

Aplicações para Internet



Cruzeiro do Sul Virtual
Educação a Distância

Material Teórico



Formulários e Vídeos

Responsável pelo Conteúdo:

Prof. Ms. Alexander Gobbato Paulino Albuquerque

Revisão Textual:

Profa. Ms. Fátima Furlan

UNIDADE

Formulários e Vídeos



- Formulários
- Outros recursos do HTML5



OBJETIVO DE APRENDIZADO

Estudar:

- Validação de documentos HTML5
- O trabalho com vídeos no HTML5
- O trabalho com áudio no HTML5



Orientações de estudo

Para que o conteúdo desta Disciplina seja bem aproveitado e haja uma maior aplicabilidade na sua formação acadêmica e atuação profissional, siga algumas recomendações básicas:



Assim:

- ✓ Organize seus estudos de maneira que passem a fazer parte da sua rotina. Por exemplo, você poderá determinar um dia e horário fixos como o seu “momento do estudo”.
- ✓ Procure se alimentar e se hidratar quando for estudar, lembre-se de que uma alimentação saudável pode proporcionar melhor aproveitamento do estudo.
- ✓ No material de cada Unidade, há leituras indicadas. Entre elas: artigos científicos, livros, vídeos e sites para aprofundar os conhecimentos adquiridos ao longo da Unidade. Além disso, você também encontrará sugestões de conteúdo extra no item **Material Complementar**, que ampliarão sua interpretação e auxiliarão no pleno entendimento dos temas abordados.
- ✓ Após o contato com o conteúdo proposto, participe dos debates mediados em fóruns de discussão, pois irão auxiliar a verificar o quanto você absorveu de conhecimento, além de propiciar o contato com seus colegas e tutores, o que se apresenta como rico espaço de troca de ideias e aprendizagem.

Formulários

Todos já devem pelo menos uma vez preenchido um formulário de cadastro na web, certo? A ideia de formulário todos conhecem, resumindo, sua principal utilidade é coletar os dados dos usuários e fazer algum tipo de processamento no servidor, porém também podemos utilizar para processar algo em *JavaScript*, no cliente.

Podemos criar nossos formulários através do HTML, utilizando tags para cada tipo de elemento de entrada de dados. No HTML5, também temos alguns atributos que irão facilitar a validação do formulário sem a necessidade de desenvolver scripts em outras linguagens, ou seja, para validar alguns dados não iremos precisar utilizar o *JavaScript*.

Como a linguagem HTML5 ainda está em desenvolvimento, alguns atributos e comandos de formulários não são compatíveis com todos os navegadores, por isso iremos utilizar somente o que for compatível com pelo menos dois navegadores (Google Chrome e Firefox).

Para criarmos um formulário podemos utilizar a tag `<form>` ou em alguns casos somente o elemento como um botão para disparar uma ação na nossa página.

Tag form

```
<form name="form1" method="post" action="">  
Elementos do formulário  
</form>
```

Tag para entrada de dados pode ter diversas variações conforme exemplos abaixo:

```
Nome: <input type="text" name="nome" />  
Senha: <input type="password" name="senha" />  
E-mail: <input type="email" name="usermail" /> (esta opção  
irá validar o campo de e-mail)  
URL: <input type="url" name="homepage" /> (usado para  
validar urls)  
<input type="submit" value="Enviar" /> (Para enviar o form  
para o servidor)
```

Exemplo1:

```

1  <!DOCTYPE html>
2
3  <html>
4    <head>
5      <title>Unidade III</title>
6      <meta name="description" content="Formulários">
7      <meta charset="utf-8">
8    </head>
9
10   <body>
11     <form name="form1" method="post" action="">
12       Nome: <input type="text" name="nome" />
13       Senha: <input type="password" name="senha" />
14       E-mail: <input type="email" name="usermail" />
15       URL: <input type="url" name="homepage" />
16       <input type="submit" value="Enviar" />
17     </form>
18   </body>
19
20 </html>

```

Figura 1

Resultado:

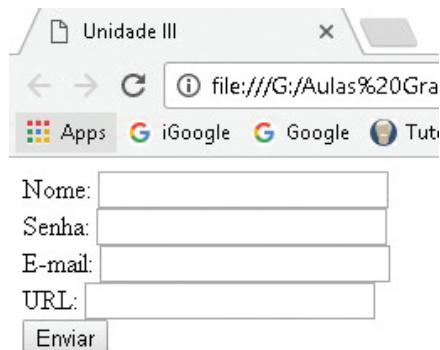


Figura 2

Campo Oculto

```
<input name="oculto" type="hidden" value="algumacoisa">
```

Checkbox (o usuário pode marcar mais de uma opção)

```
<input type="checkbox" name="carros" value="Camaro" />Camaro <br/>
<input type="checkbox" name="carros" value="Ferrari" />Ferrari
```

Button para chamar Funções em Javascript

```
<input name="botao" type="button" value="Clique aqui" onclick="alert('oi')"/>
```

Carregar arquivos no formulário

Escolha um arquivo

```
<input name="arquivo" type="file" />
```

Image para definir um botão com imagem para enviar o formulário

```
<input name="imagem" type="image" src="images.jpg" />
```

Radio (neste tipo o usuário poderá marcar somente uma opção)

```
<input type="radio" name="sexo" value="F" />Feminino <br/>
<input type="radio" name="sexo" value="M" />Masculino
```

Exemplo2:

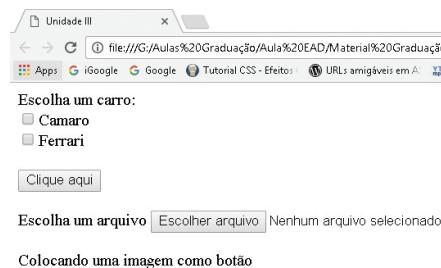
```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Unidade III</title>
5          <meta name="description" content="Formulários">
6          <meta charset="utf-8">
7      </head>
8      <body>
9          <form name="form1" method="post" action="">
10             <input name="oculto" type="hidden" value="algumacoisa">
11
12             Escolha um carro:<br/>
13             <input type="checkbox" name="carros" value="Camaro" />Camaro <br/>
14             <input type="checkbox" name="carros" value="Ferrari" />Ferrari<br/><br/>
15
16             <input name="botao" type="button" value="Clique aqui" onclick="alert('oi')"/>
17
18             <br/><br/>
19             Escolha um arquivo <input name="arquivo" type="file" />
20
21             <br/><br/>
22             Colocando uma imagem como botão<br/>
23             <input name="imagem" type="image" src="cruzeiro.jpg" />
24
25             <br/>
26             Escolha um sexo:<br/>
27             <input type="radio" name="sexo" value="F" />Feminino <br/>
28             <input type="radio" name="sexo" value="M" />Masculino
29
30         </form>
31     </body>
32 </html>

```

Figura 3

Resultado:



Escolha um sexo:
 Feminino
 Masculino

Figura 4



Sempre que desejamos que o usuário selecione mais de uma opção ao mesmo tempo, devemos utilizar o *checkbox*, agora se precisamos disponibilizar uma opção e permitir que somente e unicamente 1 opção seja selecionada, então utilizamos a opção *rádio*.

Outra nota que devemos nos atentar é que o campo com o type="hidden" o campo não será visível no browser, mas ele existe no formulário.

Tag Datalist

Define uma lista de opções para um elemento input, seria somente sugestões, o usuário poderá digitar valores diferentes.

```
<input list="frutas" name="fruta" />
<datalist id="frutas">
  <option value="Maça">
  <option value="Banana">
  <option value="Morango">
  <option value="Abacaxi">
  <option value="Melancia">
  <option value="Pera">
  <option value="Uva">
</datalist>
```

Tag Button

Nesta tag, você poderá colocar texto ou imagens como um botão clicável. Este comando pode ser utilizado fora de um formulário.

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

Tag Select

Cria uma lista de opções (conhecido como combobox)

```
<form>
<select>
  <option value="Java">Java</option>
  <option value="PHP">PHP</option>
  <option value="JSP">JSP</option>
  <option value="HTML">HTML</option>
</select>
</form>
```

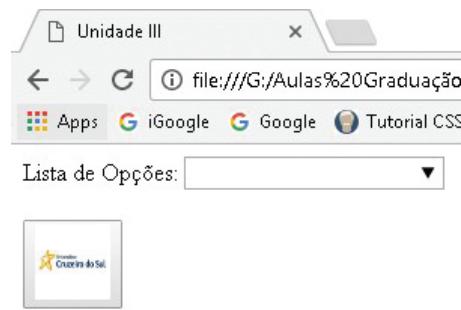
Exemplo3:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Unidade III</title>
5          <meta name="description" content="Formulários">
6          <meta charset="utf-8">
7      </head>
8      <body>
9          <form name="form1" method="post" action="">
10             Lista de Opções:
11             <input list="frutas" name="fruta" />
12             <datalist id="frutas">
13                 <option value="Maça">
14                 <option value="Melância">
15                 <option value="Perâ">
16                 <option value="Uva">
17             </datalist>
18
19             <br/><br/>
20             <button type="button">
21                 
22             </button>
23
24             <br/><br/>
25             Selecione uma linguagem:<br/>
26             <select>
27                 <option value="Java">Java</option>
28                 <option value="PHP">PHP</option>
29                 <option value="JSP">JSP</option>
30                 <option value="HTML">HTML</option>
31             </select>
32         </form>
33     </body>
34 </html>

```

Figura 5

Resultado:

Selecionar uma linguagem:
Java ▾

Figura 6

Dicas:

A tag *datalist* permite que o usuário selecione umas das opções pré-definidas e ainda permite que o usuário possa colocar uma informação adicional.

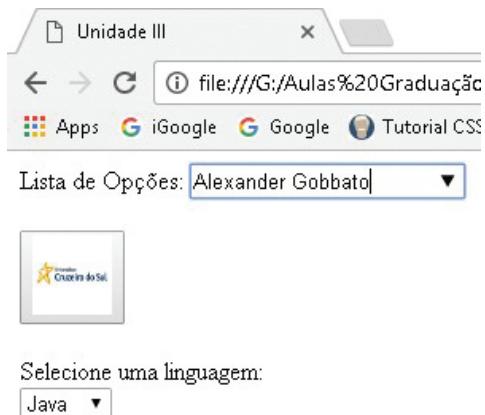


Figura 7

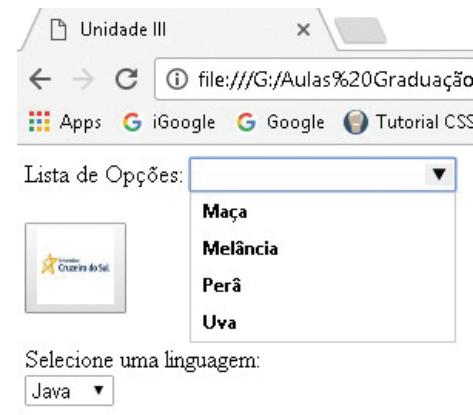


Figura 8

Tag textarea

Define um campo de texto de várias linhas

```
<form>
  <textarea rows="8" cols="70"></textarea>
</form>
```

Atributo required

Este atributo pode ser utilizado em campos para obrigar a digitação dos dados. A mensagem será ativada quando o usuário tentar enviar o formulário.

Pode ser utilizado em diversas tags de entrada de dados no formulário.

```
<input name="nome" required="required" />
<input type="submit"/>
```

Atributo autofocus

Este atributo identifica o elemento que será dado o foco quando a página for carregada. Pode ser utilizado em diversos elementos de um formulário.

```
<form action="x.php">
  Nome:<input type="text" name="nome" autofocus="autofocus" /><br />
  Idade: <input type="text" name="idade" /><br />
  <input type="submit" />
</form>
```

Atributo *placeholder*

Atributo exclusivo de elementos *input* e *textarea*, ele define uma dica para digitação no campo, assim que o usuário começar a digitar, a dica é automaticamente retirada

```
<form action="demo_form.asp">
    <input type="text" name="user" placeholder="Digite seu Usuário" /><br />
    <input type="text" name="senha" placeholder="Digite sua Senha" /><br />
    <input type="submit" value="Submit" />
</form>
```

Exemplo 4:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Unidade III</title>
5          <meta name="description" content="Formulários">
6          <meta charset="utf-8">
7      </head>
8      <body>
9          <form action="demo_form.jsp">
10             Usuário:<input type="text" name="user" placeholder="Digite seu Usuário" autofocus="autofocus" /><br />
11             Senha:<input type="text" name="senha" placeholder="Digite sua Senha" /><br />
12             Comentários:<br/>
13             <textarea rows="8" cols="70" required="required"></textarea><br />
14             <input type="submit" value="Submit" />
15         </form>
16     </body>
17 </html>
```

Figura 9

Resultado:

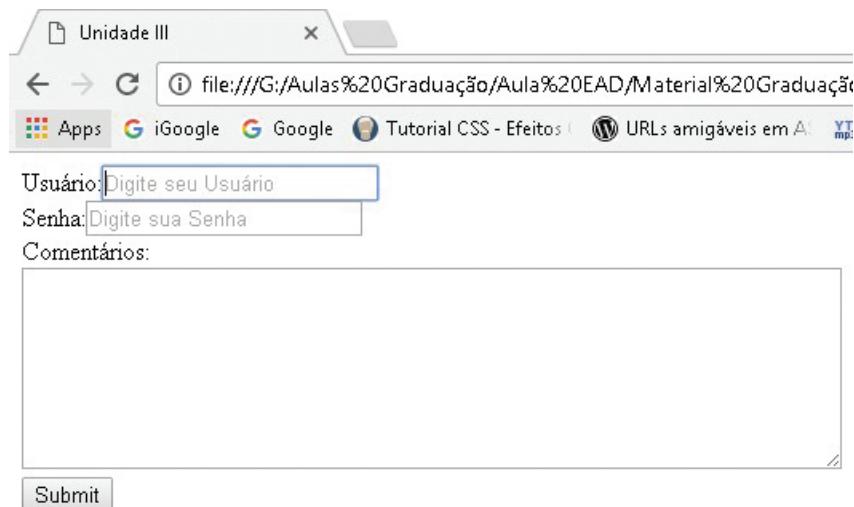


Figura 10

Dicas:

Ao abrir a página observem que o campo Usuário já está com o cursor posicionado no campo, isso se deve ao atributo *autofocus*.

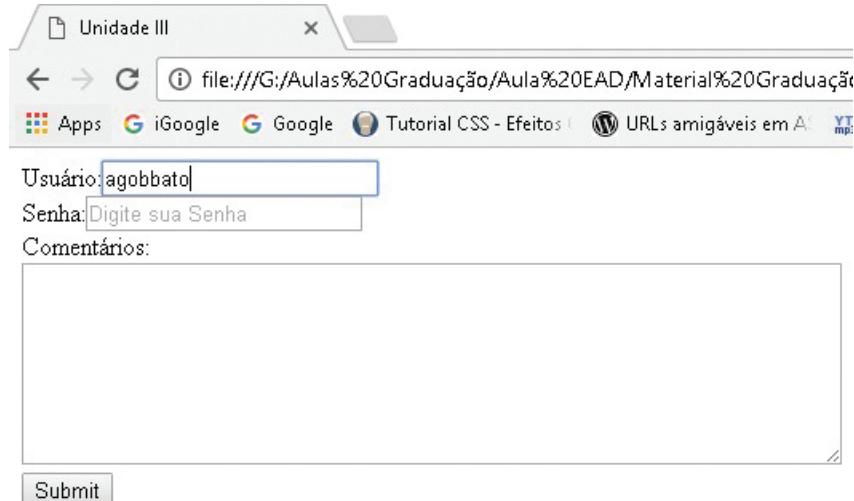


Figura 11

Outro detalhe é que podemos colocar uma informação no campo texto> Observem os campos usuário e senha, vejam que os campos possuem um pequeno texto, esse texto desaparece quando o usuário começa a digitar alguma informação, isso se deve ao atributo placeholder.

Dica:

O campo textarea é um campo texto que permite configurarmos a quantidade de linhas e colunas, definindo assim um tamanho diferente do campo type="text". Outro detalhe é que o atributo required só permite que as informações sejam enviados ao servidor se somente o campo estiver preenchido, caso contrário o próprio browser informa qual o campo deve ser preenchido.

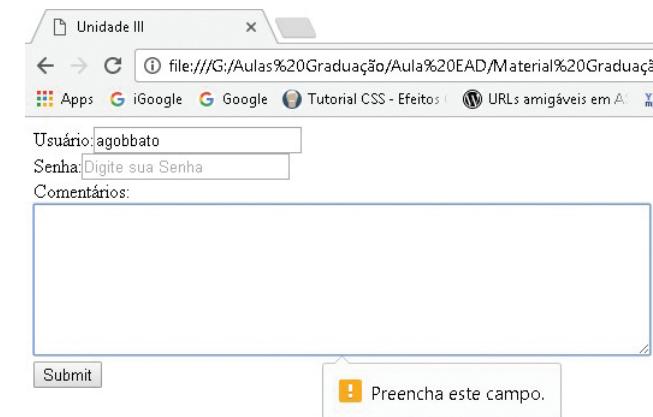


Figura 12

Vamos exemplificar agora como utilizar um formulário, enviar os dados ao servidor e visualizar o retorno do servidor.

```
server client + server side
<form action="processa.php" method="post">
```

Nome: ALCIDES TEIXEIRA BARBOZA JUNIOR

Sexo: M F

E-mail: alcidesprof@html.css

Assunto: Teste de Aula

Curso: Jogos Digitais

Mensagem:

Dúvida sobre o bloco
de notas...|

Figura 13

</form>

Observem que o atributo *action* está preenchido com a informação “processa.php”.

O arquivo “processa.php” é um arquivo que é executado no servidor, ou seja, ele captura as informações que foram preenchidas no formulário do cliente(browser).

```
<?php
header('Content-Type: text/html; charset=utf-8');
echo "Dados recebidos do formulário!!!<br/><br/>";
echo "Você digitou:<br/>";
echo "Nome: " . $_POST["nome"] . "<br />";
echo "Sexo: " . $_POST["sexo"] . "<br />";
echo "E-mail: " . $_POST["email"] . "<br />";
echo "Assunto: " . $_POST["assunto"] . "<br />";
echo "Curso: " . $_POST["curso"] . "<br />";
echo "Mensagem: " . $_POST["mensagem"] . "<br />";
?>
```

O resultado que será enviado ao cliente é o seguinte:

Dados recebidos do formulário!!!

Você digitou:

Nome: ALCIDES TEIXEIRA BARBOZA JUNIOR

Sexo: M

E-mail: alcidesprof@html.css

Assunto: Teste de Aula

Curso: TJD

Mensagem: Dúvida sobre o bloco de notas...

Figura 14

Áudio

Antes do surgimento do HTML5, não existia um padrão para reproduzir som no navegador, o que era possível somente através de plug-ins.

Na especificação HTML5, existe uma tag que se destina a incorporar um som ou stream (fluxo) de áudio na página sem a necessidade de plug-ins, essa tag seria `<audio>...</audio>`

Como ainda não fecharam o codec que será utilizado, devemos implementar o código para todos os formatos suportados de áudio, o comando ou tag para essa tarefa seria: `<source>...</source>` utilizado junto com a tag de áudio.

Tabela 1 – Compatibilidade entre os formatos e os navegadores

Navegador	MP3	WAV	OGG
Internet Explorer	SIM	SIM	NÃO
Firefox	SIM	SIM	SIM
Google Chrome	SIM	SIM	SIM
Apple Safari	SIM	SIM	NÃO

Tag simples

```
<audio src="picapau.mp3" controls="controls"></audio>
```

Para disponibilizar mais de um formato podemos utilizar a tag source dentro da tag áudio.

```
<audio controls="controls">
    <source src="som de derrota.ogg" type="audio/ogg" />
    <source src="som de derrota.mp3" type="audio/mpeg" />
    <source src="som de derrota.wav" type="audio/vnd.wave" />
    Seu navegador não aceita a tag audio
</audio>
```



Figura 15

Controlando o áudio com javascript

Esse exemplo é somente uma demonstração, nessa unidade ainda não abordaremos javascript, simplesmente para demonstrar que é possível também utilizar.

```

<audio id="som_controle_js">
    <source src="som de derrota.ogg" type="audio/ogg" />
    <source src="som de derrota.mp3" type="audio/mpeg" />
    <source src="som de derrota.wav" type="audio/vnd.wave" />
</audio>
<button id="play">Play</button> | <button id="pause">Pause</button>

<script>
var som = document.getElementById('som_controle_js');
document.getElementById('play').addEventListener('click', function() {
    som.play(); }, false);
document.getElementById('pause').addEventListener('click', function() {
    som.pause(); }, false);
</script>

```

Vídeo

Antes do surgimento do HTML5, também não existia um padrão para reproduzir vídeo no navegador, o que era possível somente através de plug-ins.

Na especificação HTML5, existe uma tag que se destina a incorporar um vídeo ou stream (fluxo) de video na página sem a necessidade de plug-ins, essa tag seria `<video>...</video>`

Como também não houve um consenso sobre o formato de vídeo a ser adotado no HTML5, devemos implementar o código para todos os formatos suportados de vídeo, o comando ou tag para essa tarefa seria: `<source>...</source>` utilizado junto com a tag de vídeo.

Tabela 2 – Compatibilidade entre os formatos e os navegadores

Navegador	MP4	WebM	OGG
Internet Explorer	SIM	NÃO	NÃO
Firefox	SIM	SIM	SIM
Google Chrome	SIM	SIM	SIM
Apple Safari	SIM	NÃO	NÃO

Um arquivo de vídeo utiliza compressão e codecs, por isso as empresas não concordaram em adotar um único formato.

MP4 = codec de vídeo MPEG4 (H.264 para MPEG)

WebM = codec de vídeo VP8

Ogg = codec de vídeo Theora

Tag simples

```

<video src="chrome.webm" controls="controls" width="400px"></video>

```

Para disponibilizar mais de um formato podemos utilizar a tag source dentro da tag video

```
<video poster="chrome.jpg" controls width="400px" >
  <source src="chrome.ogv" type="video/ogg" />
  <source src="chrome.webm" type="video/webm" />
  <source src="chrome.mp4" type="video/mp4" />
  Seu navegador não suporta a tag de video.
</video>
```

Controlando o vídeo com javascript

```
<video id="video_js" width="400px" >
  <source src="chrome.ogv" type="video/ogg"/>
  <source src="chrome.webm" type="video/webm" />
  <source src="chrome.mp4" type="video/mp4" />
  Seu navegador não suporta a tag de video.
</video>
<button id="play">Play</button> | <button id="pause">Pause</button>

<script>
var video = document.getElementById('video_js');
document.getElementById('play').addEventListener('click', function() {
  video.play(); }, false);
document.getElementById('pause').addEventListener('click', function() {
  video.pause(); }, false);
</script>
```

Outros recursos do HTML5

O HTML5 possui outros recursos avançados como o canvas, porém exigem programação em JavaScript, neste momento não iremos passar esses recursos, mas seria interessante uma breve explicação:

- **Geolocalização** – API destinada a localizar o usuário através de GPS, endereço IP, redes sem fio, sinal de telefone etc.
Armazenamento de dados no cliente (*Web Storage* e *WebSQL*)
- **Web Messaging** – Permite a comunicação entre aplicações hospedadas em diferentes origens
- **Web Socket** – Permite a comunicação bidirecional entre o cliente e o servidor
- **Web Workers** – Permite a execução de vários scripts (código JavaScript) em uma mesma máquina, rodando em segundo plano ou em paralelo

Material Complementar

Indicações para saber mais sobre os assuntos abordados nesta Unidade:



Sites

WebGL - Jellyfish

<https://goo.gl/ONbTY>

World Population

<https://goo.gl/pM5qQ>

sceneJS - 3D Engine for the Web

<http://scenejs.org/>

BioDigital

<https://goo.gl/z8G8i>

Referências

Mauricio SAMY Silva. **HTML 5 - A Linguagem de Marcação que revolucionou a WEB.** São Paulo: Novatec, 2011



Cruzeiro do Sul Virtual
Educação a Distância

www.cruzeirodosulvirtual.com.br
Campus Liberdade
Rua Galvão Bueno, 868
CEP 01506-000
São Paulo - SP - Brasil
Tel: (55 11) 3385-3000



Cruzeiro do Sul
Educacional