

# Introdução ao Desenvolvimento Web

## Tecnologia em Análise e Desenvolvimento de Sistemas

Senac - Portão - Curitiba/PR

Prof. Guilherme Matos Barbosa



# Formulários

- Os formulários desempenham um papel crucial na interação entre os usuários e os sites da web.
- Eles permitem que os usuários enviem informações, façam login, se cadastrem e realizem várias outras ações importantes.
- Com HTML5 e CSS3, podemos criar formulários altamente funcionais e visualmente atraentes para melhorar a experiência do usuário.



# Formulários

Tags:

- `<form>` : define o início de um formulário;
- `<fieldset>` : agrupa elementos de um formulário;
- `<legend>` : nomeia e descreve o conteúdo agrupado;
- `<label>` : nomeia e descreve um elemento de entrada do formulário;
- `<input>` : define um campo que vai receber dados do usuário.



# Formulários

```
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>Formulário HTML5</title>
7      <link rel="stylesheet" href="styles.css">
8  </head>
9  <body>
10     <form>
11         <!-- Campos do formulário serão adicionados aqui -->
12         <fieldset>
13             <legend>Informações Pessoais</legend>
14             <!-- Campos relacionados à informações pessoais -->
15         </fieldset>
16
17         <fieldset>
18             <legend>Preferências</legend>
19             <!-- Campos relacionados a preferências -->
20         </fieldset>
21
22         <button type="submit">Enviar</button>
23     </form>
24 </body>
25 </html>
```



# Campos de Entrada

```
1  /* Estilo para os fieldsets */
2  fieldset {
3      border: none;
4      margin-bottom: 20px;
5  }
6
7  legend {
8      font-weight: bold;
9      color: ☐ #333;
10     font-size: 1.2em;
11     margin-bottom: 10px;
12 }
```



# Formulários

O elemento `<input>` recebe um atributo `type` que define a maneira com que esse campo se comporta. O tipo padrão é um campo de texto, mas temos uma variedade de outros valores:

- `password`: cria um campo de entrada de senha onde o que o usuário digita é ocultado;
- `email`: cria um campo de entrada de e-mail que verifica se o valor preenchido é um e-mail válido;
- `date`: cria um campo de entrada que só aceita datas;
- `number`: cria um campo de entrada que só aceita números;
- `checkbox`: cria uma caixa de marcação;
- `button`: cria um botão;
- `submit`: é responsável por enviar os dados do formulário.



# Campos de Entrada

Campo de Texto:

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



# Campos de Entrada

Campo de Texto:

```
14  /* Estilo básico para todos os campos de entrada */
15  input[type="text"] {
16      width: 100%;
17      padding: 10px 0;
18      margin-bottom: 20px;
19      border: 1px solid #ccc;
20      border-radius: 5px;
21  }
```





# Campos de Entrada

Campo de Senha:

```
<label for="senha">Senha:</label>  
<input type="password" id="senha" name="senha" placeholder="Digite sua senha">
```



# Campos de Entrada

Campo de Senha:

```
14  /* Estilo básico para todos os campos de entrada */
15  input[type="text"],
16  input[type="password"] {
17      width: 100%;
18      padding: 10px 0;
19      margin-bottom: 20px;
20      border: 1px solid #ccc;
21      border-radius: 5px;
22  }
```



# Campos de Entrada

Campo de Senha:

```
<!-- Campos relacionados a preferências -->
<label for="pais">País:</label>
<select id="pais" name="pais">
  <option value="brasil">Brasil</option>
  <option value="eua">Estados Unidos</option>
  <option value="canada">Canadá</option>
</select>
```



# Campos de Entrada

Campo de Senha:

```
14  /* Estilo básico para todos os campos de entrada */
15  input[type="text"],
16  input[type="password"],
17  select {
18      width: 100%;
19      padding: 10px 0;
20      margin-bottom: 20px;
21      border: 1px solid #ccc;
22      border-radius: 5px;
23  }
```



# Campos de Entrada

## Validando Entradas:

- Podemos usar atributos HTML5 para realizar validação básica dos dados de entrada, como required para campos obrigatórios e pattern para definir um padrão de entrada para campos como e-mail.

```
<label for="email">Email:</label>  
<input type="email" id="email" name="email" placeholder="Digite seu email" required>
```



# Campos de Entrada

Validando Entradas:

```
/* Estilo básico para todos os campos de entrada */
input[type="text"],
input[type="password"],
input[type="email"],
select {
  width: 100%;
  padding: 10px 0;
  margin-bottom: 20px;
  border: 1px solid #ccc;
  border-radius: 5px;
}

input:invalid {
  border: 1px solid #FF1010;
}
```



# Campos de Entrada

Estilizando botão:

```
35  /* Estilo para o botão de envio */
36  button[type="submit"] {
37      background-color: #4CAF50;
38      color: white;
39      padding: 10px 20px;
40      border: none;
41      border-radius: 5px;
42      cursor: pointer;
43  }
44
45  button[type="submit"]:hover {
46      background-color: #45a049;
47  }
```



# Protocolo HTTP

- O HTTP é a base de todo software construído para internet;
- Ele define como deve ser feita uma requisição para uma aplicação, incluindo a URL, os parâmetros, os cabeçalhos e o tipo de resposta esperado;
- Os serviços que seguem o protocolo HTTP devem ser configurados com um verbo, que indica qual é o comportamento esperado do serviço;
- Os verbos HTTP são: OPTIONS , GET , HEAD , POST , PUT , DELETE , TRACE e CONNECT .





# Atributo *method* – Métodos GET e POST

Os formulários podem ser enviados ao servidor de duas maneiras diferentes:

O método GET (`method="get"`) envia os dados de formulário como uma string anexada à URL especificada no atributo e enviados como parte da linha de consulta (query string). Isso significa que os dados são visíveis na barra de endereço do navegador. O método GET deve ser usado para formulários simples sem dados confidenciais.

Exemplo: `<form action="processar_formulario.php" method="get">`



# Atributo *method* – Métodos GET e POST

Os formulários podem ser enviados ao servidor de duas maneiras diferentes:

O método POST (`method="post"`) oferece a capacidade de criar formulários seguros que transmitem informações confidenciais, enviando os dados de formulário por um canal separado, de modo que eles não são visíveis na barra de endereço. É comumente usado em formulários de login, cadastro e outras operações que envolvem dados confidenciais.

Exemplo: `<form action="processar_formulario.php" method="post">`