

HTML: HyperText Markup Language

Objetivo do HTML:

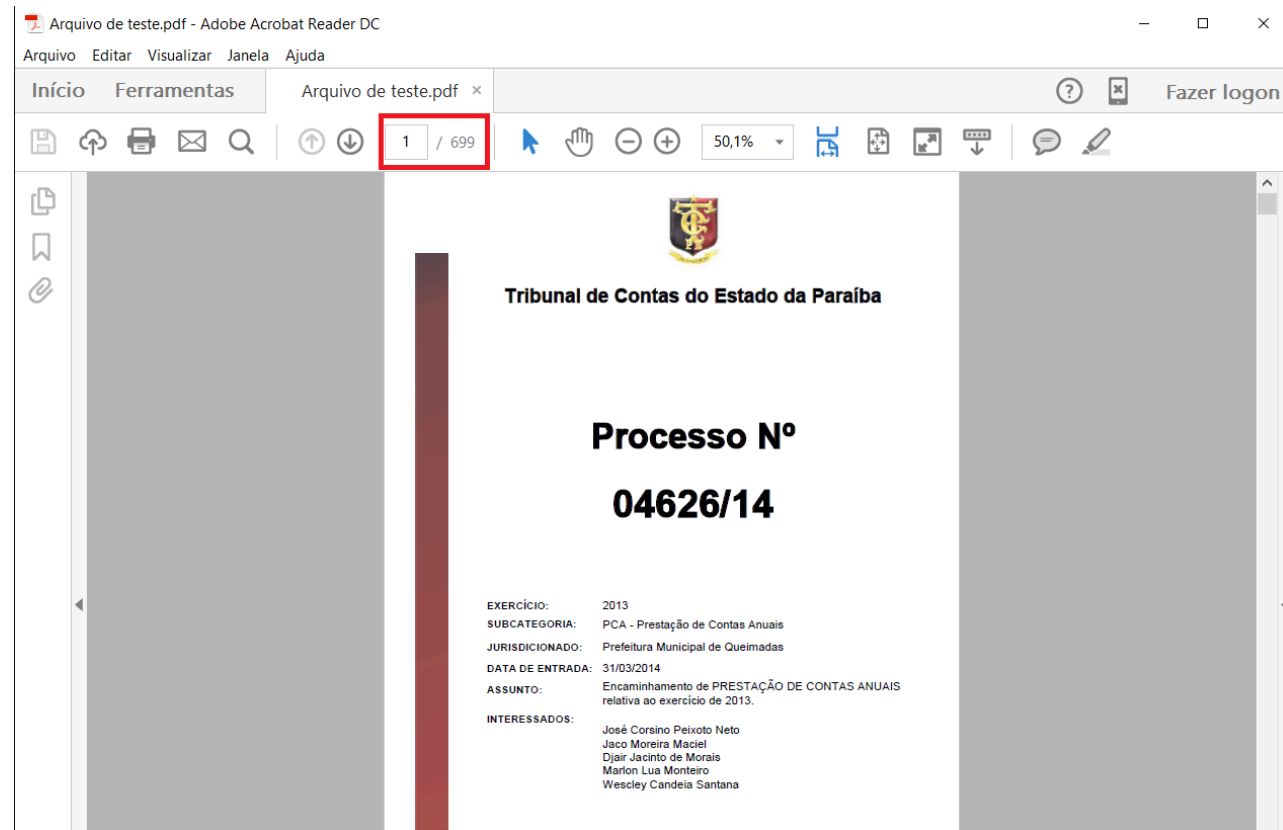
- Formatar texto
- Inserir marcações de formatação em texto simples
- Definir a estrutura de um documento web

Usos além da web:

- E-mails formatados
- Relatórios formatados
- Documentos de texto

Exemplo de outros documentos formatados

- PS (PostScript)
- PDF (Portable Document Format)
- DOCX (Microsoft Word)
- ODT (Open Document Text)
- RTF (Rich Text Format)



Estrutura Básica

```
<!DOCTYPE html>  
<html>  
  <head>  
  
  </head>  
  <body>  
  
  </body>  
</html>
```

Estrutura principal

Cabeçalho

Corpo

<HTML>

<HEAD>

</HEAD>

<BODY>

</BODY>

</HTML>

Cabeçalho

Contém informações sobre o documento:

- Título
- Metadados
- Links para arquivos externos
- Scripts

```
<head>  
  <title>Meu primeiro site</title>  
  <meta charset="UTF-8">  
  <link rel="stylesheet" href="style.css">  
  <script src="script.js"></script>  
</head>
```

Corpo

Contém o conteúdo do documento:

- Texto
- Imagens
- Vídeos
- Links
- Listas
- Tabelas
- Formulários

```
<body>  
  <h1>Meu primeiro site</h1>  
  <p>Este é um parágrafo.</p>  
    
  <a href="https://www.google.com">Google</a>  
</body>
```

Marcações (Tags)

Tags são os códigos necessários para construção das paginas web.

- As tags são delimitadas por "<" e ">"
- A maioria das tags possui uma tag de abertura e uma de fechamento
- As tags de fechamento possuem uma barra antes do nome da tag
 - Exemplo: `<p>Texto</p>`
- Algumas tags não possuem conteúdo e são auto-fecháveis
- Possuem atributos que são informações adicionais:
 - Exemplo: ``
- Atributos `id` e `class` são utilizados para identificar elementos
 - Exemplo: `<p id="conteudo" class="principal">`

Principais Tags

- `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>` : Títulos
- `<p>` : Parágrafo
- `<a>` : Link
- `` : Imagem
- ``, ``, `` : Listas
- `<hr>`, `
` : Linha horizontal, quebra de linha
- ``, ``, `<u>`, `<s>` : Negrito, itálico, sublinhado, riscado
- `<table>`, `<tr>`, `<td>`, `<th>` : Tabelas
- `<form>`, `<input>`, `<button>`, `<select>`, `<textarea>` : Formulários
- `<div>`, `` : Divisões
- `<header>`, `<footer>`, `<nav>`, `<section>`, `<article>`, `<aside>` : Seções

Tabela

A tag `table` é utilizada para criar tabelas:

- as tags `<tr>` são utilizadas para criar linhas
- as tags `<td>` são utilizadas para criar células
- as tags `<th>` são utilizadas para criar cabeçalhos

```
<table>
  <tr> <!-- Linha de cabeçalho -->
    <th>Nome</th>
    <th>Idade</th>
  </tr>
  <tr> <!-- 1ª Linha de dados -->
    <td>João</td>
    <td>25</td>
  </tr>
  <tr> <!-- 2ª Linha de dados -->
    <td>Maria</td>
    <td>30</td>
  </tr>
</table>
```


Tabela

```
<table>
  <tr> <!-- Linha de cabeçalho -->
    <th>Nome</th>
    <th>Idade</th>
  </tr>
  <tr> <!-- 1ª Linha de dados -->
    <td>João</td>
    <td>25</td>
  </tr>
  <tr> <!-- 2ª Linha de dados -->
    <td>Maria</td>
    <td>30</td>
  </tr>
</table>
```

Nome	Idade
João	25
Maria	30

Formulário

A tag `<form>` é utilizada para criar formulários:

- Os formulários são utilizados para enviar dados para o servidor
- as tags `<input>`, `<select>`, `<textarea>` são utilizadas para criar campos de entrada
- a tag `<button>` é utilizada para criar botões

```
<form>
  <label>Nome:</label>
  <input type="text" name="nome">

  <label for="email">E-mail:</label>
  <input type="email" id="email" name="email">

  <button type="submit">Enviar</button>
</form>
```

Formulário

```
<form>
  <label>Nome:</label>
  <input type="text" name="nome">

  <label for="email">E-mail:</label>
  <input type="email" id="email" name="email">

  <button type="submit">Enviar</button>
</form>
```

Nome:

E-mail:

Enviar

Links

A tag `<a>` é utilizada para criar links:

- O atributo `href` é utilizado para definir o destino do link
- O atributo `target` é utilizado para definir onde o link será aberto
 - `_self` : Abre na mesma janela
 - `_blank` : Abre em uma nova janela

```
<a href="https://www.google.com" target="_blank">Google</a>
```

```
<a href="galeria.html">  
    
</a>
```

Listas

A tag `` é utilizada para criar listas não ordenadas

A tag `` é utilizada para criar listas ordenadas

- As tags `` são utilizadas para criar itens da lista

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>
```

```
<ol>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ol>
```

Listas

Lista 1:

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>
```

Lista 2:

```
<ol>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ol>
```

Lista 1:

- Item 1
- Item 2
- Item 3

Lista 2:

1. Item 1
2. Item 2
3. Item 3

Primeiros passos: Index

1. Crie um arquivo HTML com a estrutura básica:
chamado `index.html` na pasta do projeto 'src/main/resources/templates/'

```
<!DOCTYPE html>
<html>
  <head>
    <title>UEG - Desenvolvimento Web 2025</title>
  </head>
  <body>
    <h1>Início</h1>
    <p>
      Bem-vindo ao curso de Desenvolvimento Web!
    </p>
  </body>
</html>
```

2. Abra o arquivo no navegador

Primeiros passos: alunos.html

1. Crie um arquivo chamado `alunos.html` na pasta do projeto `'src/main/resources/templates/'`
 - Título: UEG - Desenvolvimento Web 2025: Alunos
 - H1: Lista de Alunos
 - Tabela com 3 colunas: Nome, Matrícula e Ações
 - 3 linhas com informações fictícias
 - Nas linhas de dados na coluna Ações:
 - Criar uma lista não ordenada de dois itens:
 - Editar
 - Excluir
- Editar e Excluir devem ser links para as páginas `aluno-update.html` e `aluno-delete.html`

Primeiros passos: aluno-create.html

1. Crie um arquivo chamado `aluno-create.html` na pasta do projeto 'src/main/resources/templates/'
 - Título: UEG - Desenvolvimento Web 2025: Aluno
 - H1: Cadastro de Aluno
 - Formulário com os campos:
 - Nome
 - Matrícula
 - E-mail
 - Botão de Salvar

Primeiros passos: aluno-update.html

1. Crie um arquivo chamado `aluno-update.html` na pasta do projeto 'src/main/resources/templates/'
 - Título: UEG - Desenvolvimento Web 2025: Aluno
 - H1: Atualização Cadastro de Aluno
 - Formulário com os campos:
 - Nome
 - Matrícula
 - E-mail
 - Botão de Salvar

Primeiros passos: aluno-delete.html

1. Crie um arquivo chamado `aluno-delete.html` na pasta do projeto 'src/main/resources/templates/'
 - Título: UEG - Desenvolvimento Web 2025: Aluno
 - H1: Exclusão de Aluno
 - Mensagem de confirmação de exclusão do aluno, texto destacado.
 - Formulário com o campo:
 - Nome do aluno a ser excluído (não editável)
 - E-mail do aluno a ser excluído (não editável)
 - Botão de Excluir

Amarrando as páginas

1. Adicione links entre as páginas

- Na página `index.html` adicione um link para a página `alunos.html`
- Na página `alunos.html` adicione um link para a página `aluno-create.html`
- Nas paginas `aluno-update.html` e `aluno-delete.html` adicione um link para a página `alunos.html`

Mocando os dados

Iremos criar uma classe de controle para simular o acesso ao banco de dados.

Acesso ao banco de dados será abordado em aulas futuras.

AlunoController

Criar o arquivo "AlunoController.java" na pasta "src/main/java/ueg/.../controller/"

```
import java.util.*;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;

@Controller
public class AlunoController {

    static List alunos = new ArrayList<>();

    static {
        alunos.add(Map.of("nome", "João", "email", "joao@localhost"));
        alunos.add(Map.of("nome", "Maria", "email", "maria@localhost"));
        alunos.add(Map.of("nome", "José", "email", "jose@localhost"));
    }

    @GetMapping("/")
    public String index() {
        return "index";
    }

    @GetMapping("/alunos")
    public String getHome(Model model) {
        model.addAttribute("alunos", alunos);
        return "alunos";
    }
}
```

Thymeleaf em ação

1. Adicione o Thymeleaf no arquivo HTML

- Adicione o namespace `xmlns:th="http://www.thymeleaf.org"`
- Utilize a sintaxe do Thymeleaf para exibir os dados
- Modifique a tabela de alunos para exibir os dados da lista

```
<tr>
  <td>Fulano de Tal</td>
  <td>fulano@ueg.br</td>
  <td>
    <ul>
      <li>
        <a href="aluno-update.html">Editar</a>
      </li>
      <li>
        <a href="aluno-delete.html">Excluir</a>
      </li>
    </ul>
  </td>
</tr>
```

Thymeleaf em ação

```
<table>
...
<tr th:each="aluno, iter: ${alunos}">
  <td th:text="${aluno.nome}">Fulano de Tal</td>
  <td th:text="${aluno.email}">fulano@ueg.br</td>
  <td>
    <ul>
      <li>
        <a th:href="@{'/alunos/editar/' + ${iter.index}}"
          href="aluno-update.html">Editar</a>
      </li>
      <li>
        <a th:href="@{'/alunos/excluir/' + ${iter.index}}"
          href="aluno-delete.html">Excluir</a>
      </li>
    </ul>
  </td>
</tr>
</table>
```


Testar o resultado

1. Inicie a aplicação
2. Acesse a página `http://localhost:8080/alunos`
3. Verifique se a lista de alunos é exibida corretamente
4. Compare com o arquivo `alunos.html` criado anteriormente

Metodo de criação de aluno

```
@GetMapping("/alunos/cadastrar")
public String getForm() {
    return "aluno-create";
}

@PostMapping("/alunos/cadastrar")
public String postForm(
    @RequestParam String nome,
    @RequestParam String email) {
    System.out.println("Nome: " + nome);
    System.out.println("Email: " + email);
    // Adiciona um novo aluno na lista
    alunos.add(Map.of("nome", nome, "email", email));
    return "redirect:/alunos";
}
```

Tela de cadastro de aluno

Alterar o formulário de cadastro de aluno para enviar os dados para o servidor

```
<form action="/alunos/cadastrar" method="post">
```

Testar o resultado

1. Acesse a página `http://localhost:8080/alunos/cadastrar`
2. Preencha o formulário e clique em Salvar
3. Verifique se o aluno foi adicionado na lista de alunos
4. Compare com o arquivo `alunos.html` criado anteriormente

Estilizando com CSS

Na proxima aula vamos aprender a estilizar as páginas com CSS.

Pesquisar sobre os principais frameworks de CSS:

- Bootstrap: <https://getbootstrap.com/>
- Materialize: <https://materializecss.com/>
- Bulma: <https://bulma.io/>
- Foundation: <https://get.foundation/>

Escolher um para utilizar no projeto ao longo do curso, justificar a escolha.

