

Programação Web e suas Implicações

Prof. Carlos Bazilio

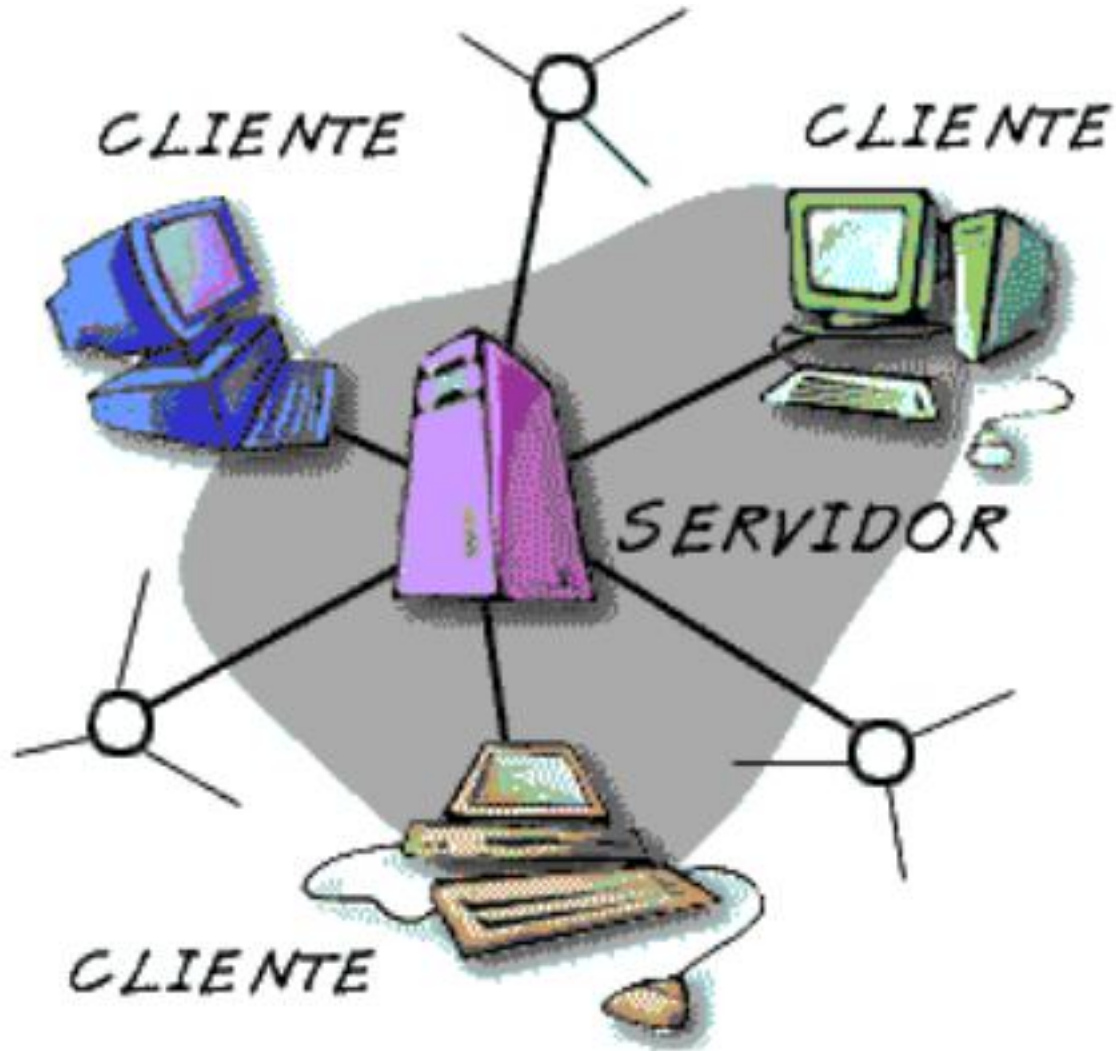
carlosbazilio@id.uff.br

<http://www.ic.uff.br/~bazilio>

Lição para Profissional de Computação

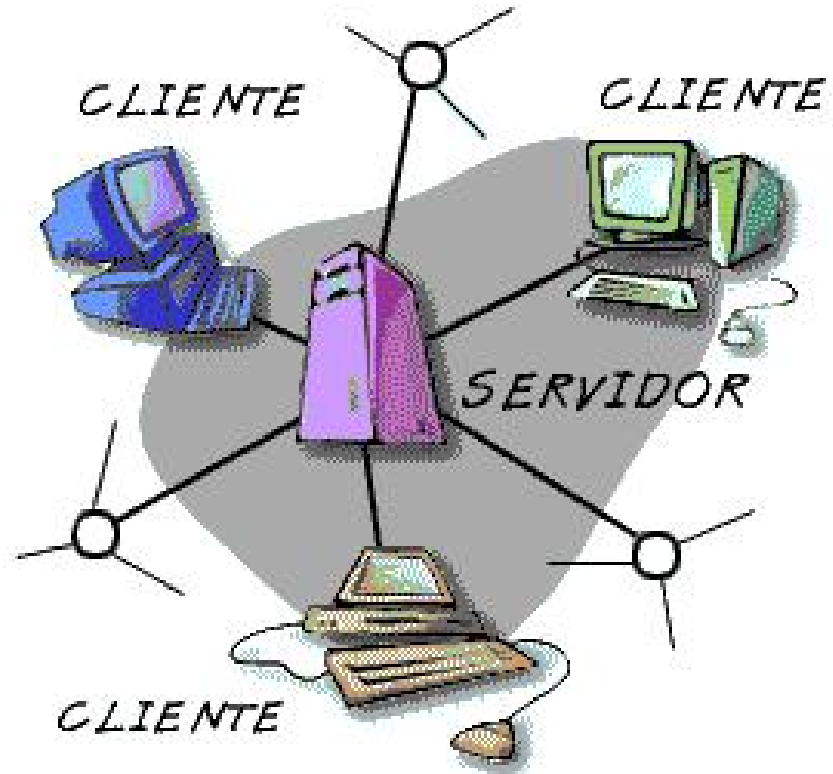
“Prostituam-se! Prostituam-se quando o assunto for linguagem de programação, IDE, metodologia, arquitetura, ou qualquer outro tipo de escolha técnica!”

Arquitectura



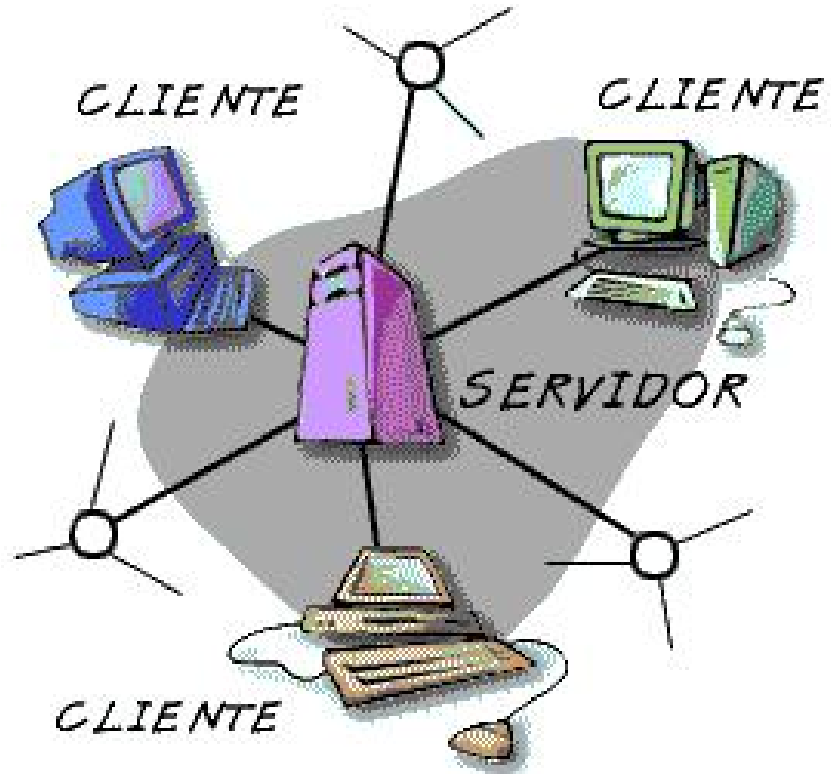
Execução no Cliente (Navegador, Browser)

- HTML
- CSS
- JavaScript
- XML



Execução no Cliente (Navegador, Browser)

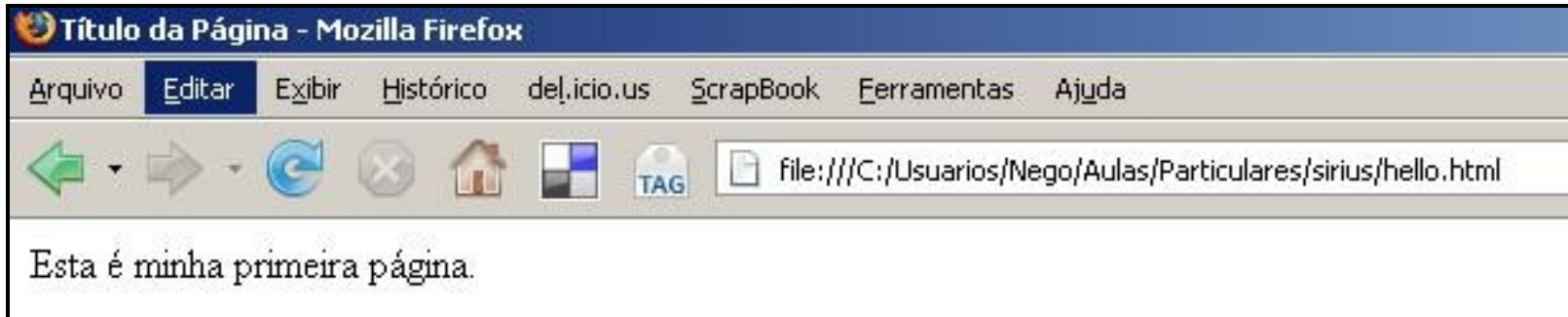
- HTML
- CSS
- JavaScript
- XML



HTML

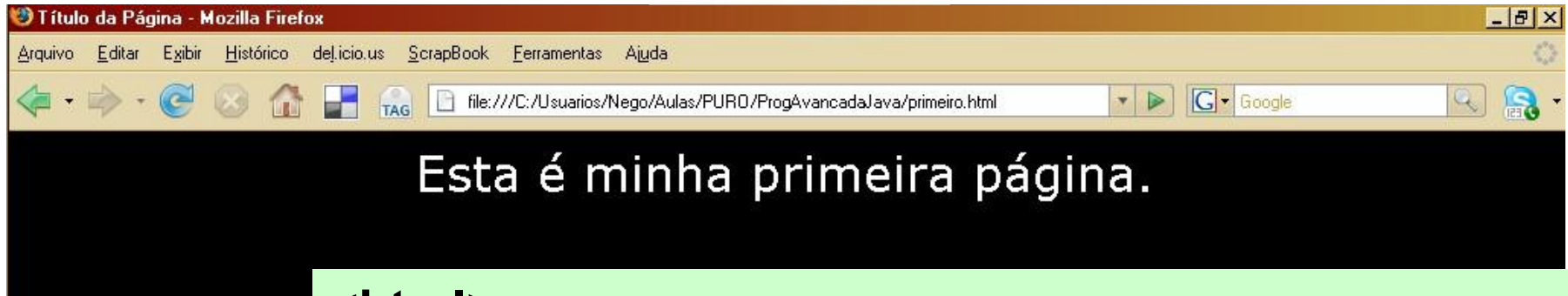
- *Hyper Text Markup Language*
- Um arquivo html contém marcadores (tags)
- Estes marcadores indicam para o navegador (browser) como a página deve ser apresentada
- Marcadores usualmente vem em pares:
`<tag></tag>`

HTML



```
<html>  
  <head>  
    <title>Título da Página</title>  
  </head>  
  <body>  
    Esta é minha primeira página.  
  </body>  
</html>
```

HTML



```
<html>  
  <head>  
    <title>Título da Página</title>  
  </head>  
  <body bgcolor="black">  
    <p align="center">  
      <font size="2" face="Verdana" color="white">  
        Esta é minha primeira página.  
      </font>  
    </p>  
  </body>  
</html>
```


HTML

Formulários

- HTML possibilita a criação de formulários online utilizando tags
- A tag `<form>` é a mais comum e permite a criação de um formulário de entrada de dados

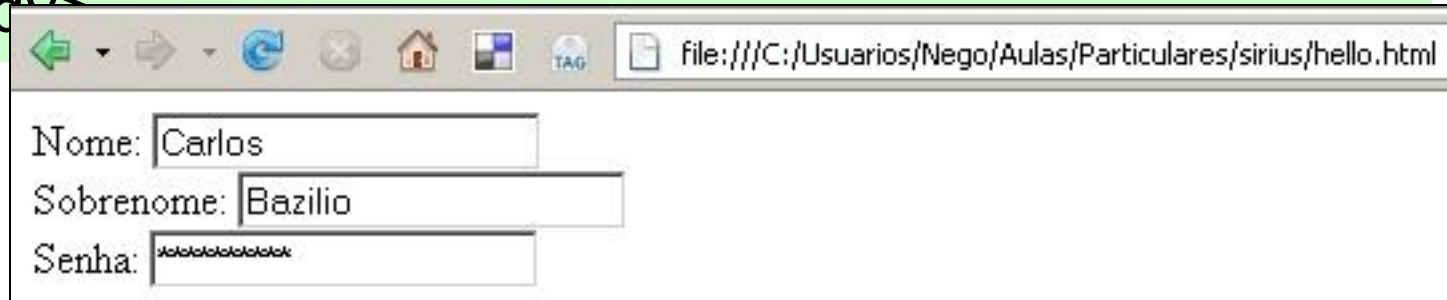
```
<body><form>
```

```
Nome: <input type="text" name="firstname"><br />
```

```
Sobrenome: <input type="text" name="lastname"><br />
```

```
Senha: <input type="password" name="senha">
```

```
</form></body>
```



HTML

Formulários

- Tipos que podem ser utilizados com a tag `<input>`:
 - `button`: Insere um botão
 - `checkbox`: Insere um checkbox; para vários itens, basta inserirmos vários “inputs” deste tipo
 - `file`: Insere botão seleção de arquivo através de uma caixa de diálogo
 - `hidden`: Campo pertencente ao formulário, mas que não será exibido na página
 - `image`: Insere uma imagem como um botão submit
 - `password`: Insere um campo password (caracteres digitados não aparecem)
 - `radio`: Insere um radiobox (análogo ao checkbox)
 - `reset`: Restaura os valores iniciais do formulário
 - `submit`: Encaminha os dados inseridos para algum arquivo
 - `text`: Insere um campo de edição de texto

HTML

Formulários

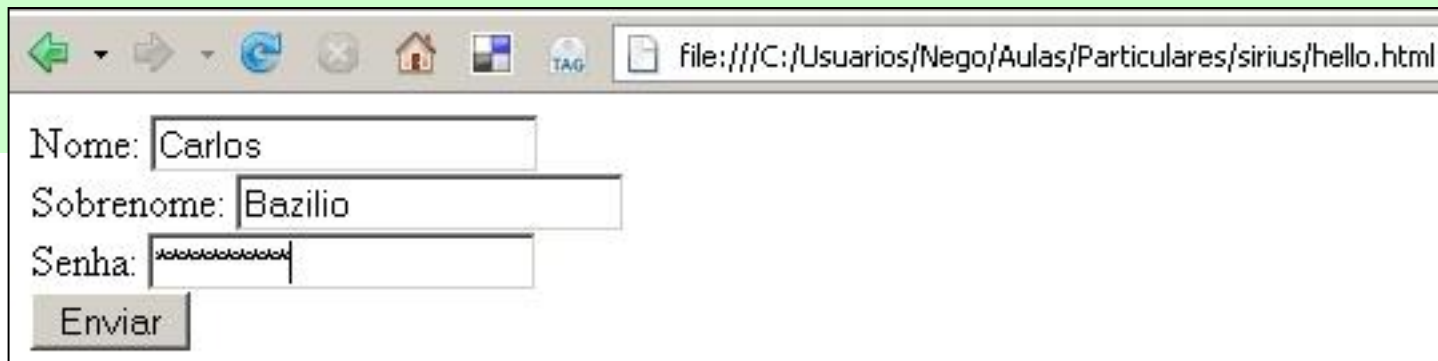
- Quando algum elemento do tipo “submit” é inserido num formulário e acionado, seus dados são enviados para um arquivo
- O arquivo mencionado é indicado pelo atributo “action” do elemento <form>
- Este arquivo deverá estar armazenado num servidor web (Apache, Tomcat, IIS, ...), e estará escrito em alguma linguagem de programação de servidores: jsp, php, asp, ...



HTML

Formulários

```
<body>  
  URL após clicar no  
  botão: /sirius/processaForm.jsp?firstname=Carlos&last  
  name=Bazilio&senha=abcdefg  
  Nome: <input type="text" name="firstname"><br />  
  Sobrenome: <input type="text" name="lastname"><br />  
  Senha: <input type="password" name="senha">  
  <input type="submit" value="Enviar">  
</form>  
</body>
```



A screenshot of a web browser window showing a form. The address bar displays the file path: file:///C:/Usuarios/Nego/Aulas/Particulares/sirius/hello.html. The form contains the following elements:

- Nome:
- Sobrenome:
- Senha:
- Enviar

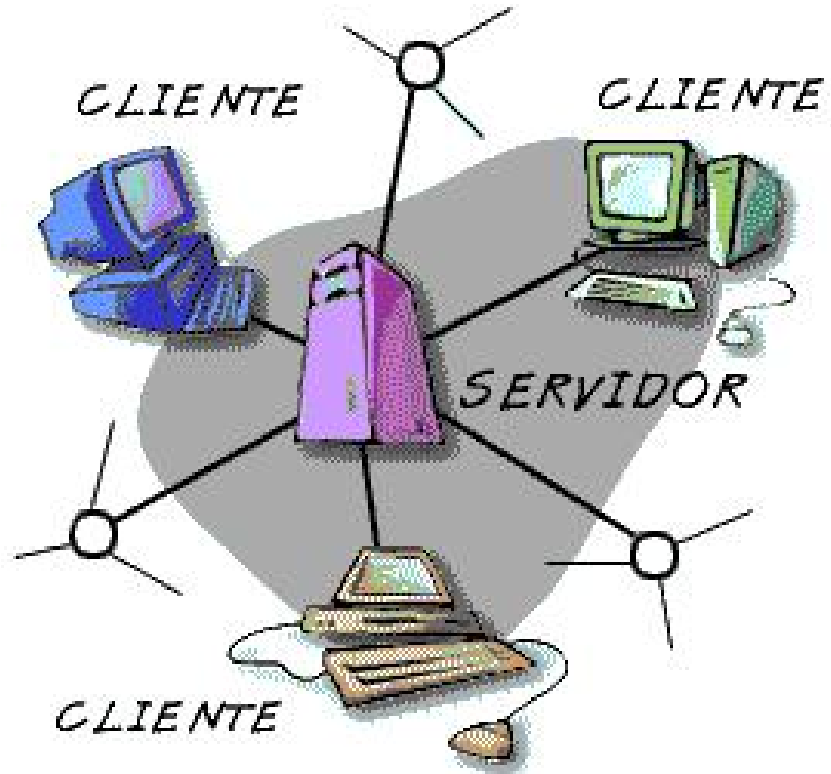
HTML

Referências

- <http://www.w3schools.com/html/default.asp>
– Lista de tags possíveis e exemplos diversos
- <http://www.w3.org/TR/html401/>
– Link do consórcio WWW que define padrões de linguagens web (em especial, HTML)

Execução no Cliente (Navegador, Browser)

- HTML
- CSS
- JavaScript
- XML



CSS

- *Cascading Style Sheets*
- Estilos definem como elementos html (textos, tabelas, links, ...) devem ser apresentados
- Permite a separação entre definição de conteúdo (informação, texto, ...) e formatação (cores, fontes, ...) em HTML
- Atualmente é o padrão para inserção de estilo na construção de páginas html

CSS

Como definir (1/3)

- Existem 3 maneiras de definir CSS para um documento HTML
 - Criando um

```
/* Conteúdo do arquivo meuestilo.css */  
body { background-color: black }  
p { font-family: "Verdana"; text-align: center; color: white; font-size: 200% }
```

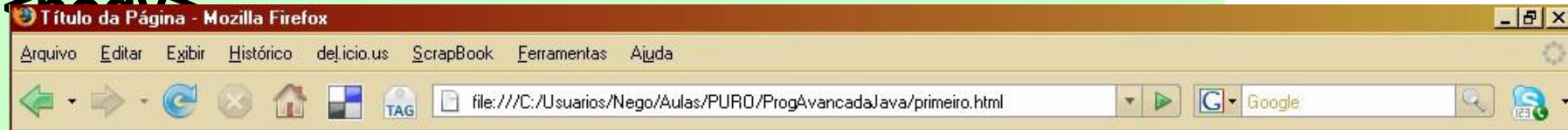
```
<html>
```

```
<head><link rel="stylesheet" type="text/css"  
href="meuestilo.css" />
```

```
<title>Título da Página</title>
```

```
</head>
```

```
<body>
```



Esta é minha primeira página.

```
</body>
```


CSS

Como definir (2/3)

- Internamente (dentro da própria página HTML a ser exibida)

```
<html>
  <head>
    <style type="text/css">
      body { background-color: black }
      p { font-family: "Verdana";
        text-align: center; color: white;
        font-size: 200% }
    </style>
  </head>
  <body>
    Esta é minha primeira página.
  </body>
</html>
```

CSS

Como definir (3/3)

- Definindo de forma inline

```
<html>  
  <head</head>  
  <body style="background-  
color: black; text-align: center">  
    Esta é minha primeira página.  
  </body>  
</html>
```

- Caso existam várias definições para um mesmo elemento, a prioridade é: inline, interna e externa

CSS

- Formato geral:
seletor { propriedade: valor }
- Uma definição de estilo em CSS será composta por uma sequência de definições como esta acima
- Exemplos:
 - body { color: black }
 - p { font-family: "Verdana"; text-align: center; color: white }
 - p {margin-left: 20px}

CSS - Classes

- Classes em CSS permitem que um mesmo elemento seja exibido de diferentes formas

- Formato usando classes:

```
seletor.classe { propriedade: valor }
```

- Exemplos:

- `p.direita {text-align: right}` // Alinha à direita
- `p.centro {text-align: center}` // Centraliza
- `.esquerda {text-align: left}` // Aplicado a qualquer elemento html que contenha classe *esquerda*

```
<p class="direita">Este parágrafo será alinhado à direita</p>
```

```
<p class="centro">Este parágrafo será centralizado</p>
```

```
<p class="esquerda">Este será à esquerda</p>
```

CSS – Pseudo-classes

- Pseudo-classes são usadas para adição de efeitos especiais a alguns seletores
- Formato:
seletor:pseudo-classe {propriedade: valor}
- Exemplos:
 - `a:link {color: #FF0000} /* link não visitado*/`
 - `a:visited {color: #00FF00} /* link visitado */`
 - `a:hover {color: #FF00FF} /* mouse sobre link */`
 - `a:active {color: #0000FF} /* link selecionado */`

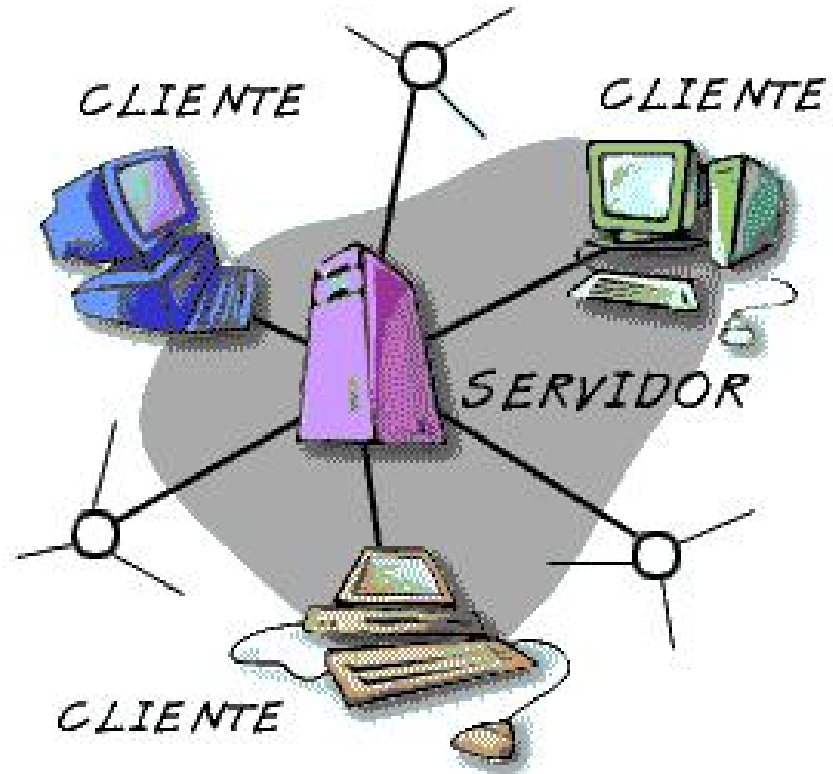
CSS

Referências

- Com CSS podemos formatar:
 - Background, Textos, Fontes, Margens, Bordas, Listas, Tabelas, ...
- <http://www.w3schools.com/css/default.asp>
 - Site com descrição de diversas construções disponíveis em CSS
- <http://www.csszengarden.com/>
 - Site com demonstração do poder expressivo de CSS

Execução no Cliente (Navegador, Browser)

- HTML
- CSS
- JavaScript
- XML



JavaScript

- É a linguagem de script utilizada por milhões de páginas web
- Foi projetada para aumentar interatividade das páginas web:
 - Validação de formulários, interação com o usuário (p.ex., tratamento de cliques de botões), detecção de navegadores, etc
- É reconhecida pela maioria dos navegadores
- Seu processamento é feito na máquina cliente (browser)
- Não há relação com Java

JavaScript

Janelas Popup

- Funções para criação de janelas popup:

- Alerta (`alert("Texto a ser exibido");`)

- Confirmação
em janela

- Entrada
de entrada

- Janelas
quando

```
<body>  
  <script type="text/javascript">  
    var nome = prompt("Seu nome");  
    if (nome != null && nome != "")  
      document.write("Oi " + nome + " !!");  
    else  
      document.write("Oi anônimo!");  
  </script>  
</body>
```

[exemplo-javascript.html](#)

JavaScript - Sintaxe

- Possui construções existentes na maioria das linguagens de programação (sintaxe similar a C):
 - Declaração de variáveis (*var x;*)
 - Comandos condicionais, repetições, definição de funções de usuário
 - Operadores de atribuição, comparação, ...
 - Tipos pré-definidos, classes de objetos, ...

JavaScript – Eventos

- Eventos são ações que podem ser detectadas por um script
- Exemplos de eventos:
 - Clique do mouse, abertura de uma página web ou imagem, envio de um formulário html, uma tecla pressionada, etc
- O tratamento destes eventos pode ser a chamada de funções do script

JavaScript – Eventos

```
<html>
  <head><script src="event.js"></script>
</head>
<body>
  
</body>
</html>
```

```
/* Conteúdo do arquivo event.js */
function mouseOver()
{
  document.imagem.src = "quadrado2.gif";
}
function mouseOut()
{
  document.imagem.src = "quadrado1.gif";
}
```

[exemplo2-javascript.html](#)

JavaScript – Objetos

- JavaScript é uma linguagem com recursos do paradigma OO
- Com isto, algumas classes utilitárias padrões estão disponíveis, as quais possuem métodos e propriedades:
 - String: manipulação de strings no script

```
/* Exemplo de função de script que calcula  
o tamanho de uma string */  
function tamanho(msg)  
{  
    return msg.length;  
}
```

- http://www.w3schools.com/jsref/jsref_obj_string.asp

JavaScript – Objetos

- Date: manipulação de datas no script

```
/* Exemplo de função de script que manipula  
datas */  
function bug(data)  
{  
    var x = new Date();  
    x.setFullYear(2000,0,0);  
    if (data > x)  
        alert("Bug do milênio!");  
}
```

- http://www.w3schools.com/jsref/jsref_obj_date.asp

JavaScript – Objetos

- Outras classes disponíveis na linguagem JavaScript:
 - Array: armazenamento de conjuntos de valores
 - http://www.w3schools.com/js/js_obj_array.asp
 - Boolean: manipulação de valores booleanos
 - http://www.w3schools.com/js/js_obj_boolean.asp
 - Math: manipulação de valores com operações matemáticas
 - http://www.w3schools.com/jsref/jsref_obj_math.asp
- Além destas, todos os objetos HTML DOM podem ser acessados através de scripts
 - https://www.w3schools.com/js/js_htmlDOM.asp

JavaScript – Objetos

Exemplo

```
/* Exemplo de função de script que manipula objetos  
de JavaScript */  
function exemplo_objetos()  
{  
    var d=new Date();  
    var dias=new Array(7);  
    dias[0]="Domingo"; dias[1]="Segunda";  
    dias[2]="Terça"; dias[3]="Quarta"; dias[4]="Quinta";  
    dias[5]="Sexta"; dias[6]="Sábado";  
    document.write("Hoje é" + dias[d.getDay()]);  
    document.write("Dia qualquer: " +  
    dias[Math.round(Math.random()*6)]);  
    document.write(navigator.appCodeName);  
}
```


JavaScript - Exemplo



```
<html>  
  <head>  
    <title>Título da Página</title>  
  </head>  
  <body>  
    <script type="text/javascript">  
      document.write("Esta é minha  
primeira página.");  
    </script>  
  </body>  
</html>
```

JavaScript – Onde ocorrem

- Uma tag `<script/>` pode ser definida numa seção head, numa seção body e também pode ser definida externamente:
 - Na seção head, os scripts são executados quando são chamados ou quando algum evento ocorre;
 - Na seção body, os scripts são executados na carga da página web
 - Para definição externa, um arquivo “.js” precisa ser fornecido com as funções necessárias

JavaScript – Exemplo

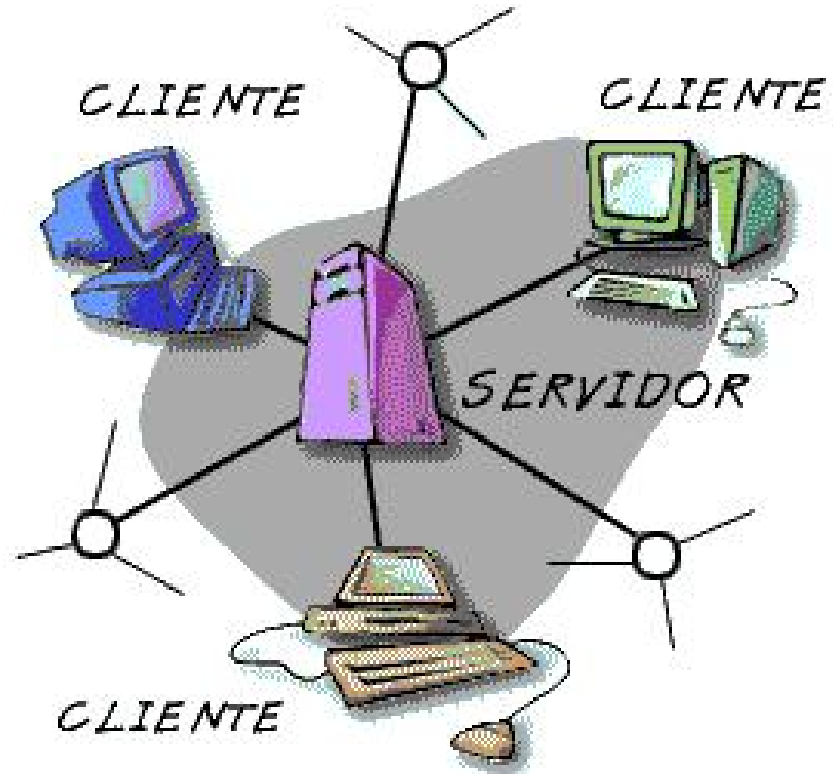
```
<html>
  <head>
    <title>Título da Página</title>
    <script src="hello.js"></script>
  </head>
  <body onload="message()">
    <script type="text/javascript">
      document.write("<h1>Esta é minha primeira
página.</h1>");
    </script>
  </body>
</html>
```

JavaScript

- Boas Práticas de Performance
 - Incluir ao final da página
- Mais Usos
 - Criação de cookies
 - Validação de entrada
 - Disparo de funções com retardo (tempo)
 - Criação de objetos próprios
- Referências
 - <http://www.w3schools.com/js/default.asp>
 - <http://oopweb.com/JavaScript/Documents/jsintro/Volume/tutorial.htm>

Execução no Cliente (Navegador, Browser)

- HTML
- CSS
- JavaScript
- XML



XML

```
<?xml version="1.0" encoding="UTF-8"?>
<livros>
  <livro isbn="0001">
    <titulo>JavaServer Pages</titulo>
    <autor>Nick Todd</autor>
    <editora>Campus</editora>
    <assunto>JSP</assunto>
  </livro>
  <livro isbn="0002">
    <titulo>Meu pé de laranja lima</titulo>
    <editora>Vozes</editora>
    <autor>Brilhante</autor>
  </livro>
</livros>
```

XML

- Um arquivo XML é definido por:
 - Um arquivo em formato ASCII
 - Tags aninhadas hierarquicamente (número indeterminado de sub-elementos)
 - Um único elemento (<tag></tag>) raiz
 - Indeterminado número de atributos (<tag atr1="val1" ...></tag>) numa tag

Justificativa para Uso de XML

- Integração de aplicações
- Formato flexível (linguagem de propósito geral)
- Tem sido amplamente utilizada
 - [Meus dados na declaração do IRPFI](#)
 - [OpenOffice.org XML Essentials](#)
 - Canais de notícia (RSS, Feeds):
 - [Artigos de Desenvolvimento Web na IBM](#)
 - [Informações sobre tecnologia do site do Globo](#)



Exemplo de Integração usando XML

IRPF 2008 - CARLOS BAZILIO MARTINS - Declaração com Desconto Simplificado

Declaração Preenchimento Fichas Ferramentas Ajuda

Declaração

- Identificação do Contribuinte
- Rend. Trib. Receb. de PJ
- Rend. Trib Receb. de PF/Ext
- Rend. Isentos e Não-Tributá
- Rend. Sujeitos à Trib. Exclus
- Imposto Pago
- Dependentes
- Alimentandos
- Pagamentos e Doações Efet
- Bens e Direitos
- Dívidas e Ônus Reais
- Informações do Cônjuge
- Espólio
- Doações a Part. Políticos, C
- Atividade Rural
- Ganhos de Capital
- Moeda Estrangeira
- Renda Variável
- Resumo da declaração

Identificação do Contribuinte

Nome: CARLOS BAZILIO MARTINS

Data de Nascimento: 07/07/1975 Título Eleitoral: _____

Endereço

No Brasil No Exterior

Houve mudança de endereço? Sim Não

Tipo: Estrada Logradouro: RIO PETROPOLIS, KM 83

Número: 38-A Complemento: CX. POSTAL 17E Bairro/Distrito: DUGUES

UF: RJ - Rio de Jan Município: Petrópolis

CEP: 25650-001 Consulta CEP DDD 0xx: 24 Telefone: 2235-9433

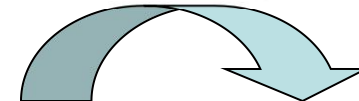
Ocupação Principal

Natureza da ocupação: 22 Servidor público de autarquia ou fundação federal

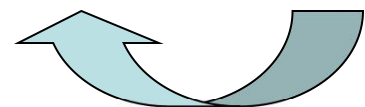
Ocupação Principal: 294 Professor do ensino superior

Esta declaração é retificadora? Sim Não

XML
(CEP)



XML (Dados
Postais)



WebService = XML sobre HTTP

Esquema XML

- Documentos XSD (esquemas), assim como DTDs, definem uma gramática para documentos XML
- Ou seja, definem o que vale e o que não vale num documento XML
- Para o exemplo dos Correios, a definição pode ser feita em comum acordo (cliente e servidor) ou determinadas pelo servidor

XML

```
<?xml version="1.0" encoding="UTF-8"?>
<livros>
  <livro isbn="0001">
    <titulo>JavaServer Pages</titulo>
    <autor>Nick Todd</autor>
    <editora>Campus</editora>
    <assunto>JSP</assunto>
  </livro>
  <livro isbn="0002">
    <titulo>Meu pé de laranja lima</titulo>
    <editora>Vozes</editora>
    <autor>Brilhante</autor>
  </livro>
</livros>
```

DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!--DTD generated by XMLSpy v2008
(http://www.altova.com)-->
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT livros ((livro+))>
<!ATTLIST livros
    xmlns:xsi CDATA #FIXED
    "http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation CDATA #IMPLIED>
<!ELEMENT livro ((titulo, ((editora, autor) | (autor, editora,
assunto))))>
<!ELEMENT editora (#PCDATA)>
<!ELEMENT autor (#PCDATA)>
<!ELEMENT assunto (#PCDATA)>
```

Esquema XML (XSD)

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="livros">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="livro" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="livro">
    <xs:complexType>
      <xs:all>
        <xs:element name="titulo" type="xs:string"/>
        <xs:element name="autor" type="xs:string"/>
        <xs:element name="editora" type="xs:string" minOccurs="0"/>
        <xs:element name="assunto" type="xs:string" minOccurs="0"/>
      </xs:all>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

XML

Vantagens e Desvantagens

- Vantagens:
 - Documento texto
 - Formato flexível
 - Reuso de ferramentas de manipulação
- Desvantagens:
 - Documento verboso (em torno de 70% são tags)
 - Manipulação manual ou através de bibliotecas

XML

Referências

- <http://www.w3.org/XML/>
 - Site do consórcio W3C sobre a linguagem
- <http://www.w3schools.com/xml/default.asp>
 - Site com informações básicas e muitos exemplos
- <http://www.xml.com/>
 - Site notícias para desenvolvimento avançado utilizando XML
- <http://xmlsucks.org/>
 - Site que fala sobre desvantagens do formato XML

Tableless

- “Antigamente” (antes da definição de CSS), os sites eram estruturados utilizando tabelas
- Conteúdos web eram disponibilizados de forma rígida e em posições fixas
- Esta prática vai contra a tendência atual de computação ubíqua (disponibilização de informações em diversos meios e momentos)
- Tableless (*less tables*) é o movimento que tenta desestimular esta prática (uso excessivo de tabelas)
- Tabelas devem ser utilizadas **apenas** para definição de tabelas (de dados, números, etc). Cada tag HTML tem uma semântica própria e deve ser usada para tal.

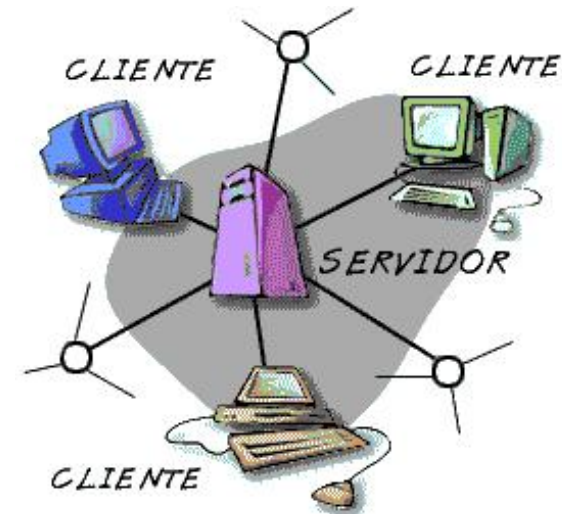
Padrões Web

Web Standards

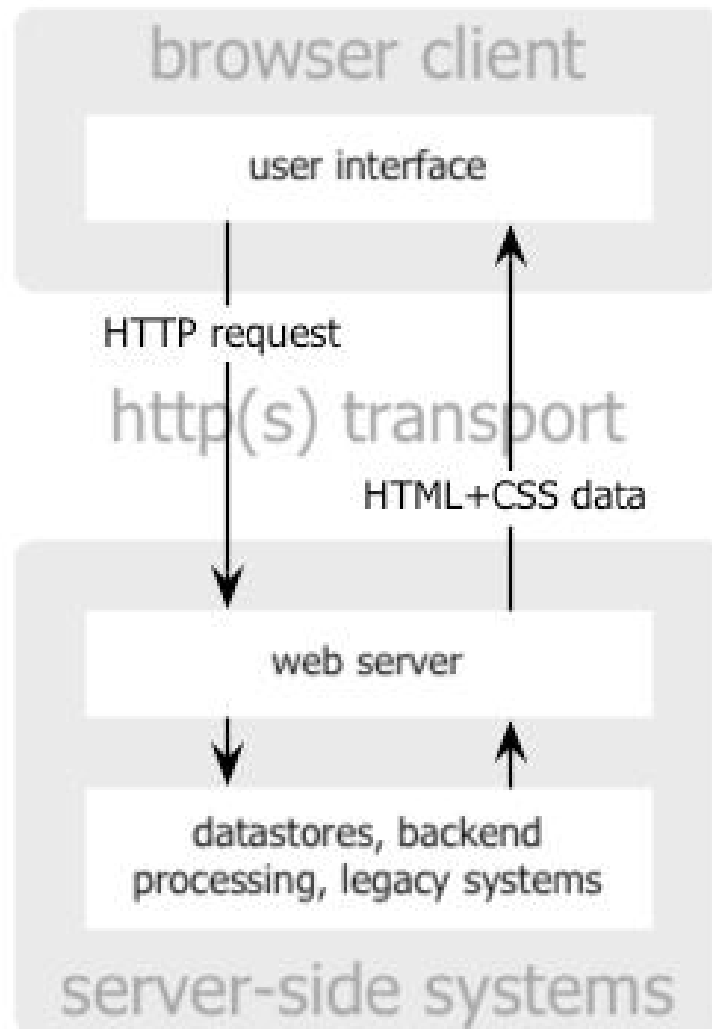
- Podem ser vistos como boas práticas na construção de páginas web
- Estes padrões procuram garantir interoperabilidade na web
- Como exemplo, podem ser abordadas questões como usabilidade e acessibilidade (permitir acesso a portadores de deficiência)
- Ou seja, é um termo mais geral e abrangente que Tableless

Execução no Servidor (Servidor Web)

- Servidores Web
 - Hospedam recursos a serem solicitados por navegadores
 - Exemplos: Apache, IIS, “Tomcat”, ...
- Linguagens de Script
 - Combinam comandos linguagem de programação com tags HTML para produzir tags HTML mais dinâmicas
 - Exemplos: ASP, JSP, PHP, ...



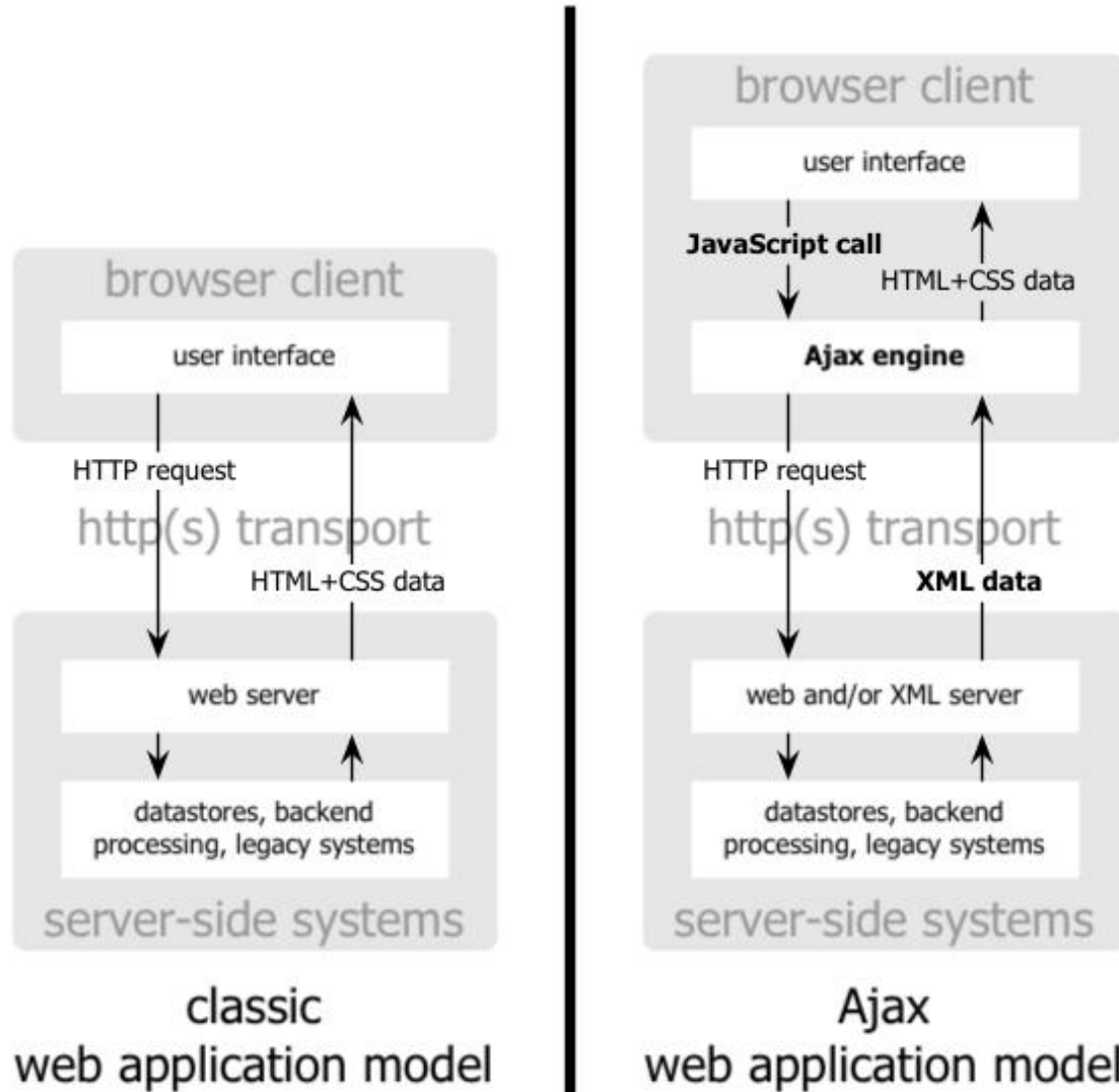
Modelo Web Clásico



AJAX

- *Asynchronous Javascript And XML*
- Construção de páginas mais dinâmicas através de chamadas assíncronas ao servidor
- Não é uma nova tecnologia ou linguagem, mas sim uma combinação de tecnologias padrões já existentes:
 - XHTML + CSS + XML + JavaScript + XSLT

AJAX



Web 2.0

- Termo que define uma tendência de projeto e websites
- Estes valorizam a criatividade, o compartilhamento de informações (APIs, RSS, ...), a colaboração entre usuários, a inferência de novos conteúdos, folksonomia (tags), etc
- **Não** está associado a nenhuma nova especificação da Web, apesar de ser beneficiar



WIKIPEDIA



Web 2.0

- Algumas tecnologias, arquiteturas e linguagens “novas”, das quais a suposta web 2.0 tira proveito:
 - AJAX: XHTML + CSS + XML + JavaScript + XSLT
 - Web Services

Tableless, AJAX, Web 2.0

Referências

- <http://del.icio.us/carlosbazilio/>
 - Meus favoritos sobre o assunto
- <http://www.hideout.com.br/usp/semacomp2005/slides/minicurso.html>
 - Minicurso apresentado na USP em 2005 sobre web 2.0
- <http://en.wikipedia.org/wiki/AJAX> e <http://www.w3schools.com/Ajax/Default.aspx>
 - Tutoriais sobre AJAX

Considerações Finais

- Em se tratando de web, muito ainda está por vir:
 - Melhor integração com dispositivos móveis, veículos, eletrodomésticos, ...
 - Mudança nas relações trabalho/local de trabalho, estudo/local de estudo, ...
 - Autêntico BBB
 - Extinção do dinheiro em papel
 - E muito trabalho para nós:
 - Novas linguagens, novos ambientes de execução, o papel crucial da engenharia de software, novos clientes, questões filosóficas, morais, ...

That's all Folks!

