



**Fecomércio RS**



**Senac**

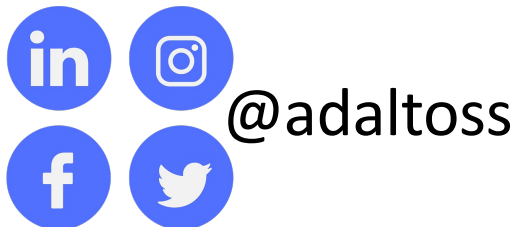
# Desenvolvimento de Serviços e APIs

## Aula07

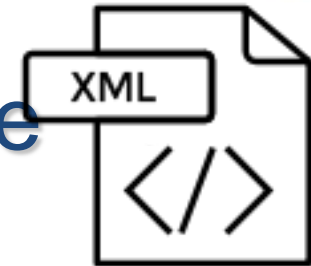
### XML

**Prof. MSc. Adalto Selau Sparremerger**

assparremerger@senacrs.com.br



# XML – eXtensible Markup Language



- Projetado para armazenar e transportar dados.
- Projetado para ser humano e legível por máquina.
- XML independe de software e hardware para armazenar e transportar dados.

```
<produto>  
  <id>1</id>  
  <nome>Coca-Cola</nome>  
  <preco>5.99</preco>  
</produto>
```

# O que é XML?

- É uma linguagem de marcação semelhante ao HTML.
- Foi projetado para armazenar e transportar dados.
- Foi projetado para ser autodescritivo.
- É uma recomendação do W3C.
- **XML não faz nada**, é apenas uma estrutura de texto para ser utilizado como armazenamento e transporte de dados.

# XML x HTML?

- XML e HTML foram projetados com diferentes objetivos:
  - XML foi projetado para transportar dados - com foco em quais dados são
  - HTML foi projetado para exibir dados - com foco na aparência dos dados
  - As tags XML **não são predefinidas**, como as tags HTML.

# XML não usa tags predefinidas

- As *tags* no exemplo abaixo (como **<produto>** e **<nome>**) não estão definidas em nenhum padrão XML.
- Essas *tags* são "inventadas" pelo autor do documento XML.
- HTML funciona com *tags* predefinidas como **<p>**, **<h1>**, **<table>** etc.
- Com o XML, o autor deve definir as *tags* e a estrutura do documento.

```
<produto>  
  <id>1</id>  
  <nome>Coca-Cola</nome>  
  <preco>5.99</preco>  
</produto>
```

# XML não usa tags predefinidas

- A maioria dos aplicativos XML funcionará conforme o esperado, mesmo que novos dados sejam adicionados (ou removidos).
- Imagine um aplicativo projetado para exibir a versão original do produtos.xml (**<produto>** **<id>** **<nome>** **<preco>**).
- Imagine uma versão mais recente do produtos.xml com a adição da *tag* **<quantidade>**.
- Da maneira como o XML é construído, a versão mais antiga do aplicativo ainda pode funcionar:

produtos.xml

```
<produto>
  <id>1</id>
  <nome>Coca-Cola</nome>
  <preco>5.99</preco>
</produto>
```

```
<produto>
  <id>1</id>
  <nome>Coca-Cola</nome>
  <preco>5.99</preco>
  <quantidade>100</quantidade>
</produto>
```

# XML - Sintaxe

- Os documentos XML devem conter um elemento raiz que é o pai de todos os outros elementos:

```
<root>  
  <child>  
    <subchild>.....</subchild>  
  </child>  
</root>
```

Fonte: <https://www.w3schools.com/xml/>



Fecomércio RS





# XML - Elemento

- Um elemento XML é tudo, desde (incluindo) a marca inicial do elemento até (incluindo) a marca final do elemento.
- Um elemento pode conter:
  - texto
  - atributos
  - outros elementos
  - ou uma mistura dos itens acima

# Importante:

- Tags XML diferenciam maiúsculas de minúsculas
- Nomes dos elementos devem começar com uma letra ou sublinhado
- Nomes dos elementos não podem começar com as letras xml (ou XML ou Xml, etc)
- Nomes dos elementos podem conter letras, dígitos, hífen, sublinhados e pontos
- Nomes dos elementos não podem conter espaços



# Implementação em AJAX consumindo XML

dados.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<lista_produtos>
  <produto>
    <id>1</id>
    <nome>Coca-Cola</nome>
    <preco>5.99</preco>
  </produto>
  <produto>
    <id>2</id>
    <nome>Pepsi</nome>
    <preco>4.49</preco>
  </produto>
</lista_produtos>
```

# Implementação em AJAX consumindo XML

servidor\_xml.php

```
<?php
    header('Content-Type: application/xml; charset=utf-8');
    $conn = mysqli_connect("localhost", "root", "", "api_rest");
    if( $conn ){
        $result = mysqli_query( $conn, "SELECT * FROM tbl_produtos");
        $xml = '<?xml version="1.0" encoding="UTF-8" ?>';
        $xml .= '<lista_produtos>';
        while( $produto = mysqli_fetch_array($result)){
            $xml .= '<produto> ';
            $xml .= '    <id>'.$produto["id"].'</id>';
            $xml .= '    <nome>'.$produto["nome"].'</nome>';
            $xml .= '    <preco>'.$produto["preco"].'</preco>';
            $xml .= '</produto> ';
        }
        $xml .= '</lista_produtos>';
        echo $xml;
    }
?>
```

# Implementação em AJAX consumindo XML

```
<script>
var http = new XMLHttpRequest();
http.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
        document.write("Dados: " + this.responseText + "<hr>");
        var dadosXml = this.responseXML;
        var produtos = dadosXml.getElementsByTagName("produto");
        document.write("<table border='1'>");
        document.write("    <tr>");
        document.write("        <th>Id</th>");
        document.write("        <th>Nome</th>");
        document.write("        <th>Preço</th>");
        document.write("    </tr>");
        for (i = 0; i < produtos.length; i++) {
            document.write("    <tr>");
            var id = produtos[i].getElementsByTagName("id");
            var nome = produtos[i].getElementsByTagName("nome");
            var preco = produtos[i].getElementsByTagName("preco");
            document.write("        <td>" + id[0].childNodes[0].nodeValue + "</td>");
            document.write("        <td>" + nome[0].childNodes[0].nodeValue + "</td>");
            document.write("        <td>" + preco[0].childNodes[0].nodeValue + "</td>");
            document.write("    </tr>");
        }
        document.write("    </table>");
    }
};

// ---->Para buscar os dados do banco de dados via PHP
http.open("GET", "servidor_xml.php", true);
// ---->Para buscar os dados em um arquivo XML
// http.open("GET", "dados.xml", true);
http.send();
</script>
```

index.html

# Repositório com os arquivos desenvolvidos em aula

[https://github.com/assparremberger/2020\\_1\\_Programacao\\_Internet\\_II/tree/master/pi2\\_aula08](https://github.com/assparremberger/2020_1_Programacao_Internet_II/tree/master/pi2_aula08)



**Fecomércio RS**



**Senac**