

formulários

# Desenvolvimento WEB I

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas  
Prof. Felipe Scheidt – IFPR – Campus Foz do Iguaçu  
2024

# Formulários

- Em páginas web é comum que sejam solicitadas informações do usuário.
- Exemplos: pesquisa de opinião, fazer login, realizar cadastro, ...
- Formulário é um elemento do HTML que permite a entrada de dados e o envio destes para o servidor.

# <form>

- Um formulário é criado com a tag **<form>**
- O escopo da tag <form> </form> delimita o início e o fim do formulário, ou seja, todas as entradas de dados devem estar contidas dentro da tag form.
- Para enviar os dados do formulário, deve existir um botão que dispara o envio.

# <form> html

<form>

Email: <input type="text" name="email">

<button>enviar</button>

</form>

Email:

# <form> - atributos

A tag form possui dois atributos que precisam ser definidos para envio correto dos dados:

- **action**: define o *destino* para onde os dados serão enviados
- **method**: define o método de envio [get/post]

```
<form action="/endpoint" method="POST">
```

```
<!-- inputs do formulario -->
```

```
<button>enviar</button>
```

```
</form>
```

# <input>

- A tag input cria um "campo", onde é permitido ao usuário entrar com uma informação.
- Uma input pode ter diferentes **tipos** (texto, número, data,...)
- O tipo é definido pelo atributo type.

# <input> - atributos

```
<form>
```

```
  <label>Nome</label>
```

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

```
  <label>Peso</label>
```

```
  <input type="number" name="peso">
```

```
</form>
```

Nome

Peso

Atributos necessários para usar a tag input no formulário:

**type:** define o tipo de input (ex.: text, number)

**name:** atribui um nome para a input (ex.: nome, peso)

É comum também associar um **label** (rótulo) para cada input.

# <input> - tipo

Exemplo de tipos de entradas:

- **text**: aceita caracteres alfanuméricos
- **radio**: é um tipo de botão, onde apenas uma opção pode ser selecionada
- **checkbox**: permite selecionar várias opções.
- **password**: um campo cria uma máscara nos caracteres digitados.
- **date**: define um campo que aceita apenas datas.



# form com duas colunas (css)

Duas colunas é um padrão de layout comum quando se trabalha com formulários. Esse layout é definido com as propriedades `display grid` e `grid-template-columns` (número de colunas). Exemplo:

```
.grid {  
  display: grid;  
  grid-template-columns: max-content max-content;  
  align-items: center;  
  gap: 10px;  
}  
  
.grid label {  
  justify-self: right;  
}
```

# form duas colunas (html)

Para aplicar o grid de duas colunas, basta vincular o nome da classe do CSS na tag <form>

```
<form class="grid">  
  <label>Nome</label>  
  <input type="text" name="nome">  
  <label>CPF</label>  
  <input type="text" name="cpf">  
  <label>Endereço</label>  
  <input type="text" name="endereco">  
</form>
```

Nome


CPF

Endereço

# Arquivo CSS externo

Para vincular um arquivo css, usamos a tag `<link>` e o atributo `href` (com o caminho para o arquivo)

```
<head>  
  <meta charset="UTF-8">  
  <link href="filename.css" rel="stylesheet">  
  <title>Page title</title>  
</head>
```

A diagram consisting of a horizontal arrow pointing to the right, positioned to the left of the `<link>` tag in the code block above. A vertical line extends upwards from the arrow's tail to the opening angle bracket of the `<link>` tag.

# Formas de vincular CSS ao HTML

## 3º – Externo

url

```
<link rel="stylesheet"
      href="https://cdn.jsdelivr.net/npm/bootstrap/dist/css/bootstrap.min.css" >
```

local

```
<link href="filename.css" rel="stylesheet">
```

---

## 2º – Na head

```
<head>
  <meta charset="UTF-8">
  <style>
    main {
      width: 600px;
      margin: auto;
    }
  </style>
</head>
```

---

## 1º – Inline

```
<div style="color: ■ red;font-family: 'Hack';">
  <h3>jungle</h3>
  <img src='../imgs/jungle01.jpg' />
</div>
```

# Validação input

**maxlength:** quantidade máxima de caracteres aceitos

**minlength:** quantidade mínima de caracteres.

```
<form class="grid">
  <label>Cidade</label>
  <input type="text" name="nome" minlength="3">
  <label>Estado</label>
  <input type="text" name="estado" maxlength="2">
</form>
```

Cidade	<input type="text" value="Foz do Iguaçu"/>
Estado	<input type="text" value="PR"/>

# radio

O tipo **radio** limita a seleção de apenas **uma** opção dentre várias. O atributo **name** deve ser igual.

```
<form class="grid1">
  <h2>Qualidade do serviço?</h2>
  <label>
    <input type="radio" name="pesquisa" value="3">
    Excelente
  </label>
  <label>
    <input type="radio" name="pesquisa" value="2">
    Bom
  </label>
  <label>
    <input type="radio" name="pesquisa" value="1">
    Péssimo
  </label>
</form>
```

Qualidade do serviço?

- ☐ Excelente
- ☐ Bom
- ☐ Péssimo

# checkbox

Permite selecionar **várias** opções

```
<form>
  <h2>Tipos de séries favoritas</h2>
  <label>
    <input type="checkbox" name="favoritos" value="1">
    Aventura
  </label>
  <label>
    <input type="checkbox" name="favoritos" value="2">
    Drama
  </label>
  <label>
    <input type="checkbox" name="favoritos" value="3">
    Ficção científica
  </label>
</form>
```

## Tipos de séries favoritas

- ☒ Aventura
- ☐ Drama
- ☒ Ficção científica

# <select>

A select é semelhante ao input radio, ou seja, permite a seleção de apenas **uma** opção. É mais prática de trabalhar quando se tem uma lista com muitas opções

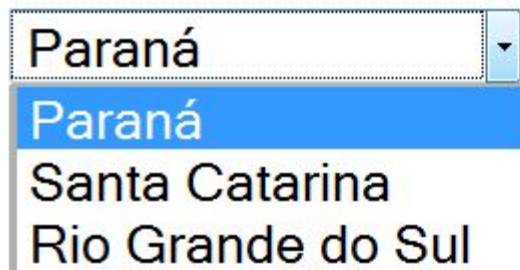
normal

Selecione o estado onde mora?

A normal select dropdown menu with a light blue border and a small downward arrow on the right. The text 'Paraná' is displayed inside the box.

clicado

Selecione o estado onde mora?

A select dropdown menu with the list of options expanded. The menu has a light blue border. The text 'Paraná' is displayed inside the box. Below the box, a list of options is shown: 'Paraná', 'Santa Catarina', and 'Rio Grande do Sul'. The 'Paraná' option is highlighted with a blue background.



# <select>

Exemplo do select em um formulário

Em qual estado você mora?

```
<form>
```

```
<select name="estados">
```

```
  <option value="AM">AMAZONAS</option>
```

```
  <option value="PR">Paraná</option>
```

```
  <option value="SC">Santa Catarina</option>
```

```
  <option value="SP">São Paulo</option>
```

```
</select>
```

```
</form>
```

# <select>

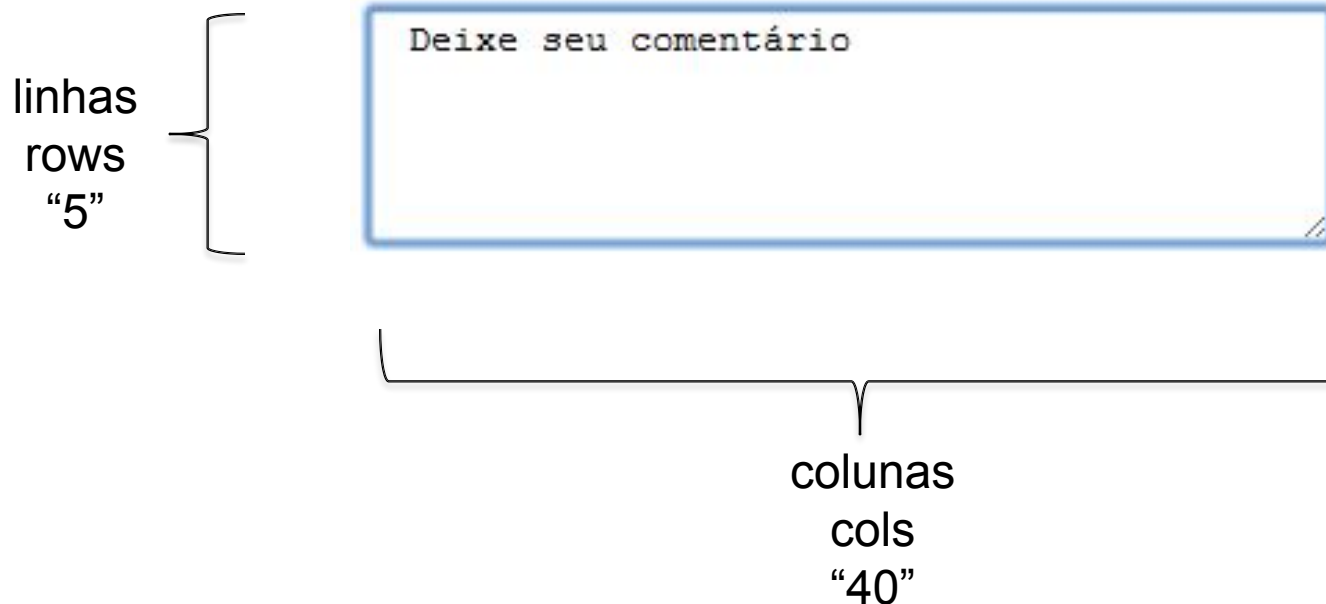
- O atributo **selected** configura uma opção padrão pré-selecionada. No exemplo abaixo, Santa Catarina será a opção padrão quando a página é carregada.

```
<select name="estados">
  <option value="AM">AMAZONAS</option>
  <option value="PR">Paraná</option>
  <option selected value="SC">Santa Catarina</option>
  <option value="SP">São Paulo</option>
</select>
```

# <textarea>

- **textarea** permite a entrada de uma grande quantidade de texto, como por exemplo em campos do tipo comentários, reclamação, observações...
- O tamanho da textarea é configurado com os atributos:
  - **rows**: quantidade de linhas
  - **cols**: quantidade de colunas

# <textarea>



```
<textarea cols="40" rows="5" name="comentario">  
    Deixe seu comentário  
</textarea>
```

## + tipos de inputs

`<input type="email">` *somente emails válidos*

`<input type="number">` *apenas um valor numérico*

`<input type="range">` *valor é definido em um intervalo*

`<input type="date">` *tipo data (calendário)*

`<input type="time">` *input do tipo horário*

`<input type="datetime-local">` *valor data e horário*

## + atributos da input

ATRIBUTO	VALORES	FUNÇÃO
<b>autocomplete</b>	on, off	define se o campo aceita autocomplete
<b>disabled</b>	disabled	define que a tag está desabilitada
<b>placeholder</b>	texto	fornece uma <b>dica</b> ao usuário sobre a entrada
<b>required</b>	required	campo obrigatório
<b>min</b>	número ou data	valor <b>mínimo</b> que o input aceita
<b>max</b>	número ou data	valor <b>máximo</b> que o input aceita

# Formatação validação da input

Podemos formatar o estilo da validação, em caso de uma entrada **inválida** ou **válida** usando o CSS:

```
/* regra para valores inválidos*/  
input:required:invalid {  
}  
/* regra para valores válidos */  
input:required:valid {  
}  
/* regra para valores inválidos  
quando input recebe o foco*/  
input:focus:invalid {  
}  
/* regra para valores validos  
quando input recebe o foco*/  
input:focus:valid {  
}
```

# Exercício 1

Faça o código HTML e CSS do formulário abaixo usando grid Layout.

Data de nascimento:	<input type="text" value="dd/mm/aaaa"/>
Escolaridade:	<input type="text" value="Ensino Fundamental ▼"/>
Situação da Escolaridade:	<input type="radio"/> Completo <input type="radio"/> Incompleto
Nacionalidade:	<input type="text" value="Brasil ▼"/>
UF onde nasceu:	<input type="text" value="Paraná ▼"/>
<input type="button" value="enviar"/>	



# Exercício 2

Nome

Email

Password

Possui conhecimento?

☐

JavaScript

☐

HTML

☐

CSS

Curso



Turno

☐

Tarde

☐

Noite

Motivo matrícula

Opções em Curso:

- Análise e Desenvolvimento
- Física
- Matemática

# Exercício 3 (validação)

Nome	<input type="text" value="La"/>
Email	<input type="text" value="user@example.org"/>
Password	<input type="password" value="..."/>

Faça outra versão do **exercício 2** contendo a validação dos campos

**obrigatórios:**

- Nome
- Email
- Password

**Nome** possui no mínimo 3 caracteres.

**Password** deve ter no mínimo 6 caracteres e no máximo 10 caracteres.

→ Escreva a **regra CSS** que formata o texto da input na cor vermelha quando a entrada digitada é **inválida**

# Exercício 4

## Cadastro usuário

Nome

Email

Nascimento



Pais



Idade

DDD

Telefone