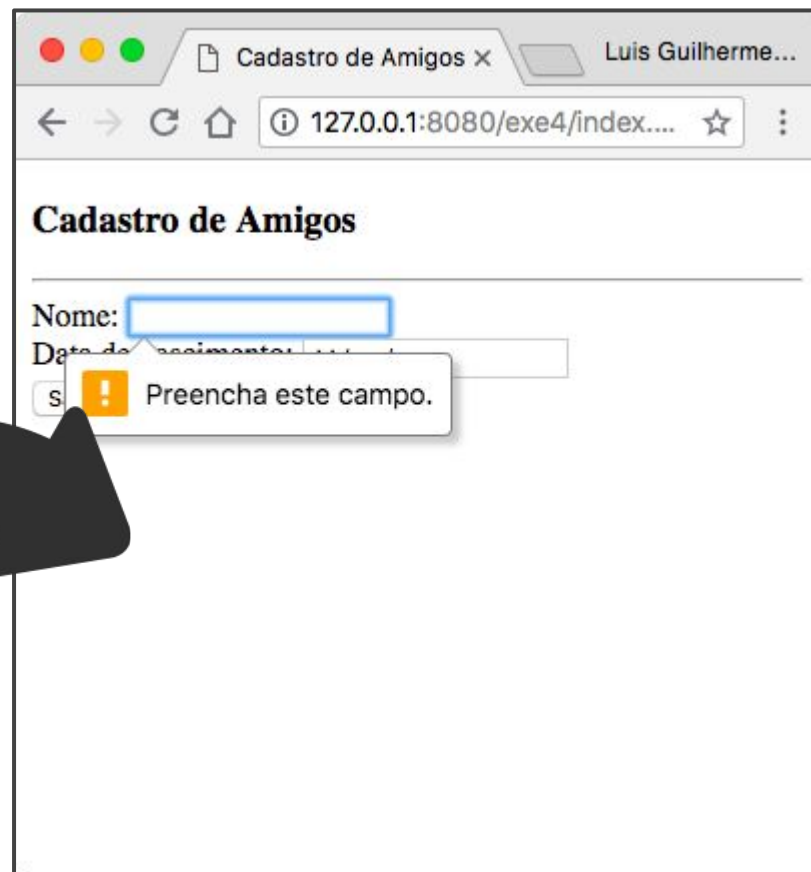
127.0.0.1:8080/exe-amigos...

Cadastro de Amigos

Nome:

Data de nascimento:

Salvar



Cadastro de Amigos x Luis Guilherme...

127.0.0.1:8080/exe4/index....

Cadastro de Amigos

Nome:

Data de nascimento:

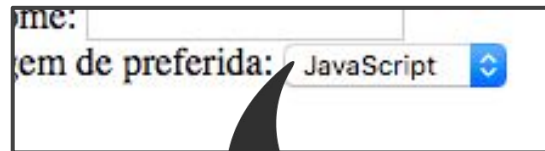
Salvar

! Preencha este campo.

Tags <select> and <option> (exe5/index.html)

A tag <select> permite acrescentar um ComboBox no HTML. Os elementos são inseridos pela tag <option>

```
<select name="linguagem">
  <option value="javascript">JavaScript</option>
  <option value="java">Java</option>
  <option value="c_sharp">C#</option>
  <option value="c_plusplus">C++</option>
  <option value="php">PHP</option>
  <option value="python">Python</option>
  <option value="kotlin">Kotlin</option>
  <option value="swift">Swift</option>
  <option value="objective_c">Objective C</option>
  <option value="groovy">Groovy</option>
  <option value="outras">outras</option>
</select>
```



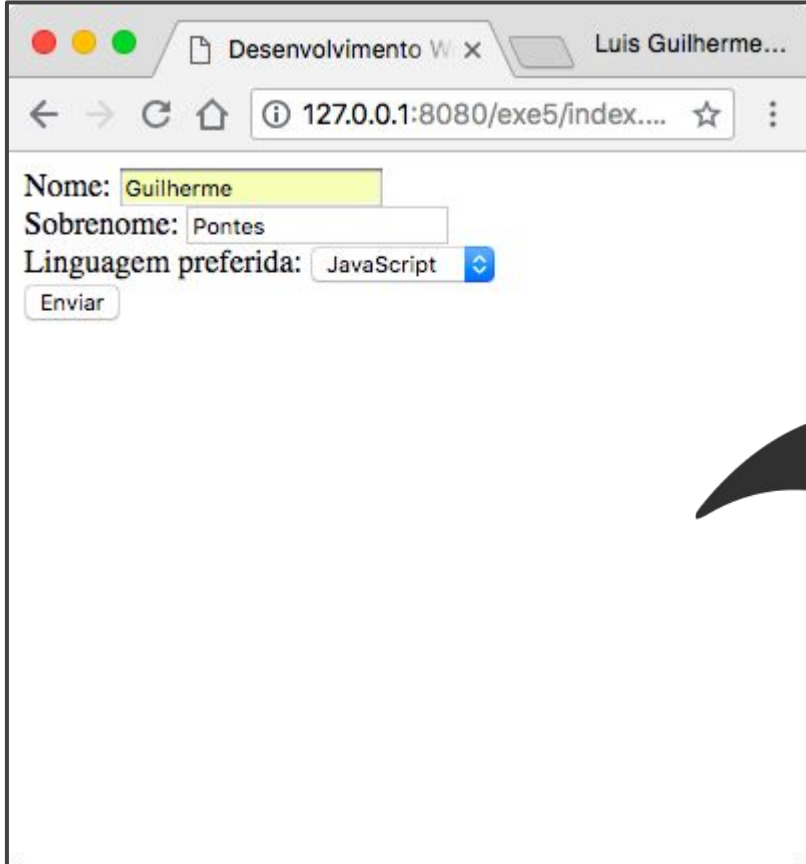
me:
em de preferida: JavaScript



me:
em de preferida

- ✓ JavaScript
- Java
- C#
- C++
- PHP
- Python
- Kotlin
- Swift
- Objective C
- Groovy
- outras

Tags <select> and <option> (exe5/index.html)



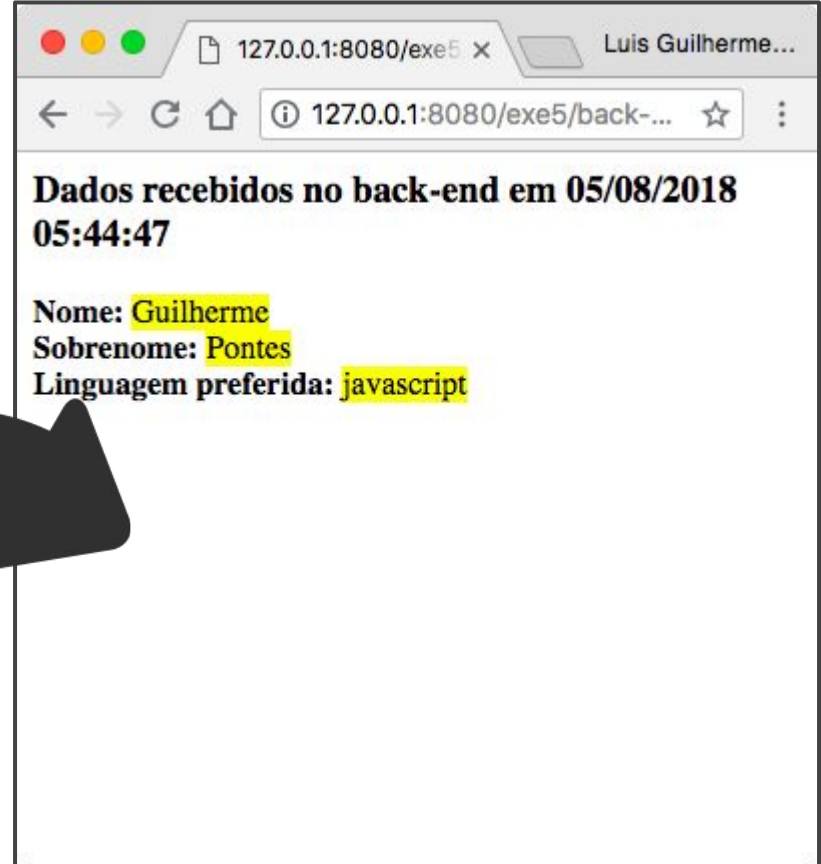
Desenvolvimento W x Luis Guilherme...

127.0.0.1:8080/exe5/index....

Nome:

Sobrenome:

Linguagem preferida: JavaScript



127.0.0.1:8080/exe5/back-...

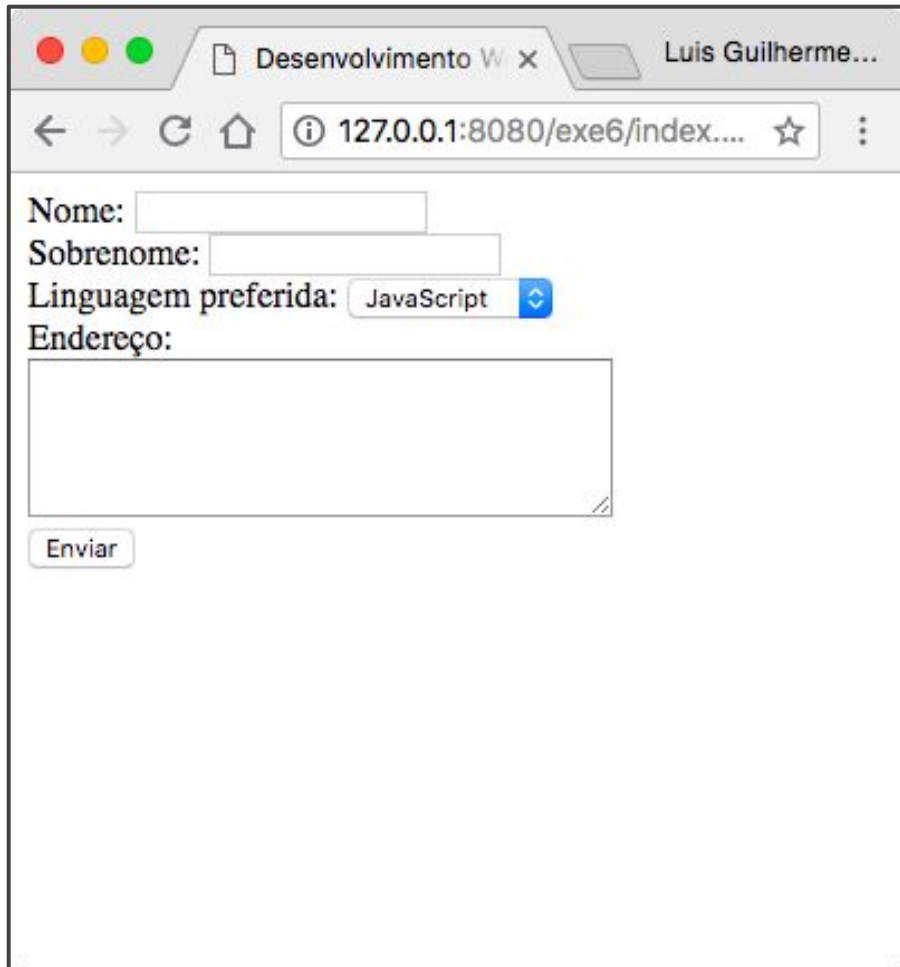
Dados recebidos no back-end em 05/08/2018 05:44:47

Nome: **Guilherme**
Sobrenome: **Pontes**
Linguagem preferida: **javascript**

Tag <textarea> (exe6/index.html)

A tag <textarea> pode ser utilizada para textos grandes. Os atributos *cols* e *rows* servem, respectivamente, para definir o tamanho de colunas e linhas.

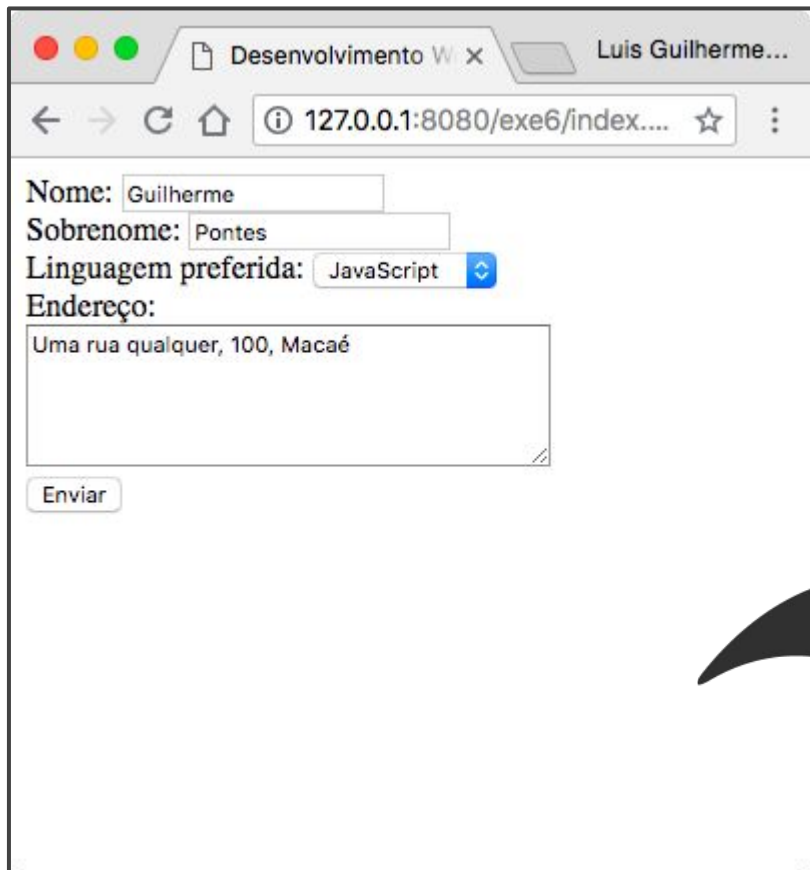
```
Endereço:<br />
<textarea
  name="endereco"
  cols="40"
  rows="5">
</textarea>
<br />
```



The screenshot shows a web browser window with the title 'Desenvolvimento W x' and the user 'Luis Guilherme...'. The address bar displays '127.0.0.1:8080/exe6/index....'. The form contains the following elements:

- Nome:
- Sobrenome:
- Linguagem preferida:
- Endereço:
- Enviar:

Tags <textarea> (exe6/index.html)



Desenvolvimento W x Luis Guilherme...

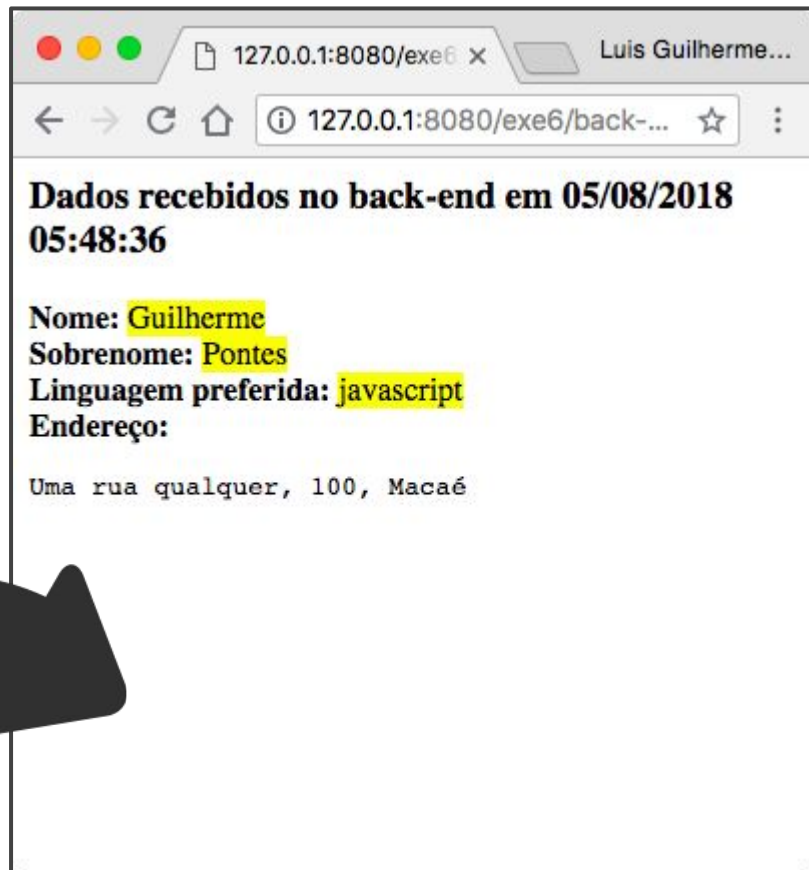
127.0.0.1:8080/exe6/index....

Nome:

Sobrenome:

Linguagem preferida:

Endereço:



127.0.0.1:8080/exe6/back-...

Dados recebidos no back-end em 05/08/2018 05:48:36

Nome: **Guilherme**

Sobrenome: **Pontes**

Linguagem preferida: **javascript**

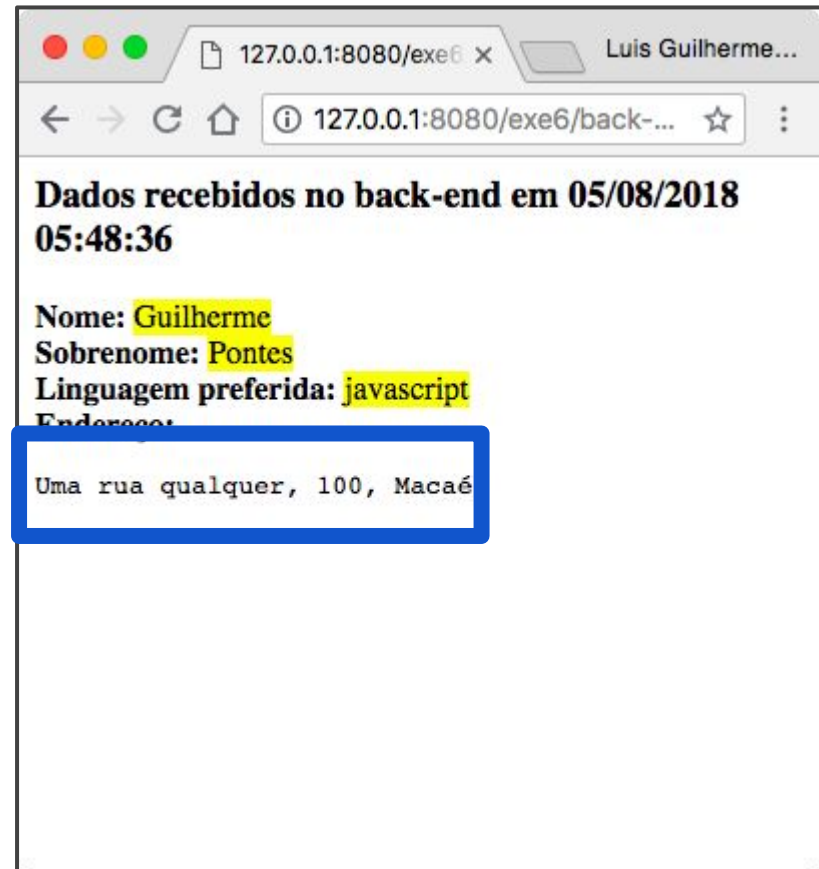
Endereço:

Uma rua qualquer, 100, Macaé

Tags <textarea> (exe6/index.html)

Neste exemplo estamos utilizando a tag <pre>, que exibe texto pré-formatado com uma fonte de espaçamento fixo (fonte courier).

Formulários reais devem utilizar vários campos distintos para cadastro de endereço.



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8080/exe6'. The page content includes a title 'Dados recebidos no back-end em 05/08/2018 05:48:36'. Below the title, there are four lines of text: 'Nome: Guilherme', 'Sobrenome: Pontes', 'Linguagem preferida: javascript', and 'Endereço:'. The 'Endereço:' label is partially obscured by a blue rectangular box. Inside this box, the text 'Uma rua qualquer, 100, Macaé' is displayed in a monospaced font, demonstrating the use of the <pre> tag for pre-formatting text.

Boas Práticas para Formulários

Tag <label> (exe7/index.html)

```
<label for="nome">Nome:</label>  
<input type="text" id="nome" name="nome" />  
<br />
```



Desenvolvimento W x Luis Guilherme...

127.0.0.1:8080/exe7/index....

Nome:

Sobre:

Linguagem preferida: JavaScript

Endereço:

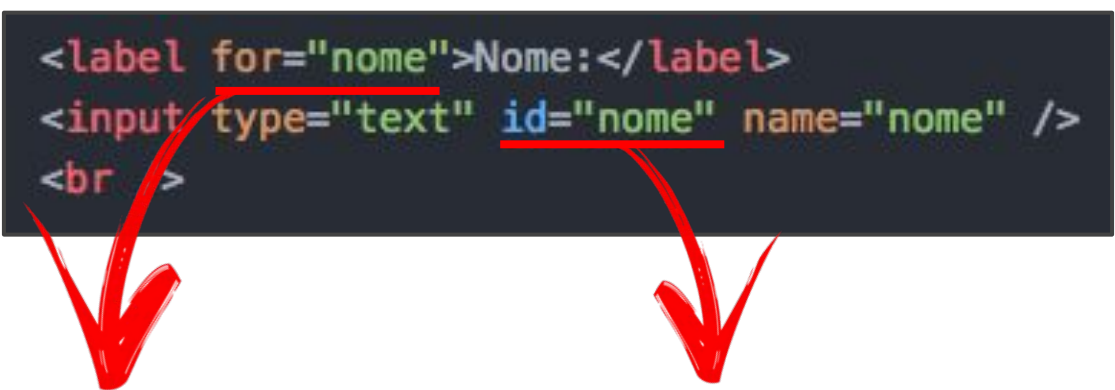
Enviar

O <label> é uma tag semântica utilizada exclusivamente para indicar os textos associados aos campos de entrada do formulário.

Além da vantagem semântica, ele oferece uma usabilidade útil aos usuários: ao clicar sobre uma <label>, o campo ao qual ela pertence recebe o foco.

Tag <label> (exe7/index.html)

```
<label for="nome">Nome:</label>  
<input type="text" id="nome" name="nome" />  
<br />
```



O atributo *for* do <label> deve apontar para o ID da tag HTML.

O campo ID das tags cria uma identificação única na página HTML. Ele é muito útil para organização dos itens no formulário (com suas labels) e para obtenção dos elementos via JavaScript.



Atenção: a estilização CSS via ID deve ser desencorajada!! Veremos detalhes no futuro...

Atributos *tabindex* e *autofocus* (exe8/index.html)

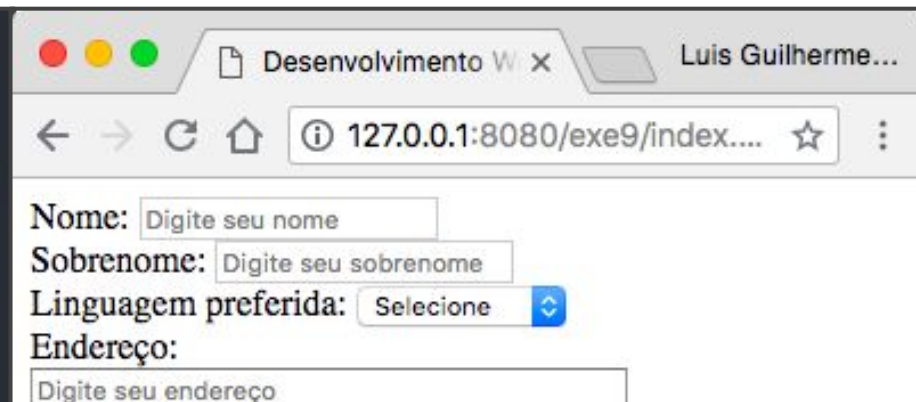
```
<form method="POST" action="back-end.php">
  <label for="nome">Nome:</label>
  <input type="text" id="nome" name="nome" tabindex="1" autofocus />
  <br />
  <label for="sobrenome">Sobrenome:</label>
  <input type="text" id="sobrenome" name="sobrenome" tabindex="2" />
  <br />
  <label for="linguagem">Linguagem preferida:</label>
  <select id="linguagem" name="linguagem" tabindex="3" >=
  <br />
  <label for="endereco">Endereço:</label><br />
  <textarea
    id="endereco"
    name="endereco"
    cols="40"
    rows="5"
    tabindex="4">
  </textarea>
  <br />
  <input type="submit" value="Enviar" tabindex="5" />
</form>
```

O atributo *tabindex* serve para ordenar os elementos do HTML pelo TAB do teclado. O primeiro índice é 1.

O *autofocus* serve para definir o campo que ganhará o foco quando o usuário abrir a página. Se houver mais de um, o primeiro que estiver qualificado com *autofocus* ganhará.

Atributo *placeholder* (exe9/index.html)

```
<label for="nome">Nome:</label>
<input
  type="text" id="nome" name="nome"
  tabindex="1" autofocus
  placeholder="Digite seu nome" />
<br />
```



A screenshot of a web browser window. The address bar shows the URL '127.0.0.1:8080/exe9/index....'. The page content displays a form with the following fields: 'Nome:' with a text input containing the placeholder 'Digite seu nome'; 'Sobrenome:' with a text input containing the placeholder 'Digite seu sobrenome'; 'Linguagem preferida:' with a dropdown menu showing 'Selecione'; and 'Endereço:' with a text input containing the placeholder 'Digite seu endereço'.

O atributo *placeholder* serve para deixar um "texto" auxiliar dentro dos campos. Quando o usuário começar a digitar, este texto deixa de ser exibido e dá lugar ao valor inserido. Note que ele possui um estilo padrão com fonte ligeiramente mais clara do que o real valor dos campos.



É uma boa prática utilizar o *placeholder* para otimizar a organização dos itens em uma página responsiva. Veremos no futuro técnicas para esconder as tags `<label>` quando o dispositivo for mobile.

Atributo *placeholder* (exe9/index.html)


```
<label for="linguagem">Linguagem preferida:</label>
<select id="linguagem" name="linguagem" tabindex="3">
  <option value="" disabled selected>Selecione</option>
  <option value="javascript">JavaScript</option>
  <option value="java">Java</option>
  <option value="c_sharp">C#</option>
  <option value="c_plusplus">C++</option>
  <option value="php">PHP</option>
  <option value="python">Python</option>
  <option value="kotlin">Kotlin</option>
  <option value="swift">Swift</option>
  <option value="objective_c">Objective-C</option>
  <option value="groovy">Groovy</option>
  <option value="outras">outras</option>
</select>
<br />
```

O elemento `<select>` não aceita o atributo *placeholder*. Para resolver isso, nós podemos criar um `<option>` qualificado com os atributos *disabled* e *selected* e um *value* em branco.

Linguagem preferida:

Atributo *placeholder* (exe9/index.html)

```
<label for="endereco">Endereço:</label><br />
<textarea
  id="endereco" name="endereco"
  cols="40" rows="5" tabindex="4"
  placeholder="Digite seu endereço"></textarea>
<br />
```



Endereço:

Digite seu endereço

Cuidado ao definir a tag `<textarea>`. Ela recebe o espaçamento e as quebra de linhas que existirem no HTML. Para evitar isso, feche-a logo após a tag de abertura.

Tag <button> (exe10/index.html)

```
<input  
  type="submit"  
  value="Enviar" />
```

Botões padrões do HTML.

VS

```
<button type="submit">  
  Enviar  
</button>
```

Botão cujo conteúdo pode ser organizado dentro das tags de abertura e fechamento. Além de ser mais semântico, a combinação com imagens é facilitada.

```
<button type="submit" tabindex="5">  
  <span>Enviar</span>  
    
</button>
```



Exercício em Sala

No arquivo *cadastro.html* do currículo, crie um formulário conforme a imagem do próximo slide. Regras:

- Enviar dados para um script chamado *inserir-contato.php*
- Método POST
- Utilize *tabindex* e *autofocus* (no primeiro campo)
- O atributo *name* dos campos deve ser *nome*, *email*, *hamburguer* e *termos*
- Com exceção dos termos, todos devem ser obrigatórios
- Utilize a tag <button>



Cadastro de Contatos

[Currículo Cadastro](#)

Nome:

E-mail:

Qual você preferi?

- ☐ Hamburger de siri
- ☐ Hamburger de lagosta
- ☐ Hamburger de camarão

☐ Aceita os termos?

Salvar


```
<form action="inserir-contato.php" method="POST">
  <label for="nome">Nome:</label>
  <input type="text" id="nome" name="nome" tabindex="1" required autofocus />
  <br /><br />
  <label for="email">E-mail:</label>
  <input type="email" id="email" name="email" tabindex="2" required />
  <br /><br />
  <strong>Qual você preferi?</strong><br />
  <input type="radio" id="siri" name="hamburger" value="siri" tabindex="3" required />
  <label for="siri">Hamburger de siri</label>
  <input type="radio" id="lagosta" name="hamburger" value="lagosta" tabindex="4" />
  <label for="lagosta">Hamburger de lagosta</label>
  <input type="radio" id="camarao" name="hamburger" value="camarão" tabindex="5" />
  <label for="camarao">Hamburger de camarão</label>
  <br /><br />
  <input type="checkbox" id="termos" name="termos" tabindex="6" />
  <label for="termos">Aceita os termos?</label>
  <br /><br />
  <button type="submit">Salvar</button>
</form>
```

Resolvendo o Exercício



A screenshot of a web browser window. The address bar shows the URL '127.0.0.1:8080/vitae-resolv...'. The page title is 'Cadastro de Contatos'. Below the title is a link 'Currículo Cadastro'. The form contains the following fields: 'Nome:' with the value 'Guilherme', 'E-mail:' with the value 'lgapontes@gmail.com', and a section 'Qual você preferi?' with three radio button options: 'Hamburguer de siri' (selected), 'Hamburguer de lagosta', and 'Hamburguer de camarão'. There is also a checkbox 'Aceita os termos?' which is checked. At the bottom is a 'Salvar' button.

Desenvolvimento W x Luis Guilherme...

127.0.0.1:8080/vitae-resolv... ☆

Cadastro de Contatos

[Currículo Cadastro](#)

Nome:

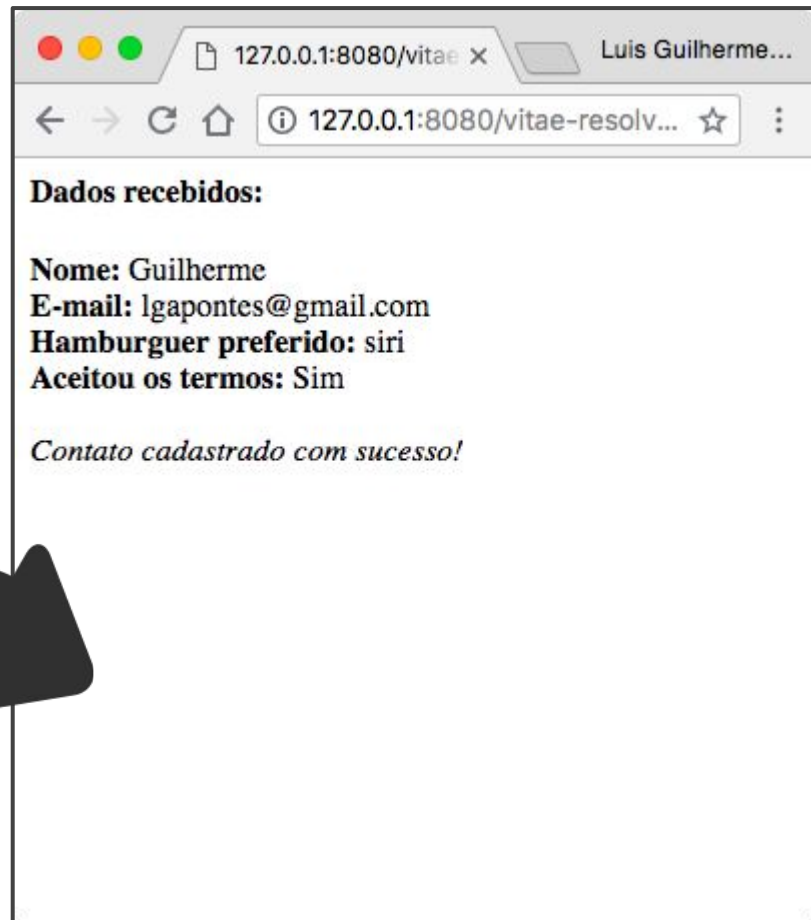
E-mail:

Qual você preferi?

- ☒ Hamburguer de siri
- ☐ Hamburguer de lagosta
- ☐ Hamburguer de camarão

☒ Aceita os termos?

Salvar



A screenshot of a web browser window showing the result of the registration. The address bar shows the URL '127.0.0.1:8080/vitae-resolv...'. The page displays the received data: 'Nome: Guilherme', 'E-mail: lgapontes@gmail.com', 'Hamburguer preferido: siri', and 'Aceitou os termos: Sim'. Below this is a message 'Contato cadastrado com sucesso!'.

127.0.0.1:8080/vitae x Luis Guilherme...

127.0.0.1:8080/vitae-resolv... ☆

Dados recebidos:

Nome: Guilherme
E-mail: lgapontes@gmail.com
Hamburguer preferido: siri
Aceitou os termos: Sim

Contato cadastrado com sucesso!

Obrigado!

Próximo passo: CSS