

# Introdução a Linguagens de Marcação

## Desenvolvimento Web I

Ricardo de la Rocha Ladeira  
{ricardo.ladeira@ifc.edu.br}



**INSTITUTO FEDERAL**  
Catarinense  
Campus Blumenau

# Linguagens de Marcação

- ▶ Uma linguagem de marcação (*markup language*) é uma linguagem computacional capaz de descrever e estruturar um documento.
- ▶ A marcação ocorre por meio de marcadores (*tags*), que funcionam para atribuir algum significado aos dados.
- ▶ Algumas (meta)linguagens de marcação conhecidas:
  - ▶ LaTeX
  - ▶ SGML
  - ▶ HTML
  - ▶ XML
  - ▶ Markdown
  - ▶ SVG

# SGML

- ▶ SGML significa ***Standard Generalized Markup Language***.
- ▶ É uma meta-linguagem que descreve linguagens de marcação, entre elas
  - ▶ HTML
  - ▶ XML

# XML

- ▶ XML significa ***eXtensible Markup Language***.
- ▶ É uma linguagem de marcação para representação, armazenamento e transporte de dados. Foi desenvolvida para ser autodescritiva e é uma recomendação da W3C desde 1998 (W3SCHOOLS, 2020).
- ▶ Aplicações
  - ▶ Padrão utilizado por ferramentas como Microsoft Office e Google Docs.
  - ▶ Serialização de objetos.
  - ▶ NF-e
  - ▶ PDF

# XML

- ▶ Suas *tags* **não são predefinidas**, o desenvolvedor pode criar *tags* conforme sua necessidade, seguindo as regras de aceitação de um código XML válido.
- ▶ Seus documentos têm **estrutura de árvore**, iniciando por **uma raiz** (somente uma!) e ramificando para filhos, subfilhos (netos) etc.
- ▶ Os elementos podem ter conteúdo e atributos.
- ▶ Todo elemento XML tem *tags* de abertura e fechamento:  
`<comida></comida>`

# XML

- ▶ O XML prolog, se presente, deve estar na primeira linha, garantindo que não ocorrerão problemas com acentuação:

```
<?xml version="1.0"encoding="UTF-8"?>
```

- ▶ As *tags* XML são **case sensitive**, ou seja, diferencia letras maiúsculas de minúsculas.

- ▶ Os elementos precisam estar corretamente aninhados. É errado:

```
<?xml version="1.0" encoding="UTF-8"?>  
<carro>  
  <nome>Polo</nome>  
  <motor><versao>TSI 100</motor></versao>  
</carro>
```

# XML

- Valores de atributos precisam sempre estar entre aspas.

```
<?xml version="1.0" encoding="UTF-8"?>
<carro tipo="popular">
  <nome>Polo</nome>
  <motor><versao>TSI 100</motor></versao>
</carro>
```

# XML

## Exemplo

```
<?xml version="1.0" encoding="UTF-8"?>
<mensagem>
  <de>Tove</de>
  <para>Jani</para>
  <assunto>Reminder</assunto>
  <corpo>Don't forget me this weekend!</corpo>
</mensagem>
```

Fonte: adaptado do exemplo de W3SCHOOLS (2020) disponível em <https://www.w3schools.com/xml/note.xml>.



## XML/XHTML

- ▶ Códigos XML pode ser estilizados por uma tecnologia chamada XSLT (*EXtensible Stylesheet Language Transformations*).
- ▶ A partir do XML, houve uma tentativa de se utilizar a tecnologia XHTML, unindo aspectos do XML com o HTML. Esta tecnologia ainda faz parte das recomendações da W3C, mas vem sendo cada vez menos vista.

# HTML

## Contextualização

- ▶ HTTP foi criado para permitir o compartilhamento de documentos científicos.
- ▶ Mas compartilhando texto, como indicar o que é uma tabela? Um título? O que precisa ser enfatizado? O que é uma referência? O que é um parágrafo?

# HTML

- ▶ HTML: *HyperText Markup Language* (Linguagem de Marcação de Hipertexto).
- ▶ Linguagem de marcação para páginas *web*.
- ▶ Resolve o problema citado.

# HTML

- ▶ Linguagem universal, aceita pelos navegadores mais conhecidos.
- ▶ Não é linguagem de **programação**, mas de **marcação**.
- ▶ Marca os objetos da página. Imagem, título, parágrafo, rodapé...

# HTML

- ▶ HTML é um conjunto de códigos de marcação.
- ▶ Navegadores implementam os códigos de marcação (*tags*). Se o nome da *tag* estiver incorreto, poderá resultar em erros na página.
  - ▶ Estes erros são ditos **sintáticos**.
- ▶ *Tag*: etiqueta.
  - ▶ *Tag* de abertura: `<body>`
  - ▶ *Tag* de fechamento: `</body>`

# HTML

- ▶ *Tags* têm um significado associado. Chamamos este significado de **semântica**.
- ▶ O significado da *tag* `<p>` é: todo conteúdo marcado por esta *tag* representa um parágrafo.
- ▶ Assim, se escrevermos `<p>Notícias</p>`, estamos cometendo um erro semântico. Por quê?
- ▶ O valor semântico correto é muito importante (discutiremos mais adiante na disciplina).

# HTML

## Primeiro website

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <title>Meu primeiro website</title>
  <meta charset="utf-8">
</head>
<body>
  <h1>Joao da Silva</h1>
  <p>Esta e a pagina oficial de Joao da Silva.
    </p>
</body>
</html>
```

# HTML

## Primeiro website

- ▶ Digitar no bloco de notas (notepad), mousepad ou gedit.
- ▶ Salvar como... `index.htm` ou `index.html`



# HTML

- ▶ Tags: `<html>`, `<body>`, `<p>`...
- ▶ Tags de abertura: `<html>`, `<body>`, `<p>`...
- ▶ Tags de fechamento: `</html>`, `</body>`, `</p>`...

# HTML

## !DOCTYPE

- ▶ Não é uma tag HTML.
- ▶ Comando utilizado pelos navegadores (*browsers*).
- ▶ Devemos acrescentar sempre em nossos arquivos.

# HTML

## Tags

- ▶ `<html>` Todo website é escrito entre essas tags. `</html>`
- ▶ Entre elas: `<head>` `</head>` e `<body>` `</body>`
- ▶ Nessa ordem: head body (cabeça corpo, como se fosse um corpo humano)

# HTML

## head

- ▶ head: parte inteligente do site.
- ▶ Veremos melhor mais adiante.
- ▶ Pode conter dados sobre o documento (metadados).

# HTML

## head

```
► <head>  
  <title>Título da página</title>  
  </head>
```

# HTML

## body

- ▶ body: conteúdo visível do documento.
- ▶ Contém:
  - ▶ links
  - ▶ textos
  - ▶ imagens
  - ▶ tabelas
  - ▶ listas
  - ▶ formulários
  - ▶ ...

# HTML

- Tag `<p>`: cria parágrafos.

```
<!DOCTYPE html>
<html lang="pt-br">
<head><title>Meu website</title></head>
<body>
  <p>Esta e a pagina de Joao da Silva.</p>
  <p>Outro paragrafo na pagina. Posso fazer
    uma introducao aqui. Seja bem-vindo ao
    website XXX. Aqui voce vai encontrar tudo
    sobre YYY.</p>
  <p>Aqui posso explicar que a ideia de criar
    o site surgiu em 2014, quando... </p>
</body>
</html>
```

# HTML

- ▶ "Esta é a página de João da Silva" parece mais um parágrafo ou um título?
- ▶ Tags de títulos/cabeçalhos na página: `<h1>`, `<h2>`, ..., `<h6>`



# HTML

```
<!DOCTYPE html>
<html lang="pt-br">
<head><title>Meu website</title></head>
<body>
  <h1>Esta e a pagina de Joao da Silva.</h1>
  <p>Outro paragrafo na pagina. Posso fazer
    uma introducao aqui. Seja bem-vindo ao
    website XXX. Aqui voce vai encontrar tudo
    sobre YYY.</p>
  <p>Aqui posso explicar que a ideia de criar
    o site surgiu em 2014, quando... </p>
</body>
</html>
```

# HTML

- ▶ Troque as tags `<h1>` `</h1>` por `h2`, `h3`... `h6` e observe as diferenças.

# HTML

## Elementos vazios

- Algumas tags não possuem fechamento.

```
<img>
```

```
<br>
```

```
<hr>
```

```
<meta>
```

```
...
```

# HTML

## Elementos vazios

- ▶ `<br>`: quebra de linha.
- ▶ `<img>`: insere uma imagem.
- ▶ `<hr>`: insere um traço horizontal na página.

# HTML

## Elementos vazios

- ▶ Muitas vezes são escritos como neste exemplo:

- ▶ `<br />`

# HTML

```
<!DOCTYPE html>
<html lang="pt-br">
<body>
  <hr>
  <h2>Esta e a pagina de Joao da Silva.</h2>
  <hr>
  <p>Primeiro paragrafo da minha pagina.<br>
    Posso fazer uma apresentacao da pagina
    aqui.</p>
  <p>Este e meu segundo paragrafo. Aqui posso
    explicar que a ideia de criar o site
    surgiu em meados de 2014, quando...</p>
</body>
</html>
```

# HTML

- ▶ Exemplo de exercício:
- ▶ `<p>Meu parágrafo</p>`
- ▶ Identifique a tag de início, a tag de fim e o conteúdo da tag.

# HTML

## Comentarios

► `<!-- - Isto é um comentário - ->`



# HTML

## Curiosidade

- ▶ Em qualquer *browser* podemos ver o código-fonte das páginas.
- ▶ Podemos ver as *tags* que são usadas.

# HTML

## Atributos

- ▶ tags HTML podem ter atributos
- ▶ `<html lang="pt-br">`

# Exercício

Crie esta página web

**Estudante do IFC cria sua primeira pagina web**

**Ato foi executado durante a aula**

---

Escrevi alguma coisa.

E aqui um pouco mais.

---

# Exercício

## Resposta

```
<!DOCTYPE html>
<html lang="pt-br">
<body>
  <h1>Estudante do IFC cria sua primeira
    pagina web</h1>
  <h4>Ato foi executado durante a aula</h4>
  <hr>
  <p>Escrevi alguma coisa.</p>
  <br><br>
  <p>E aqui um pouco mais.</p>
  <hr>
</body>
</html>
```

## Exercício

- ▶ Havia elementos vazios na página? Se sim, quais?
- ▶ Havia elementos com atributos? Se sim, quais?

# HTML

- ▶ Linguagem de marcação.
- ▶ Marca os objetos da página. Imagem, título, parágrafo, rodapé...
- ▶ Arquivo `.html` ou `.htm`.

# HTML

## Primeiro website

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <title>Meu primeiro website</title>
</head>
<body>
  <h1>Joao da Silva</h1>
  <p>Esta e a pagina oficial de Joao da Silva.
    </p>
</body>
</html>
```

# HTML

- ▶ Tags: `<html>`, `<body>`, `<p>`...
- ▶ Tags de abertura: `<html>`, `<body>`, `<p>`...
- ▶ Tags de fechamento: `</html>`, `</body>`, `</p>`...



# HTML

```
<!DOCTYPE html>
<html lang="pt-br">
<body>
  <hr>
  <h1>Esta e a pagina de Joao da Silva.</h1>
  <hr>
  <p>Primeiro paragrafo da minha pagina.<br>
    Posso fazer uma apresentacao da pagina
    aqui.</p>
  <p>Este e meu segundo paragrafo. Aqui posso
    explicar que a ideia de criar o site
    surgiu em meados de 2014, quando...</p>
</body>
</html>
```

# HTML

## Atributos

- ▶ tags HTML podem ter atributos
- ▶ `<html lang="pt-br">`

# HTML

## Links

- ▶ Ligação com outra página, arquivo ou algum ponto da página atual.
- ▶ *tag* `<a>`.

# HTML

## Links

- Acrescentem em algum arquivo HTML:

```
<a href="http://www.google.com.br">Google</a>
```

# HTML

## Links

- ▶ Abrindo em outra aba:

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

# HTML

## Links

- ▶ Exercício:
- ▶ Crie uma página que contenha título, no mínimo um parágrafo e um link para a página do IFC Blumenau. A frase que servirá como link deve ser `Clique aqui para entrar na página do IFC Blumenau.`

# HTML

## Links

- ▶ *Link* para um ponto da própria página:
- ▶ Marca-se o ponto da página que será o destino do clique (para que ponto da página quero ir ao clicar no *link*?) acrescentando ao elemento o parâmetro `id`:

```
<h1 id="topo">Seja bem-vindo a este site</h1>
```

- ▶ Marca-se o *link* colocando no parâmetro `href` o valor do `id` do elemento destino, com a cerquilha/*hashtag* (`#`) na frente:

```
<p>Fim da pagina. <a href="#topo">Clique aqui  
e va para o topo da pagina</a></p>
```

# HTML

## Listas

- ▶ Podemos criar marcadores ordenados e não ordenados em HTML.
- ▶ O marcador não ordenado é representado pela *tag* `<ul>`.
- ▶ O marcador ordenado é representado pela *tag* `<ol>`.
- ▶ Os itens de listas ordenadas e não ordenadas são representados pela *tag* `<li>`.
- ▶ Quando devo usar uma lista ordenada ou não ordenada?
  - ▶ Lista de praias de Santa Catarina
  - ▶ Classificação do Campeonato Brasileiro de Futebol
  - ▶ Ranking de praias mais poluídas do mundo
  - ▶ Resultado final de votação das chapas das eleições do Grêmio Estudantil



# HTML

## Listas

- Exemplo de lista **não ordenada**:

```
<h1>Praias de Santa Catarina:</h1>
<ul>
  <li>Quatro Ilhas</li>
  <li>Zimbros</li>
  <li>Mariscal</li>
  <li>Canto Grande</li>
</ul>
```

# HTML

## Listas

- Exemplo de lista **ordenada**:

```
<h1>Classificacao do Campeonato:</h1>
<ol>
  <li>Gremio</li>
  <li>Sao Paulo</li>
  <li>Corinthians</li>
  <li>Flamengo</li>
</ol>
```

# HTML

## Imagens

- ▶ Tag `<img>`.
- ▶ Atributo `src` indicando o caminho e o nome da imagem.
- ▶ Atributo `alt` indicando o nome alternativo da imagem.

```

```

- ▶ A imagem pode estar na web ou armazenada localmente (no seu computador).

# HTML

## Imagens

- ▶ Problemas comuns no carregamento de imagens:
  - ▶ Nome do arquivo incorreto (exemplo: `imagem.jpg`  $\neq$  `imagem.JPG`).
  - ▶ Caminho incorreto informado (exemplo:  
`C:\Users\imagem.jpg`  $\neq$  `C:\imagem.jpg`  $\neq$  `imagem.jpg`).
- ▶ Boas práticas
  - ▶ Criar uma pasta dentro do projeto (`img`) e colocar lá todas as imagens.
  - ▶ Utilizar, no parâmetro `src`, o **caminho relativo** para acessar a imagem.

# HTML

## Imagens

- ▶ Mas como referenciar a imagem se ela estiver em uma pasta diferente do arquivo HTML?
- ▶ Há duas possibilidades: utilizar o **caminho absoluto** ou o **caminho relativo**.

# HTML

## Imagens

- ▶ Referenciando com o **caminho absoluto**:
- ▶ Crie um arquivo HTML, que contenha a *tag* `img`, na Área de Trabalho.
- ▶ Crie uma pasta chamada `img` na Área de Trabalho e insira nesta pasta a imagem do seu arquivo HTML.
- ▶ No atributo `src`, utilize o caminho completo do arquivo. Exemplos:
  - ▶ Windows: `C:\Users\joao\Desktop\img\Imagem.jpg`
  - ▶ Linux: `/home/joao/Área de Trabalho/img/Imagem.jpg`

# HTML

## Imagens

- ▶ Referenciando com o **caminho relativo**:
- ▶ Crie um arquivo HTML, que contenha a *tag* `img`, na Área de Trabalho.
- ▶ Crie uma pasta chamada `img` na Área de Trabalho e insira a imagem do seu arquivo HTML.
- ▶ No atributo `src`, digite “img/Imagem.jpg”
- ▶ **Prefira o caminho relativo** ao absoluto.

# HTML

## Imagens

- ▶ Referenciando com o **caminho relativo**:
- ▶ Crie um arquivo HTML, que contenha a tag `img`, na Área de Trabalho.
- ▶ Crie uma pasta chamada `img` na Área de Trabalho e insira a imagem do seu arquivo HTML.
- ▶ No atributo `src`, digite “`img/Imagem.jpg`”
- ▶ **Prefira o caminho relativo** ao absoluto. Por quê?
  - ▶ Se o projeto for movido ou copiado para outro local (ou enviado por exemplo, pelo sigaa para o professor corrigir o trabalho), as imagens ainda serão carregadas.



## Exercício

- ▶ **TAREFA!** Crie esta página (`exercicio1.html`):



### Instituto Federal Catarinense

---

#### Avisos

- [01/01/2017] [Estamos de férias!](#)
- [10/01/2017] [Segunda chamada de matrícula](#)
- [25/01/2017] [Comunicado](#)
- [06/02/2017] [Volta às aulas!](#)

- ▶ Os *links* devem direcionar o usuário para a página `exercicio2.html`.

## Exercício

- ▶ A página `exercicio2.html` deve ser assim:



**Instituto Federal Catarinense**

---

**Página em Construção!**

Você selecionou uma página que ainda está em construção. Pedimos desculpas pelo ocorrido.



# HTML

## Formatação

- ▶ A estilização das páginas será feita com CSS. Mas podemos aplicar algumas formatações com *tags* HTML:
  - ▶ `<strong>`: negrito com valor semântico
  - ▶ `<b>`: negrito sem valor semântico
  - ▶ `<em>`: itálico com valor semântico
  - ▶ `<i>`: itálico sem valor semântico
  - ▶ `<u>`: sublinhado
  - ▶ `<mark>`: destaque

# HTML

## Tabelas

- ▶ `<table>`: marca a tabela.
- ▶ `<tr>`: marca uma linha na tabela.
- ▶ `<th>`: marca uma célula de cabeçalho na tabela.
- ▶ `<td>`: marca uma célula na tabela.
  
- ▶ O que colocamos dentro de um `<td>`?
  
- ▶ **IMPORTANTE**: a tabela deve ser usada para marcar *dados tabulares*.

# HTML

## Tabelas

```
<!DOCTYPE html>
<html lang="pt-br">
<body>
  <table>
    <tr>
      <th>Time</th><th>Pontos</th>
    </tr>
    <tr>
      <td>Gremio</td><td>15</td>
    </tr>
    <tr>
      <td>Juventude</td><td>8</td>
    </tr>
    <tr>
      <td>Inter</td><td>0</td>
    </tr>
  </table>
</body></html>
```

# HTML

## Tabelas

- ▶ A tabela também pode ter uma legenda.
- ▶ *tag* `<caption>`.
- ▶ Após a *tag* `table`:  
`<caption>`Tabela 1: dados sobre XXX`</caption>`

# HTML

## Tabelas

- ▶ Facilitando a criação de tabelas:  
`http://www.tablesgenerator.com/html\_tables`
- ▶ Geradores de tabelas online: informa a estrutura e ele gera o código.
- ▶ Basta copiar o código para o seu arquivo HTML!

## Referências

- ▶ W3SCHOOLS. Introduction to XML. What is XML?  
Disponível em:  
[https://www.w3schools.com/xml/xml\\_what\\_is.asp](https://www.w3schools.com/xml/xml_what_is.asp).  
Acesso em: 21 out. 2020.



# Introdução a Linguagens de Marcação

## Desenvolvimento Web I

Ricardo de la Rocha Ladeira  
{ricardo.ladeira@ifc.edu.br}



**INSTITUTO FEDERAL**  
Catarinense  
Campus Blumenau