

DESIGN WEB

AULA 3 – INTRODUÇÃO HTML

Isabel Bustamante



UNIDADE CURRICULAR

Unidade Curricular: Design Web	
Carga Horária: 60h	
Função: <ul style="list-style-type: none">F.1 : Produzir Interfaces para internet, de acordo com metodologia e padrões de qualidade, usabilidade, interatividade, robustez, acessibilidade e segurança da informação.F.2 : Desenvolver sistemas para internet, de acordo com metodologia e padrões de qualidade, usabilidade, interatividade, robustez, acessibilidade e segurança da informação.	
Objetivo Geral: Propiciar o desenvolvimento de capacidades básicas e socioemocionais relativas à lógica de programação que subsidiarão o desenvolvimento das capacidades técnicas da ocupação	
Conteúdos Formativos	
Capacidades Básicas	Conhecimentos
<ul style="list-style-type: none">Conhecer os limites de atuação profissional em Web Design;	Introdução 1.1. Como a Web funciona 1.2. Padrões Web 2. HTML

UNIDADE CURRICULAR

- Estruturar documentos web usando a linguagem HTML;
- Utilizar as melhores práticas – web standards – relacionadas pelo W3C;
- Formatar a apresentação de documentos web utilizando CSS;
- Estruturar um documento web para encontrar informações rapidamente.

- 2.1. Sintaxe
- 2.2. Títulos
- 2.3. Parágrafo
- 2.4. Links
- 2.5. Listas
- 2.6. Imagens
- 2.7. Tabelas
- 2.8. Formulários
- 2.9. Informações meta
- 2.10. Seções
- 3. Formatação com CSS
 - 3.1. Sintaxe
 - 3.2. Formas de inclusão
 - 3.3. Textos, cores e fontes
 - 3.4. Plano de fundo
 - 3.5. Dimensões, espaçamentos e bordas
 - 3.6. Lista
 - 3.7. Posicionamentos
 - 3.8. Subclasses
- 4. Arquitetura da informação
 - 4.1. Introdução
 - 4.2. Comportamento e necessidades do usuário
 - 4.3. Estruturas de navegação
 - 4.4. Estruturas de busca
 - 4.5. Padrões de design
 - 4.6. Navegação em aplicações web ricas
 - 4.7. Pesquisa e identificação das informações dos usuários
 - 4.8. Projeto de arquitetura da informação

CRONOGRAMA

13/03

Início

15/04

Avaliação

13/05

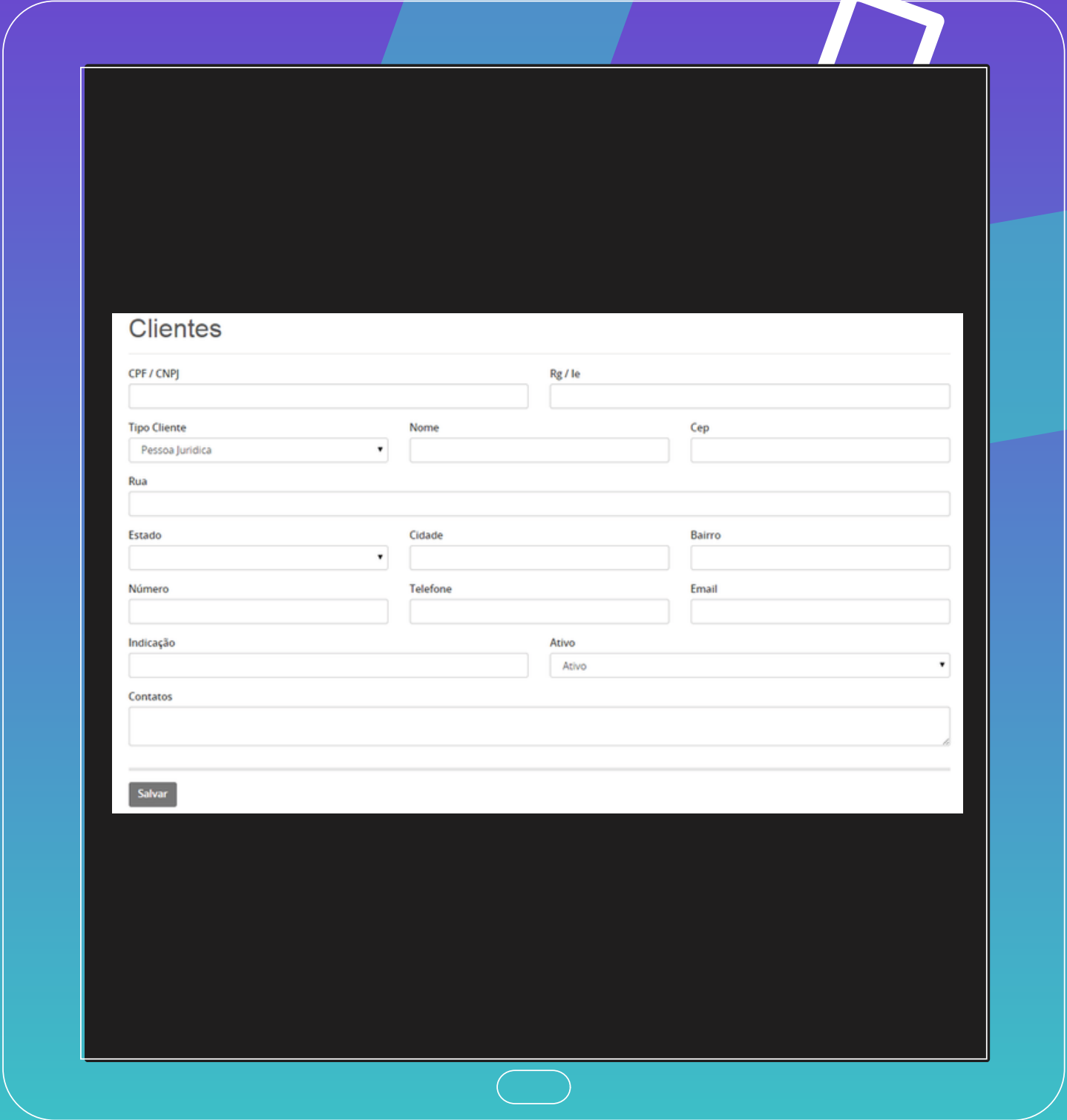
Trabalho

15/05

Recuperação

Formulário HTML: criando forms HTML e enviando dados

O uso de formulário HTML (ou HTML form) é muito frequente na web, é um dos principais pontos de interação entre usuários e uma aplicação web. Geralmente, eles funcionam enviando os dados para um servidor, porém também podem ser utilizados para alguma função na própria página.



The image shows a tablet screen with a web form titled "Clientes". The form contains the following fields:

- CPF / CNPJ (text input)
- Rg / Ie (text input)
- Tipo Cliente (dropdown menu, currently showing "Pessoa Juridica")
- Nome (text input)
- Cep (text input)
- Rua (text input)
- Estado (dropdown menu)
- Cidade (text input)
- Bairro (text input)
- Número (text input)
- Telefone (text input)
- Email (text input)
- Indicação (text input)
- Ativo (checkbox, currently checked)
- Contatos (text input)
- Salvar (button)

Como inserir um formulário HTML

O formulário HTML é representado pela tag de abertura `<form>` e a de fechamento `</form>`. Dentro dessas tags, serão colocados todos os elementos que compõem este formulário.

Para a tag `<form>` podem ser atribuídos o atributo `method` e o atributo `action`.

O atributo `action` define o local (através de uma URL) ao qual serão enviados todos os dados recolhidos do formulário.

O atributo `method` define o método HTTP com que o formulário HTML irá lidar com os dados recebidos. São eles: o método GET e o método POST.

Os métodos GET e POST

Para entender a diferença entre esses dois métodos, é necessário compreender como funcionam as requisições HTTP. De uma forma resumida, sempre que você acessa um recurso através da web, o navegador envia uma requisição através da URL. Portanto o HTTP é uma requisição que consiste em duas partes, o cabeçalho e o corpo. O cabeçalho contém um conjunto de metadados globais envolvendo os recursos do navegador. Já o corpo pode conter informações necessárias para o servidor conseguir processar a solicitação anterior.

```
<!--Formulário HTML através do método POST-->  
<form method="post" action="/receber_dados.php">  
    ...  
</form>
```

```
<!--Formulário HTML através do método GET -->  
<form method="get" action="/receber_dados.php">  
    ...  
</form>
```

Os métodos GET e POST



POST





GET





O método POST



O método POST é utilizado no navegador para conversar com o servidor. Os dados são enviados ao servidor através do corpo da solicitação HTTP. Portanto, é como se falássemos ao servidor “Servidor, verifique esses dados e me retorne um resultado adequado”. Dessa forma, ao enviar um formulário através do método POST, os dados serão anexados ao corpo da solicitação HTTP.



O método GET



O método GET é usado pelo navegador para solicitar que o servidor envie de volta um determinado recurso. Portanto, é como se falássemos ao servidor “Servidor, eu quero obter este recurso.”. Nesse caso, o navegador envia um corpo vazio. Portanto, já que o corpo fica vazio, se um formulário for enviado através do método GET, os dados serão reconhecidos pelo servidor através da URL.



The background is a gradient of purple and blue. It features several geometric shapes and symbols: a large blue checkmark at the top left, a white plus sign above it, a white downward-pointing triangle at the top center, a white square at the top right, a large teal checkmark at the bottom left, a white zigzag line below it, a white upward-pointing triangle at the bottom center, and a white circle at the bottom right. A white bracket-like shape frames the left and right sides of the text area.

Observações a respeito do método GET

Dessa forma, podemos perceber que ao enviar um formulário com os campos Nome e Idade, a URL iria se parecer com algo como: `url?nome=valor&idade=value`.

Esse é o método padrão do formulário HTML. Caso não seja declarado o atributo `method`, o formulário funcionará através do método GET.

EXEMPLO DE FORMULÁRIO DA TABELA HTML

```
<form method="get" action="envio_dados.php">  
  <input type="text" name="nome" placeholder="digite seu Nome aqui">  
  <input type="e-mail" name="email" placeholder="Digite seu e-mail">  
  <input type="submit" name="enviar" value="Enviar">  
</form>
```

Observações a respeito do método GET

- O tamanho de uma URL é limitado a cerca de 3000 caracteres;
- Nunca use o GET para enviar dados confidenciais, pois esses dados ficarão visíveis na URL;
- É útil para envios de formulários em que um usuário deseja marcar o resultado;
- GET é melhor para dados não seguros, como strings de consulta no Google.

ELEMENTOS DE FORMULÁRIO HTML

- **<input>** – Define um campo de entrada de dados;
- **<textarea>** – Define uma área de texto, podendo conter diversas linhas de texto;
- **<button>** – Define um botão;
- **<select>** – Define uma lista selecionável, também conhecida como drop-down;
- **<option>** – Define uma lista de opções dentro de um drop-down select;
- **<optgroup>** – Define um grupo de opções;
- **<fieldset>** – Define um grupo de campos;
- **<label>** – Define um rótulo ou legenda para um campo ou controle do formulário;
- **<output>** – Define elementos de saída para o formulário;
- **<legend>** – Define um título para o conjunto de campos.

A TAG <INPUT>

A tag input em HTML é uma das mais utilizadas dentro de um formulário HTML e considerada a principal. Esta tag representa os campos de entrada de dados de um formulário. Existem diversos tipos, principalmente após a atualização do HTML para a versão HTML5, que trouxe diversos tipos novos de <input>.

```
<form method="get" action="envio_dados.php">  
  <input type="text" name="nome" placeholder="digite seu Nome aqui">  
  <input type="e-mail" name="email" placeholder="Digite seu e-mail">  
  <input type="submit" name="enviar" value="Enviar">  
</form>
```

ATRIBUTOS DA TAG <INPUT>

- **type** – Esse atributo representa o tipo de campo de entrada ao qual essa tag pertence e como ele se comporta
- **name** – define um nome para o campo de entrada
- **value** – define um valor inicial para o campo de entrada.
- **placeholder** – permite digitarmos uma “descrição” que será visualizada enquanto o campo de entrada não possuir nenhum valor.
- **required** – manter o preenchimento de um campo obrigatório

```
<form method="get" action="envio_dados.php">  
  <input type="text" name="nome" placeholder="digite seu Nome aqui">  
  <input type="e-mail" name="email" placeholder="Digite seu e-mail">  
  <input type="submit" name="enviar" value="Enviar">  
</form>
```

https://www.homehost.com.br/blog/tutoriais/formulario-html/#Como_inserir_um_formulario_HTML

https://www.w3schools.com/html/html_forms.asp