

Laboratório 10

Neste laboratório vamos construir um formulário que utilizará algumas funções JavaScript para validações e tornar a página mais dinâmica.

Página HTML:

1 – Crie uma pasta chamada lab10. Nessa pasta, crie um arquivo chamado **form** com a extensão **.html (form.html)**. Neste arquivo será criada nossa página HTML. Conforme a figura abaixo:

```
<html lang="pt-BR">
<head>
  <meta charset="utf-8">
  <title>Página Formulário</title>
  <link rel='stylesheet' type='text/css' href='style.css'>
</head>

<body>
  <header>
    <h1>Meu Formulário</h1>
  </header>
  <main>
    <section id="formulario">
      <form action="form-submit.html" method="get">
        <fieldset>
          <legend>Formulário de Contato</legend>
          <div>
            <label for="nome">Nome:</label>
            <input type="text" id="nome" name="nome" required>
          </div>
          <div class="linha">
            <div>
              <label for="nomeMae">Nome da mãe:</label>
              <input type="text" id="nomeMae" name="nomeMae" required>
            </div>
            <div>
              <label for="nomePai">Nome do pai:</label>
              <input type="text" id="nomePai" name="nomePai">
            </div>
          </div>
          <div class="linha">
            <div>
              <label for="cpf">CPF:</label>

```

```
        <input type="text" id="cpf" name="cpf" required>
    </div>
    <div>
        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required>
    </div>
    <div>
        <label for="estadoCivil">Estado Civil:</label>
        <select id="estadoCivil" name="estadoCivil" required>
            <option>-</option>
            <option value="solteiro">Solteiro</option>
            <option value="casado">Casado</option>
            <option value="viuvo">Viuvo</option>
            <option value="uniao_estavel">União Estavel</option>
        </select>
    </div>
</div>
<div class="linha">
    <div>
        <label for="estado">Estado:</label>
        <select id="estado" name="estado" required>
            <option>-</option>
            <option value="bahia">Bahia</option>
            <option value="sao_paulo">São Paulo</option>
            <option value="rio_janeiro">Rio de Janeiro</option>
            <option value="tocantins">Tocantins</option>
        </select>
    </div>
    <div>
        <label for="municipio">Municipio:</label>
        <select id="municipio" name="municipio" required>
            <option>-</option>
        </select>
    </div>
</div>
<div class="linha">
    <button type="reset">Limpar</button>
    <button id="btEnviar" type="submit" disabled>Enviar</button>
</div>
</fieldset>
</form>
</section>
</main>

<footer>
```

```
<p>&copy; 20XX Minha Página. Todos os direitos reservados.</p>
</footer>
</body>
</html>
```

Essa página é semelhante à criada no laboratório 3. E no final, devemos ter um resultado semelhante ao apresentado na figura abaixo.

Criando o CSS:

1 – Crie na pasta lab10 um arquivo chamado **style** com a extensão **.css** (**style.css**) com o conteúdo abaixo.

```
html, body {
    height: 100%;
    margin: 0;
    padding: 0;
    background-color: white;
    font-family: Arial, sans-serif;
    line-height: 1.6;
}

header,
footer {
    flex-shrink: 0;
    text-align: center;
    background: #89CFF0;
    color: white;
    padding: 20px;
}

footer p {
    margin: 0;
}

main {
    height: auto;
    overflow: auto;
    display: flex;
    flex-wrap: wrap;
    align-items: center;
    justify-content: space-between;
    margin: auto;
```

```
    gap: 5px;
    padding: 20px;
}

section {
    margin-top: 20px;
}

form {
    display: flex;
    align-items: center;
    font-weight: bold;
}

label {
    margin-bottom: 5px;
}

fieldset {
    width: 80%;
    margin: 0 auto;
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
}

input,
select {
    box-sizing: border-box;
    width: 100%;
    padding: 8px;
    border: 1px solid #ccc;
    border-radius: 5px;
    background-color: white;
}

input:disabled {
    background-color: lightgray;
}

div {
    width: 100%;
    margin-bottom: 20px;
}
```

```

.linha {
  display: flex;
  align-items: center;
  justify-content: space-between;
  gap: 5px;
}

button {
  display: block;
  width: 100%;
  padding: 10px;
  margin-top: 20px;
  background-color: #4CAF50;
  color: white;
  border: none;
  border-radius: 5px;
  font-size: 16px;
  cursor: pointer;
  text-align: center;
}

button:hover {
  background-color: #45a049;
}

button:disabled {
  cursor: auto;
  background-color: gray;
}

#formulario {
  height: 60vh; /* Define a altura do main para ocupar a altura total da
viewport */
  flex: 1;
  background: lightcyan;
  padding: 20px;
  border: 1px solid #ccc;
  border-radius: 5px;
}

```

Criando o JavaScript:

1 – Crie na pasta lab10 um arquivo chamado **form** com a extensão **.js (form.js)**. Vamos adicionar ao cabeçalho o link para esse arquivo JavaScript.

```

<head>
  <meta charset="utf-8">
  <title>Página Formulário</title>
  <link rel='stylesheet' type='text/css' href='style.css'>

  <script src="form.js"></script>
</head>

```

2 – No arquivo **form.js** vamos criar nossa primeira função para validar o campo cpf do formulário. O algoritmo para validar o CPF é o seguinte:

- O CPF deve possuir 11 dígitos;
- Todos os dígitos não podem ser iguais;
- Realizamos o cálculo do primeiro dígito verificador:
 - Pegamos os primeiros 9 dígitos e multiplicamos cada um dos números, da direita para a esquerda, por números crescentes a partir do número 2.
Ex.: $111.444.777-35 = 1*10 + 1*9 + 1*8 + 4*7 + 4*6 + 4*5 + 7*4 + 7*3 + 7*2 = 162$
 - Pegamos o resultado, dividimos por 11 e pegamos o resto da divisão
 - Se ela for menor que 2, então o dígito é igual a 0
 - Se ela for maior ou igual a 2, então o dígito verificador é igual a 11 menos o resto da divisão
Ex.: $162 \% 11 = 8 \rightarrow 11 - 8 = 3$
- Realizamos o cálculo do segundo dígito verificador:
 - Pegamos os primeiros 9 dígitos junto com o primeiro dígito verificador calculado.
 - Aplicamos a multiplicação semelhante ao cálculo do primeiro dígito
Ex.: $111.444.777-35 = 1*11 + 1*10 + 1*9 + 4*8 + 4*7 + 4*6 + 7*5 + 7*4 + 7*3 + 3*2 = 204$
 - Pegamos o resultado, dividimos por 11 e consideramos o resto da divisão
 - Se o resto for menor que 2, então o dígito é igual a 0
 - Senão, o dígito verificador é igual a 11 menos o resto da divisão
Ex.: $204 \% 11 = 6 \rightarrow 11 - 6 = 5$

DICA: Será necessário utilizar a função *getElementById()* para pegar o valor digitado pelo usuário.

3 – Adicione sua função de validar cpf no campo cpf do formulário. Vamos fazer isso no evento blur.

```

<div>
  <label for="cpf">CPF:</label>
  <input type="text" id="cpf" name="cpf" required onblur="validarCPF()">
</div>

```

4 – Vamos criar agora uma função para selecionar os municípios de acordo com o estado selecionado.

Note que estamos manipulando o DOM para buscar os elementos que desejamos verificar e/ou adicionar informações.

```
function seleccioneMunicipio() {
  let estado = document.getElementById("estado").value;

  let municipioElement = document.getElementById("municipio");
  removeOptions(municipioElement);
  addOption(municipioElement, "-", "-");
  switch (estado) {
    case "bahia":
      addOption(municipioElement, "salvador", "Salvador");
      addOption(municipioElement, "feira_santana", "Feira de Santana");
      break;
    case "sao_paulo":
      addOption(municipioElement, "sao_paulo", "São Paulo");
      addOption(municipioElement, "adamantina", "Adamantina");
      break;
    case "rio_janeiro":
      addOption(municipioElement, "rio_janeiro", "Rio de Janeiro");
      addOption(municipioElement, "niteroi", "Niterói");
      break;
    case "tocantins":
      addOption(municipioElement, "palmas", "Palmas");
      addOption(municipioElement, "araguaína", "Araguaína");
      break;
    default:
      break;
  }
}
```

4 – Para termos um código mais simples encapsulamos as funções para adicionar e remover as opções.

```
function removeOptions(selectElement) {
  for(let i = (selectElement.options.length - 1); i >= 0; i--) {
    selectElement.remove(i);
  }
}

function addOption(selectElement, value, text) {
  let option = document.createElement("option");
  option.value = value;
  option.text = text;
  selectElement.add(option);
}
```

5 – Por fim, vamos adicionar a função no evento change do elemento select dos estados.

```
<select id="estado" name="estado" required onchange="seleccioneMunicipio()">
```

6 – Vamos criar uma função para validar os campos obrigatórios do nosso formulário.

```
function verificarObrigatorios() {  
    let nome = document.getElementById("nome").value;  
    let cpf = document.getElementById("cpf").value;  
  
    // Falsy value  
    if (!nome || !cpf) {  
        document.getElementById("btEnviar").disabled = true;  
    } else {  
        document.getElementById("btEnviar").disabled = false;  
    }  
}
```

7 – Agora vamos colocar a função de verificar obrigatórios para ser executada periodicamente.

Para isso, vamos usar a função *setInterval()* que serve para executar uma função repetidamente, em intervalos de tempo constantes, medidos em milissegundos (ms).

```
function aoCarregar() {  
    setInterval(verificarObrigatorios, 1000);  
}
```

8 – Para finalizar, precisamos colocar a função *aoCarregar()* no evento de load da página HTML.

```
<body onload="aoCarregar()">
```

Submit:

1 – Crie na pasta chamada lab10 o arquivo chamado **form-submit** com a extensão **.html** (**form-submit.html**). Neste arquivo será criada nossa página HTML. Conforme a figura abaixo:

```
<html lang="pt-BR">  
  
<head>  
    <meta charset="utf-8">  
    <title>Página Formulário</title>  
    <link rel='stylesheet' type='text/css' href='style.css'>  
    <script src="form.js"></script>  
</head>
```



```
<body>
  <header>
    <h1>Meu Formulário</h1>
  </header>

  <main>
    <section id="formulario">
      <form action="form.html" method="get">
        <fieldset>
          <legend>Dados Recebidos</legend>
          <div>
            <label for="nome">Nome:</label>
            <input type="text" id="nome" name="nome" disabled>
          </div>
          <div class="linha">
            <div>
              <label for="nomeMae">Nome da mãe:</label>
              <input type="text" id="nomeMae" name="nomeMae" disabled>
            </div>
            <div>
              <label for="nomePai">Nome do pai:</label>
              <input type="text" id="nomePai" name="nomePai" disabled>
            </div>
          </div>
          <div class="linha">
            <div>
              <label for="cpf">CPF:</label>
              <input type="text" id="cpf" name="cpf" disabled>
            </div>
            <div>
              <label for="email">Email:</label>
              <input type="email" id="email" name="email" disabled>
            </div>
            <div>
              <label for="estadoCivil">Estado Civil:</label>
              <input type="text" id="estadoCivil" name="estadoCivil"
disabled>
            </div>
          </div>
          <div class="linha">
            <div>
              <label for="estado">Estado:</label>
              <input type="text" id="estado" name="estado" disabled>
            </div>
          </div>
        </form>
      </section>
    </main>
  </body>
```

```

        <label for="municipio">Municipio:</label>
        <input type="text" id="municipio" name="municipio" disabled>
    </div>
</div>
<div class="linha">
    <button id="btEnviar" type="submit">Voltar</button>
</div>
</fieldset>
</form>
</section>
</main>

<footer>
    <p>&copy; 20XX Minha Página. Todos os direitos reservados.</p>
</footer>
</body>

</html>

```

Note que o código é muito semelhante ao código HTML anterior. A principal diferença é que os campos estão desabilitados pela propriedade *disabled*.

2 – No arquivo JavaScript, vamos criar a função para carregar as informações que foram preenchidas no formulário anterior.

```

function dadosRecebidos() {
    const urlParams = new URLSearchParams(window.location.search);
    document.getElementById('nome').value = urlParams.get('nome');
    document.getElementById('nomeMae').value = urlParams.get('nomeMae');
    document.getElementById('nomePai').value = urlParams.get('nomePai');
    document.getElementById('email').value = urlParams.get('email');
    document.getElementById('cpf').value = urlParams.get('cpf');
    document.getElementById('estadoCivil').value = urlParams.get('estadoCivil');
    document.getElementById('estado').value = urlParams.get('estado');
    document.getElementById('municipio').value = urlParams.get('municipio');
}

```

A função *URLSearchParams()* é utilizada para ler os parâmetros da URL (ou seja, os dados após o ? na URL).

3 – Vamos colocar no evento de load da página para carregar essas informações recebidas do formulário.

```

<body onload="dadosRecebidos()">

```

4 – Para testar, preencha os dados na página **form.html** e clique no botão de enviar. Ao fazer isso, será verificado qual a *action* que está no elemento *form* e dispara a requisição de acordo com essa *action*.