

INSTITUTO FEDERAL SUL-RIO-GRANDENSE

UNIVERSIDADE ABERTA DO BRASIL

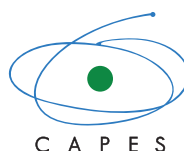
**Programa de Fomento ao Uso das
TECNOLOGIAS DE COMUNICAÇÃO E INFORMAÇÃO NOS CURSOS DE GRADUAÇÃO - TICS**

TICS

LINGUAGEM DE PROGRAMAÇÃO WEB

Anubis Graciela de Moraes Rossetto

Ministério da
Educação



Copyright© 2011 Universidade Aberta do Brasil
Instituto Federal Sul-rio-grandense

Apostila de Linguagem de Programação Web

ROSSETTO, Anubis Graciela de Moraes

2012/1

Produzido pela Equipe de Produção de Material Didático da
Universidade Aberta do Brasil do Instituto Federal Sul-rio-grandense

TODOS OS DIREITOS RESERVADOS

PRESIDÊNCIA DA REPÚBLICA

Dilma Rousseff

PRESIDENTE DA REPÚBLICA FEDERATIVA DO BRASIL

MINISTÉRIO DA EDUCAÇÃO

Fernando Haddad

MINISTRO DO ESTADO DA EDUCAÇÃO

Luiz Cláudio Costa

SECRETÁRIO DE EDUCAÇÃO SUPERIOR - SESU

Eliezer Moreira Pacheco

SECRETÁRIO DA EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

Luís Fernando Massonetto

SECRETÁRIO DA EDUCAÇÃO A DISTÂNCIA – SEED

Jorge Almeida Guimarães

PRESIDENTE DA COORDENAÇÃO DE APERFEIÇOAMENTO DE PESSOAL DE
NÍVEL SUPERIOR - CAPES

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SUL-RIO-GRANDENSE [IFSUL]

Antônio Carlos Barum Brod

REITOR

Daniel Espírito Santo Garcia

PRÓ-REITOR DE ADMINISTRAÇÃO E DE PLANEJAMENTO

Janete Otte

PRÓ-REITORA DE DESENVOLVIMENTO INSTITUCIONAL

Odeli Zanchet

PRÓ-REITOR DE ENSINO

Lúcio Almeida Hecktheuer

PRÓ-REITOR DE PESQUISA, INOVAÇÃO E PÓS-GRADUAÇÃO

Renato Louzada Meireles

PRÓ-REITOR DE EXTENSÃO

IF SUL-RIO-GRANDENSE CAMPUS PELOTAS

José Carlos Pereira Nogueira

DIRETOR-GERAL DO CAMPUS PELOTAS

Clóris Maria Freire Dorow

DIRETORA DE ENSINO

João Róger de Souza Sastre

DIRETOR DE ADMINISTRAÇÃO E PLANEJAMENTO

Rafael Blank Leitzke

DIRETOR DE PESQUISA E EXTENSÃO

Roger Luiz Albernaz de Araújo

CHEFE DO DEPARTAMENTO DE ENSINO SUPERIOR

IF SUL-RIO-GRANDENSE

DEPARTAMENTO DE EDUCAÇÃO A DISTÂNCIA

Luis Otoni Meireles Ribeiro

CHEFE DO DEPARTAMENTO DE EDUCAÇÃO A DISTÂNCIA

Beatriz Helena Zanotta Nunes

COORDENADORA DA UNIVERSIDADE ABERTA DO BRASIL – UAB/IFSUL

Marla Cristina da Silva Sopeña

COORDENADORA ADJUNTA DA UNIVERSIDADE ABERTA DO BRASIL – UAB/
IFSUL

Cinara Ourique do Nascimento

COORDENADORA DA ESCOLA TÉCNICA ABERTA DO BRASIL – E-TEC/IFSUL

Ricardo Lemos Sainz

COORDENADOR ADJUNTO DA ESCOLA TÉCNICA ABERTA DO BRASIL – E-TEC/
IFSUL

IF SUL-RIO-GRANDENSE

UNIVERSIDADE ABERTA DO BRASIL

Beatriz Helena Zanotta Nunes

COORDENADORA DA UNIVERSIDADE ABERTA DO BRASIL – UAB/IFSUL

Marla Cristina da Silva Sopeña

COORDENADORA ADJUNTA DA UNIVERSIDADE ABERTA DO BRASIL – UAB/
IFSUL

Mauro Hallal dos Anjos

GESTOR DE PRODUÇÃO DE MATERIAL DIDÁTICO

PROGRAMA DE FOMENTO AO USO DAS TECNOLOGIAS DE COMUNICAÇÃO E INFORMAÇÃO NOS CURSOS DE GRADUAÇÃO –TICS

Raquel Paiva Godinho

GESTORA DO EDITAL DE TECNOLOGIAS DE INFORMAÇÃO E COMUNICAÇÃO –
TICS/IFSUL

Ana M. Lucena Cardoso

DESIGNER INSTRUCIONAL DO EDITAL TICS

Lúcia Helena Gadret Rizzolo

REVISORA DO EDITAL TICS

EQUIPE DE PRODUÇÃO DE MATERIAL DIDÁTICO – UAB/IFSUL

Lisiane Corrêa Gomes Silveira

GESTORA DA EQUIPE DE DESIGN

Denise Zarnottz Knabach

Felipe Rommel

Helena Guimarães de Faria

Lucas Quaresma Lopes

Tabata Afonso da Costa

EQUIPE DE DESIGN

Catiúcia Klug Schneider

GESTORA DE PRODUÇÃO DE VÍDEO

Gladimir Pinto da Silva

PRODUTOR DE ÁUDIO E VÍDEO

Marcus Freitas Neves

EDITOR DE VÍDEO

João Eliézer Ribeiro Schaun

GESTOR DO AMBIENTE VIRTUAL DE APRENDIZAGEM

Giovani Portelinha Maia

GESTOR DE MANUTENÇÃO E SISTEMA DA INFORMAÇÃO

Carlo Camani Schneider

Efrain Becker Bartz

Jeferson de Oliveira Oliveira

Mishell Ferreira Weber

EQUIPE DE PROGRAMAÇÃO PARA WEB

SUMÁRIO

GUIA DIDÁTICO	9
UNIDADE A - INTRODUÇÃO AO DESENVOLVIMENTO PARA WEB	13
A internet e a World Wide Web	15
Linguagens de Programação para Web client-side e server-side	19
Tendências Web	21
Atividades	23
Introdução ao HTML	24
Atividades	33
UNIDADE B - LINGUAGEM DO LADO CLIENTE	37
Introdução a linguagem JavaScript	39
Objetos Javascript	47
Eventos Javascript	51
Funções	52
Atividades - parte 1	53
Arrays	59
Strings	56
Data e Hora	57
Atividades - parte 2	58
Objeto Window	63
Objeto Document	65
Solução passo-a-passo	66
Atividades - parte 3	69
UNIDADE C - INTRODUÇÃO A LINGUAGEM PHP	75
Linguagem PHP	77
Características da linguagem	77
Ferramentas necessárias	77
Estrutura da linguagem	79
Atividades - parte 1	89
Arrays	90
Strings	93
Datas	94
Atividades - parte 2	98
Funções	101
Atividades - parte 3	106
Sessões	107
Cookies	109
Atividades - parte 4	112
UNIDADE D - LINGUAGEM PHP COM ACESSO A BANCO DE DADOS	113
PHP com Banco de Dados	115
Atividades - parte 1	122
Área administrativa	123
Atividades - parte 2	134
Autenticação de usuário para área administrativa	135
Atividades - parte 3	140

APRESENTAÇÃO

Prezado(a) aluno (a),

Bem-vindo (a) ao espaço de estudo da Disciplina de Linguagem de Programação de Programação para Web.

O desenvolvimento de aplicações para web evidenciou uma significativa expansão nos últimos anos, possibilitando que os seus usuários tenham acesso a aplicações a partir do navegador de uma máquina com acesso à internet. A crescente aplicação nesta área é originada, sobretudo, dos avanços obtidos nas tecnologias de rede e da redução dos custos inerentes às tecnologias. Dada a dinâmica intrínseca da área e a importância que tem alcançado nos mais diferentes segmentos, a demanda por aplicações para esse ambiente tem crescido sobremaneira, ao mesmo tempo em que crescem as necessidades de resolução de problemas cada vez mais complexos.

O desenvolvimento de aplicações para a ambiente Web envolve a utilização de diferentes tecnologias, a fim de se alcançar um resultado de qualidade com recursos atuais e importantes para o usuário.

Nesta disciplina, serão desenvolvidos conceitos com utilização de linguagens de programação adequadas à criação de aplicações voltadas para Web. Para tanto, serão utilizadas linguagens de programação client-side e server-side com acesso a banco de dados.

Nas unidades, serão abordados os seguintes conteúdos: Linguagem HTML: confecção e formatação de páginas, formulários com métodos GET e POST; recursos da linguagem JavaScript: estruturas de controle, funções e eventos, validação de formulários, manipulação de janelas; recursos da linguagem PHP: estruturas de controle, manipulação de arrays, strings e datas, funções, sessões e cookies; desenvolvimento com banco de dados: conexão, consultas, inserção, alteração, exclusão e controle de acesso de usuário.

Esperamos que, através dos conteúdos e das atividades propostas, você possa estabelecer subsídios para a compreensão dos recursos de programação para a Web. E, para tal, você pode contar com toda a equipe.

Bom trabalho!

Objetivo Geral

Ao final desta disciplina, o aluno será capaz de atuar no desenvolvimento de aplicações para a Web, empregando tecnologias emergentes, visando suprir as necessidades do mundo do trabalho.

Habilidades

- Reconhecer as tecnologias e ferramentas disponíveis para o desenvolvimento para web.
- Utilizar linguagens client-side e server-side no desenvolvimento de aplicações para web.
- Compreender, utilizar e controlar formulários em aplicações Web.
- Explorar os recursos de uma linguagem de script para validação de formulários, manipulação de janelas, uso de funções e eventos.

- Utilizar os recursos da linguagem PHP, como estruturas condicionais e de repetição, manipulação de arrays, strings e datas.
- Organizar o script utilizando a modularização e funções permitindo a reutilização do código.
- Explorar os recursos do uso de sessões e cookies com PHP.
- Criar aplicações com o uso de uma linguagem do lado servidor (PHP) com acesso a banco de dados.

Metodologia

A disciplina será desenvolvida em 100 horas através do Ambiente Virtual de Aprendizado Moodle, onde serão disponibilizados materiais a serem estudados para subsidiar a aprendizagem. O Moodle será o canal de comunicação direto entre discentes e tutores, com as seguintes possibilidades de interação:

- Disponibilizar aos discentes as tarefas a serem realizadas.
- Publicar os materiais de apoio e de leitura complementar.
- Acompanhar o desempenho dos discentes em relação às atividades propostas.
- Interagir com a turma através de fórum de discussão, salas de chat e correio eletrônico.
- Acessar e avaliar as tarefas realizadas pelos discentes.
- Estimular o trabalho cooperativo entre os discentes.
- Promover o estudo autônomo.
- Acompanhar a frequência de acesso ao ambiente pelos discentes.
- Acessar links interessantes e relacionados ao curso.

Avaliação

A avaliação do desenvolvimento e envolvimento do discente em todas as unidades curriculares considerará os seguintes elementos:

- A participação nas aulas à distância, através das ferramentas de comunicação do ambiente virtual.
- A realização e a entrega das atividades solicitadas, observando a relevância e pertinência aos conteúdos abordados e solicitados no trabalho.
- A avaliação final e presencial.

Programação

Primeira Semana

As atividades a serem desenvolvidas na primeira semana são:

1. Fórum: Apresentação do professor, da disciplina e questões gerais.
2. Leitura e estudo do conteúdo: Introdução ao Desenvolvimento para Web.
3. Participação do Fórum de discussão proposto pelo professor formador.

Segunda Semana

As atividades a serem desenvolvidas na segunda semana são:

1. Leitura e estudo do conteúdo: Introdução ao HTML.
2. Realização da atividade: Unidade A – Introdução ao HTML (exercícios).
3. Participação em Chat em horário marcado pelo professor formador para discutir questões relativas aos exercícios propostos.

Terceira Semana

As atividades a serem desenvolvidas na terceira semana são:

1. Leitura e estudo do conteúdo: Introdução à linguagem JavaScript – Parte 1.
2. Realização da atividade: Unidade B – Introdução à linguagem JavaScript (exercícios parte 1).
3. Participação em aula remotamente ministrada via Webconferência.

Quarta Semana

As atividades a serem desenvolvidas na quarta semana são:

1. Leitura e estudo do conteúdo: Introdução à linguagem JavaScript - Parte 2.
2. Realização da atividade: Unidade B – Introdução à linguagem JavaScript (exercícios parte 2).
3. Participação do Fórum de discussão proposto pelo professor formador.

Quinta Semana

As atividades a serem desenvolvidas na quinta semana são:

1. Leitura e estudo do conteúdo: Introdução à linguagem JavaScript - Parte 3.
2. Realização da atividade: Unidade B – Introdução à linguagem JavaScript (exercícios parte 3).
3. Participação em Chat em horário marcado pelo professor formador para discutir questões relativas aos exercícios propostos.

Sexta Semana

As atividades a serem desenvolvidas na sexta semana são:

1. Leitura e estudo do conteúdo: Introdução à linguagem PHP – parte 1.
2. Assistir a vídeo: Instalação e configuração de ferramentas.
3. Realização da atividade: Unidade C – Introdução à linguagem PHP (exercícios parte 1).
4. Participação do Fórum de discussão proposto pelo professor formador.

Sétima Semana

As atividades a serem desenvolvidas na sétima semana são:

1. Leitura e estudo do conteúdo: Introdução à linguagem PHP – parte 2.
2. Realização da atividade: Unidade C – Introdução à linguagem PHP (exercícios parte 2).
3. Participação do Fórum de discussão proposto pelo professor formador.

Oitava Semana

As atividades a serem desenvolvidas na oitava semana são:

1. Leitura e estudo do conteúdo: Introdução à linguagem PHP – parte 3.
2. Realização da atividade: Unidade C – Introdução à linguagem PHP (exercícios parte 3).
3. Participação em Chat em horário marcado pelo professor formador, para discutir questões relativas aos exercícios propostos.

Nona Semana

As atividades a serem desenvolvidas na nona semana são:

1. Leitura e estudo do conteúdo: Introdução à linguagem PHP – parte 4.
2. Realização da atividade: Unidade C – Introdução à linguagem PHP (exercícios parte 4).
3. Participação do Fórum de discussão proposto pelo professor formador.

Décima Semana

As atividades a serem desenvolvidas na décima semana são:

1. Leitura e estudo do conteúdo: Linguagem PHP com acesso a banco de dados – Parte 1.
2. Assistir vídeo: Instalação e configuração de ferramentas .
3. Realização da atividade: Unidade D – Linguagem PHP com acesso a banco de dados – (exercícios parte 1).
4. Participação em Chat em horário marcado pelo professor formador, para discutir questões relativas aos exercícios propostos.

Décima Primeira Semana

As atividades a serem desenvolvidas na décima primeira semana são:

1. Leitura e estudo do conteúdo: Linguagem PHP com acesso a banco de dados – Parte 2.
2. Realização da atividade: Unidade D – Linguagem PHP com acesso a banco de dados – (exercícios parte 2).
3. Participação do Fórum de discussão proposto pelo professor formador.

Décima Segunda Semana

As atividades a serem desenvolvidas na décima segunda semana são:

1. Leitura e estudo do conteúdo: Linguagem PHP com acesso a banco de dados – Parte 3.
2. Realização da atividade: Unidade D – Linguagem PHP com acesso a banco de dados – (exercícios parte 3).
3. Participação em Chat em horário marcado pelo professor formador, para discutir questões relativas aos exercícios propostos.

Referências:

NIEDERAUER, J. *Desenvolvendo websites com PHP*: Aprenda a criar websites dinâmicos e interativos. São Paulo: Novatec, 2004.

NIEDERAUER, Juliano. *PHP para quem conhece PHP*. São Paulo: Novatec, 2006.

SOARES, Wallace. *PHP 5*: Conceitos, Programação e Integração com Banco de Dados. 4 ed. São Paulo: Érica, 2007.

PHP. *Manual do PHP*. Disponível na Web em http://php.net/manual/pt_BR/index.php

Professor-Autor

Anubis Graciela de Moraes Rossetto

Graduada em Ciência da Computação pela Universidade de Passo Fundo em 1998; Especialização em Sistemas de Informação pela Universidade de Passo Fundo (2002). Mestre em Ciência da Computação pela Universidade Federal de Santa Catarina em 2007; Atualmente, doutoranda em Ciência da Computação pela Universidade Federal do Rio Grande do Sul. Tem 9 anos de experiência em docência e atualmente é professora do Instituto Federal Sul-Rio-Grandense. A área de atuação é em Sistemas Distribuídos, com interesse em sistemas para web e computação móvel.

<<http://buscatextual.cnpq.br/buscatextual/visualizacv.jsp?id=K4127507Y1>>



tics



Introdução ao desenvolvimento para Web

Unidade A
Linguagem de Programação Web

UNIDADE

A

INTRODUÇÃO AO DESENVOLVIMENTO PARA WEB

Introdução

Atualmente, acessamos a Web para obter diversos serviços, tais como:

- comunicar com colegas e amigos via softwares com MSN, skype, gtalk ou pelas redes sociais
- fazer compras (de quase todos os produtos)
- ler notícias
- entregar a declaração de IR
- consultar dados bancários e/ou realizar transações
- consultar a lista de filmes em cartaz no cinema
- consultar a programação da televisão
- consultar a previsão do tempo
- fazer inscrição para o processo seletivo
- fazer inscrição para o Enem
- ouvir rádio, procurar emprego, ...

Enfim, utilizamos este recurso com inúmeros propósitos diferentes e a tendência é que cada vez mais serviços sejam disponibilizados. Bem, mas então precisamos entender melhor como funciona, os conceitos e as possibilidades de aplicação para que possamos ser profissionais qualificados na criação de soluções para o ambiente Web.

Para começar, precisamos saber diferenciar Internet de Web. Você sabe?

A Internet e a World Wide Web

Internet

Você já deve ter ouvido/lido que a internet tem mais de 50 anos. Pois bem, por que a Internet existe há tanto tempo, mas o uso dos seus recursos é bem mais recente?

Isso porque a Internet é

a grande rede de computadores, ou seja, o conjunto de computadores que estão conectados através de várias redes. Então podemos entender a Internet como a infraestrutura.

A Internet surgiu em meados de 1969 com um projeto do governo americano denominado ARPANET. O objetivo do projeto era interligar universidades e instituições de pesquisa militares.

Web

E a Web? Bem, ela é bem mais nova. Em 1991, Tim Berners-Lee, pesquisador do CERN, lançou a WWW (*World Wide Web* – rede de alcance mundial) ou simplesmente, Web, com o objetivo de compartilhar arquivos (HTML e outros), tendo o navegador como ferramenta de acesso. Assim, a Web é

é um serviço da Internet, um ambiente onde os documentos são publicados, disponibilizados e acessados.

Então, podemos dizer que a web usa a Internet, mas ela em si não é a Internet. Sempre que acessamos um site, estamos navegando na web, mas quando usamos o MSN, por exemplo, estamos usando um software que usa a Internet e não necessariamente a Web.

Os documentos disponibilizados na Web podem estar na forma de vídeos, sons, hipertextos e figuras. Para visualizar as informações, usamos o navegador (*browser*) para descarregar os documentos (ou “páginas”) de servidores web (ou “sítios”) e mostrá-los ao usuário. O usuário pode então seguir as hiperligações na página para outros documentos ou mesmo enviar informações de volta para o servidor para interagir com ele. Esta funcionalidade é baseada em três padrões: URI, HTTP e HTML. Vamos ver mais detalhadamente cada um deles.

Atenção:

Internet e internet possuem significados diferentes. Enquanto internet significa um conjunto de redes de computadores interligadas, a Internet se refere à internet global e pública. Assim, existem inúmeras internets espalhadas por redes particulares, seja interligando empresas, universidades ou residências. Porém, existe somente uma rede única e global, o conjunto de todas as redes, a Internet.

URI - Universal Resource Identifier (Identificador Universal de Recursos)

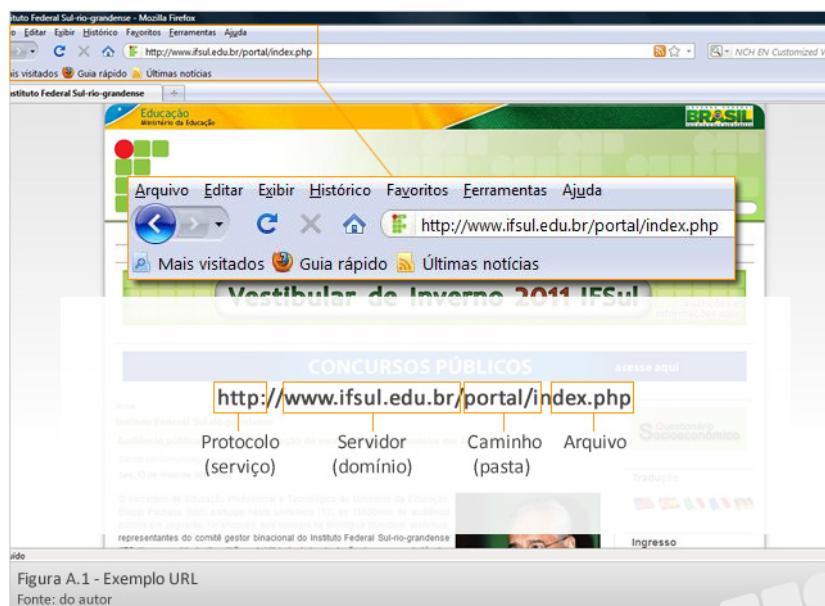
URI é a base do sistema de endereçamento da Web que usa uma cadeia de caracteres para identificar ou denominar um recurso na Internet. Um URI pode ser classificado como um localizador (URL) ou um nome (URN), ou ainda como ambos.

Um URN (*Uniform Resource Name* - Nome de Recursos Uniforme) é como o nome de uma pessoa, enquanto que um URL (*Uniform Resource Locator* - Localização de Recursos Uniforme) é como o seu endereço.

Os endereços que utilizamos atualmente na Web são os URLs, que possuem a seguinte estrutura:

- o protocolo de acesso ao recurso desejado (http)
- a máquina a ser conectada (www.ifsul.edu.br)
- o caminho de pastas até o recurso (portal/)
- o recurso (arquivo) a ser obtido (index.php)

Veja exemplo apresentado na figura A.1 abaixo:



HTTP (HyperText Transfer Protocol - Protocolo de Transferência de Hipertexto)

O HTTP é o protocolo que especifica como o navegador e o servidor web se comunicam. Cada vez que você aciona um link, seu navegador realiza uma comunicação com um servidor da Web através deste protocolo.

HTML (HyperText Markup Language - Linguagem de Marcação de Hipertexto)

O HTML é uma linguagem de marcação utilizada para produzir documentos que são interpretados por navegadores. O HTML mescla textos com códigos especiais (tags ou etiquetas).

As etiquetas (ou tags) permitem estruturar o conteúdo do documento e dar instruções ao navegador sobre o que fazer. As etiquetas são elementos entre parênteses angulares (< e >). A maioria das etiquetas tem sua correspondente de fechamento:

```
<etiqueta>...</etiqueta>
```

Exemplos:

```
<p> Introdução ao desenvolvimento para Web </p>
```

Acima com a etiqueta <p></p> o texto será apresentado em um novo parágrafo.

```
<b> Linguagem de Marcação de Hipertexto </b>
```

Com a etiqueta texto será apresentado em negrito.

Como podemos perceber vários novos conceitos vão sendo apresentados quando começamos a falar em Web e Internet. Por isso, esse documento se destina a apresentar várias informações importantes que devemos saber antes de iniciarmos o aprendizado específico das tecnologias para programação. A seguir vamos falar com pouco sobre: W3C, XHTML, CSS, linguagens do lado cliente e servidor e sobre tendências para a Web.

W3C

Conhecer o que é o W3C (*World Wide Web Consortium*) é muito importante, pois ele é responsável por desenvolver os padrões para criação e a interpretação de conteúdos para a Web. O W3C é um consórcio internacional com cerca de 300 membros, que agrega empresas, órgãos governamentais e organizações independentes, fundado por Tim Berners-Lee em 1994. O objetivo principal é desenvolver protocolos comuns e fóruns abertos que promovam a evolução da Web e assegurem a sua interoperabilidade. Assim, os sites que seguem esses padrões podem ser acessados e visualizados por qualquer pessoa ou tecnologia, independente de hardware ou software utilizados, de maneira rápida e compatível com os novos padrões e tecnologias que possam surgir.

Padrões seus como HTML, XHTML e CSS (que vamos mencionar mais adiante) são muito populares, contudo, em muitos casos são usados de forma errônea devido ao desconhecimento da especificação. É um dever de todo o desenvolvedor Web respeitar e seguir os padrões de acessibilidade do W3C, pois de outro modo poderá impor barreiras tecnológicas a diversas pessoas, desestimulando e até mesmo impedindo o acesso a suas páginas.

Saiba mais:

Para saber mais sobre o W3C acesse:

[<www.w3c.org>](http://www.w3c.org)

Uma iniciativa interessante do W3C é um validador de páginas HTML que foi disponibilizado com o objetivo de auxiliar os desenvolvedores na tarefa de verificar e corrigir os códigos criados por eles. O endereço é <http://validator.w3.org/>.

XHTML

XHTML? Por que este X? Bem, a XHTML (*EXtensible HyperText Markup Language* - Linguagem Extensível para Marcação de Hipertexto) foi criada para substituir a HTML, sendo uma linguagem mais “pura, clara e limpa”

A XHTML é uma reformulação da HTML, baseada em XML (*EXtensible Markup Language*). A XML tem como princípio que você pode criar seus próprios elementos de marcação e atributos para escrever seu documento. Isto significa que é você quem cria sua linguagem de marcação. A transformação de um documento existente de HTML para XHTML é uma tarefa bem simples, uma vez que a XHTML é baseada na HTML.

Mas o que muda de fato?

Esse processo de padronização tem em vista a exibição de páginas web em diferentes dispositivos (televisão, smartphone, celular, tablet).

As principais diferenças são:

- todas as tags devem ser escritas em letras minúsculas;
- as tags devem estar convenientemente aninhadas;
- os documentos devem ser bem formados;
- o uso de tags de fechamento é obrigatório;
- elementos vazios devem ser fechados;
- sintaxe para atributos.

CSS

Talvez você nunca tenha ouvido falar do CSS (*Cascading Style Sheets* - Folha de Estilos em Cascata) e nem tenha percebido a sua aplicação, mas praticamente todos os sites que você acessa usam este recurso. O CSS é também uma linguagem, mas esta é usada para criar estilos que definem o layout do documento HTML. Por exemplo, estilos para fontes, cores, margens, linhas, alturas, larguras, imagens de fundo, posicionamentos e muito mais.

O CSS é suportado por qualquer navegador e proporciona inúmeras opções de formatação para apresentação do layout de forma bastante sofisticada.

É fundamental que um desenvolvedor web saiba aplicar os recursos desta linguagem que foi proposta justamente como objetivo de separar o estilo do conteúdo da página.

Talvez você se pergunte se não pode fazer isso com HTML. Bem, a questão é que a recomendação W3C e os padrões de desenvolvimento indicam que o HTML deve ser usado para estruturar conteúdos e o CSS para formatar conteúdos estruturados.

Os benefícios do uso de CSS são:

- controle do layout de vários documentos a partir de uma simples folha de estilos;
- maior precisão no controle do layout;
- aplicação de diferentes layouts para servir diferentes mídias (tela, impressora, etc.);
- emprego de variadas, sofisticadas e avançadas técnicas de desenvolvimento que facilitam a manutenção.

Atenção:

Esta disciplina não prevê abordar a linguagem CSS. No entanto, é importante que o aluno compreenda sua aplicação, até porque na linguagem HTML nos deteremos as tags que tratam apenas da estrutura do documento.

Linguagens de Programação para Web: client-side e server-side

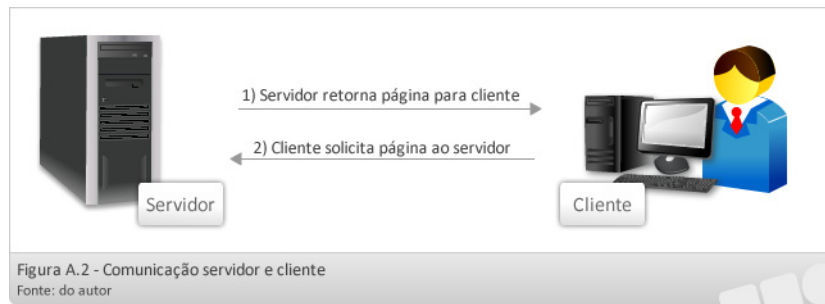
Nesta seção vamos falar sobre dois tipos de linguagens que são usadas no desenvolvimento para Web e diferenciar em que situações devemos usar cada uma delas. Primeiro, precisamos saber o que significa client-side e server-side:

Client-side (lado cliente):

O computador do usuário que está acessando um site por meio do navegador.

Server-side(lado servidor):

Servidor que hospeda o site.

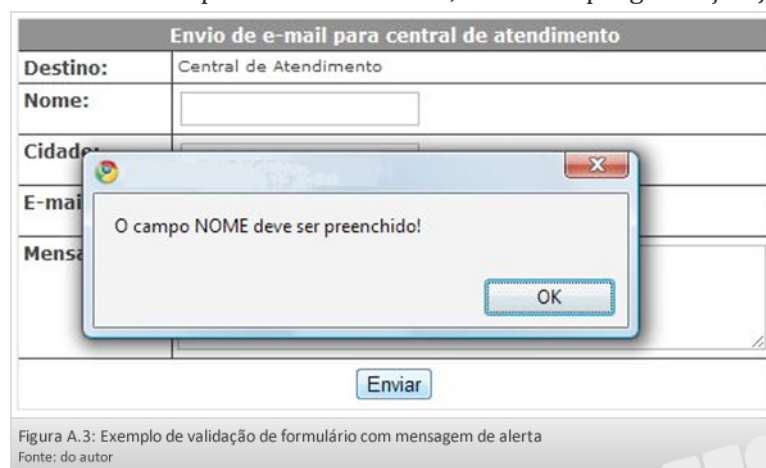


A figura A.2 apresenta o Cliente e o Servidor que interagem no processo de acesso ao um site.

Então quando falamos de uma **linguagem de script** (linguagem interpretada) **do lado cliente**, significa que esta linguagem está sendo interpretada pelo navegador do cliente.

Já uma **linguagem de script do lado servidor** tem seu código interpretado no servidor, sendo que o cliente recebe apenas o resultado do seu processamento.

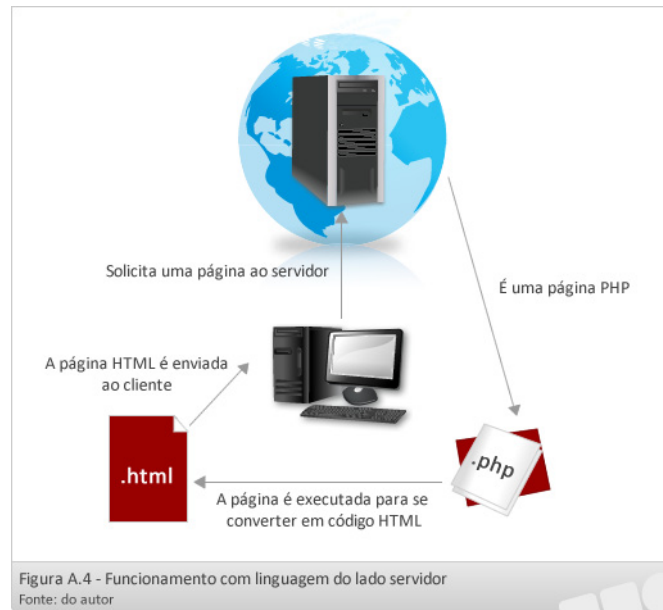
Com relação à linguagem do lado cliente, é utilizado o **JavaScript** para que seja possível manipular a página do usuário diretamente, tomando ações dinâmicas que vão desde emitir uma mensagem, validação de um formulário, abrir e configurar janelas, manipulação de objetos da página, detectar versão do navegador. A figura A.3 apresenta um exemplo de validação de formulário que apresenta mensagens ao usuário quando os campos obrigatórios não são preenchidos. Assim, para situações que necessitam de interação com o usuário e não dependem do servidor, usamos a programação JavaScript.



A linguagem do lado servidor é instalada no servidor e vai rodar “por trás dos panos”, fornecendo a lógica principal da aplicação.

A figura A.2 apresenta como ocorre o funcionamento:

- Quando o usuário faz uma requisição (entra numa página, clica num link, etc.), o pedido é enviado ao servidor.
- O servidor recebe a requisição e verifica se a página tem código que precisa ser interpretado. Em caso positivo faz o processamento.
- Depois, transforma o resultado final em um XHTML e envia ao navegador do cliente. Assim, o cliente já recebe tudo pronto.



Uma linguagem do lado servidor é necessária, por exemplo, quando precisamos interagir com um banco de dados (que está no servidor). Por exemplo, quando você acessa um site que publica notícias, muito provavelmente, este site armazena as notícias em uma base de dados e uma linguagem do lado servidor é utilizada para recuperar esses dados e gerar dinamicamente uma página HTML de visualização para o usuário.

Existem várias linguagens do lado servidor: PHP, JSP, ASP, Python. Nossa disciplina estudará a linguagem PHP (*Hypertext Preprocessor*) que é atualmente uma das linguagens mais utilizadas, open-source e independente de plataforma. Quando abordamos a linguagem falaremos mais sobre ela.

Tendências Web

Falaremos um pouco agora sobre alguns importantes conceitos que são tendências para a Web. É visível como as páginas Web evoluíram, melhoraram com o passar do tempo. Atualmente, os usuários têm mais liberdade, podem participar e interagir com blogs, redes sociais e outros serviços que estão cada vez mais presentes. Dois conceitos bem atuais relacionados a este novo cenário da Web é a Web 2.0 e a Web 3.0.

WEB 2.0

A Web 2.0 não é uma nova Web, mas sim um termo utilizado para denotar um novo conceito de aplicações, que reforça a ideia de troca de informações e colaboração dos internautas com sites e serviços virtuais. Neste conceito, o ambiente on-line se torna mais dinâmico e os usuários colaboram para a organização do conteúdo.

Alguns exemplos que estão alinhados a este novo conceito são a enciclopédia *Wikipedia*, cujas informações são disponibilizadas e editadas pelos próprios internautas, o Orkut, o Facebook, o Youtube, o Twitter, Flickr, LinkedIn, Blogs, entre outros.

Com a Web 2.0, o grande desafio é que os sites precisam lidar com um público cada vez mais ativo e presente, que deseja interagir. Por isso, a divulgação e a publicidade se tornaram cada vez mais necessários nesse cenário.

Associado a isso, outras tecnologias vêm surgindo e requerem que o desenvolvedor faça uso delas para aprimorar os sites. Vamos falar um pouquinho de uma tecnologia que os desenvolvedores para Web devem conhecer: AJAX.

AJAX (*Asynchronous Javascript and XML* - Assíncrono de Javascript e XML) é o uso metodológico de tecnologias como Javascript e XML com o objetivo de tornar páginas Web mais interativas com o usuário, utilizando-se de solicitações assíncronas de informações. Assíncrono se refere ao método para se comunicar com um servidor web sem a necessidade de recorrer ao recarregamento de página, não precisa haver sincronia do carregamento de dados com o carregamento da página Web. Isso permite que a navegação seja mais interativa para o usuário. Segundo Jesse James Garrett (inventor do termo), “AJAX não é uma tecnologia. São, não verdade, várias tecnologias, juntando-se de maneiras novas e poderosas.” A Google utiliza muito este recurso nas suas ferramentas, por exemplo, quando se pretende fazer um login e o usuário e/ou senha são inválidos. Para fazer essa verificação, a página faz uma solicitação ao servidor que retorna um resultado. A página trata esse resultado e mostra uma mensagem na página para o usuário, tudo isso sem recarregar a página (a figura A.4 apresenta um exemplo).

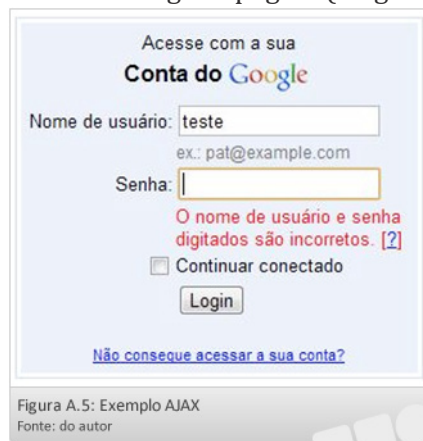


Figura A.5: Exemplo AJAX
Fonte: do autor

WEB 3.0

A Web 3.0 ainda não está presente, mas é também um novo conceito de Web que pretende ser a organização e o uso de maneira mais inteligente de todo o conhecimento já disponível na Web. Este paradigma surgiu a partir da necessidade de desenvolvimento de programas que entendam como fazer melhor uso da grande quantidade de dados. Assim, o foco principal da Web 3.0 é a Web semântica.

A Web Semântica tem como premissa uma web com toda sua informação organizada de forma que não somente seres humanos possam entendê-la, mas principalmente máquinas. O objetivo é tornar tarefas como a pesquisa sobre filmes e sobre comida mais rápidas e mais fáceis. Em vez de múltiplas pesquisas, você poderá digitar uma ou duas frases complexas no seu navegador e este irá analisar sua pergunta, pesquisar por todas as respostas possíveis e então, organizar os resultados para você.

Síntese

Bem pessoal, creio que foi possível observar que o desenvolvimento para Web requer o conhecimento de diversas tecnologias, cada uma com seu propósito de aplicação. Esta disciplina terá como enfoque principal preparar o aluno para a parte de programação de aplicações Web com linguagem do lado cliente (JavaScript) e linguagem do lado servidor (PHP). Assim, poderemos criar soluções que acompanham a tendência para Web com recursos interessantes e que possibilitam facilidade de uso ao usuário.

Atividades

Agora que você conhece as principais tecnologias que envolvem o desenvolvimento para Web, você está desafiado a olhar diferente para os sites que acessa.

Inicialmente, acesse o vídeo que foi disponibilizado sobre a “A evolução da Web”.

Após, acesse vários sites que você já conhece e faça uma análise para depois discutirmos sobre alguns aspectos conforme as questões norteadoras abaixo:

1. Considerando os conceitos e tecnologias abordados, você já possuía tais conhecimentos?
2. Quando você navega em sites identifica os recursos que foram apresentados?
3. Você compreende por que são necessárias várias tecnologias para o desenvolvimento e a aplicação de cada uma?
4. Consegue identificar situações em que o site aplica uma linguagem do lado cliente?
5. Consegue identificar se o site usa uma linguagem do lado servidor?
6. Pesquise mais informações sobre Web 2.0 e Web 3.0 e compartilhe com seus colegas.
7. Faça uma pesquisa sobre o que é RSS (também uma tendência da Web 2.0) e compartilhe com seus colegas.

Introdução ao HTML

Esta unidade tem objetivo fazer uma revisão da estrutura principal da linguagem (X)HTML, focando principalmente em formulários. Estes conhecimentos servirão de base para as próximas unidades.

Abordaremos a XHTML a fim de seguir as recomendações da W3C, pois desta forma você estará apto a construir páginas que tenham como características a rapidez no carregamento, a facilidade de manutenção, acessibilidade aos mais variados agentes de usuário e cuja estrutura esteja separada da apresentação e estilização.

A grafia (X)HTML é utilizada para se fazer referência tanto à HTML quanto à XHTML. Todos os elementos que serão apresentados são válidos para a HTML. No entanto, seguiremos a especificação da XHTML que é mais rigorosa. Por exemplo, ao escrevermos HTML, é correta a grafia dos elementos tanto em minúsculo como em maiúsculo, mas ao escrevermos em XHTML, só é válida a grafia de elementos em letras minúsculas.

As versão atual da HTML é a 4.01 e da XHTML é a 1.1. Porém está em elaboração e promete profundas alterações a especificação para a HTML 5.0.

Para criar documentos (X)HTML você pode usar um simples editor de texto como o Bloco de Notas. Existem ferramentas que auxiliam no desenvolvimento e dão suporte visual, os chamados editores WYSIWYG (What You See Is What You Get - O que você vê é o que você obtém). Um exemplo é a ferramenta Dreamweaver da Adobe bastante utilizada por desenvolvedores. Porém, cabe destacar que é uma ferramenta que precisa de licença de uso. Por hora vamos usar o bloco de notas para aprender a (X)HTML.

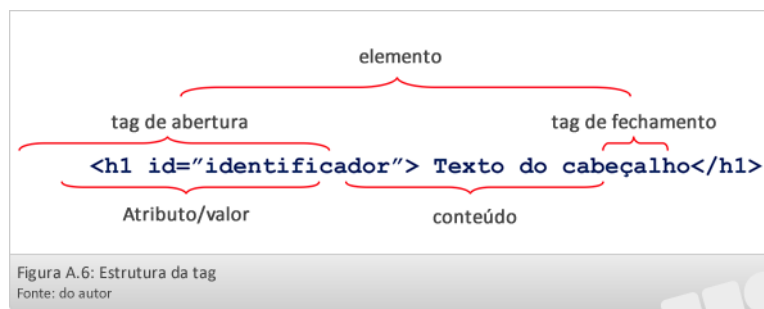
Estrutura do (X)HTML

Antes de qualquer coisa, é necessário conhecer o significado de alguns termos que vamos usar daqui para frente.

Termos	Significado
Renderização	Transformação da marcação (X)HTML do documento em algo capaz de ser lido e entendido pelo usuário. A renderização é feita pelo navegador. Quando visualizamos uma página é porque o navegador renderizou a página para nós.
Documento e página	Documento é o arquivo completo que contém toda a marcação. A página web ou simplesmente página é o resultado da renderização do documento.
Elemento	É a menor unidade de marcação. Cada elemento tem uma destinação específica, por exemplo, para marcar títulos, devemos obrigatoriamente usar o elemento previsto para marcação de títulos; para marcar parágrafos existe um elemento específico também, e assim por diante.
Elemento vazio	São representados apenas pela tag(etiqueta) de abertura, não existe a tag de fechamento. Estes elementos não possuem conteúdos para marcar e se destinam a fornecer informações adicionais.

Tags (ou etiquetas)

Os elementos de marcação são representados por um par de tags(ou etiquetas) que são os sinais < e >. Geralmente, na marcação usamos uma tag de abertura e uma tag de fechamento, entre as quais se insere o conteúdo. A tag de fechamento possui por sua vez uma / (**barra**) logo após o sinal de <. Veja na figura A.6 a estrutura:



Observe que dentro da tag de abertura foi inserido o atributo **id** com o valor “*identificador*”. Atributos são usados para definir uma propriedade de um elemento. Eles devem ser colocados sempre na tag de abertura, logo após o nome do elemento, precedido de um espaço. O atributo é composto de um nome, um sinal de igual (=) e um valor de atributo, cercado por aspas duplas (") ou simples ('). No caso do nosso exemplo, o atributo é o **id**, que serve para identificar, de maneira única, um elemento dentro de um documento (X)HTML.

Comentários

Muitas vezes é necessário incluir comentários no nosso documento (X)HTML. Nesse caso podemos usar da seguinte forma:

```
<!-- comentário -->
```

Para iniciar o comentário usamos o marcado <!-- e para fechar -->. Assim, o navegador irá ignorar o texto incluído entre as marcações de comentário.

Estrutura mínima de um documento

Um documento (X)HTML possui uma estrutura mínima dividida em três partes, ou seja, uma espécie de esqueleto do documento. Vejamos:

1. seção de declaração do tipo de (X)HTML
2. seção head (cabeçalho)
3. seção body (corpo)

A figura A.7 apresenta a estrutura mínima de um documento documento. A linha 1 contém a primeira parte do documento que declara o tipo de (X)HTML ou *doctype*. Após, na linha 2 está o elemento `<html>` que é aberto no início do documento e fechado apenas no fim. Dentro do elemento `<html>` estão as outras duas seções. A seção do cabeçalho do documento é aberta na linha 3 e fechada na linha 5, indicada pelo elemento `<head></head>`. Já a seção do corpo do documento é aberta na linha 7 e fechada na linha 9, indicado pelo elemento `<body></body>`.

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <title>Exemplo</title>
5 </head>
6
7 <body>
8 texto do documento...|
9 </body>
10 </html>
```

Figura A.7 - Estrutura mínima de um documento (X)HTML
Fonte: do autor

Doctype

Doctype é a declaração do tipo de documento e é escrita na primeira linha do documento. O doctype da linha 1 da imagem A.5 quer dizer o seguinte: *a marcação para o presente documento está escrita em linguagem html pública, segundo as regras estabelecidas pelo W3C para XHTML 1.0 Transitional. O conjunto de elementos e atributos, bem como as regras de sintaxe empregadas neste documento, está de acordo com o que prescreve o documento hospedado no site do W3C no seguinte endereço: <http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>*

Veja na tabela abaixo os tipos de *doctype*:

Strict	<i>doctype</i> mais rígido e possibilita um código mais limpo. É usada em conjunto com estilos CSS (<i>Cascading Style Sheets</i>). Não pode usar elementos e atributos declarados em desuso pelo W3C e nem marcação de frames.
Transitional	<i>doctype</i> mais flexível e deve ser usado para possibilitar uma navegação por navegadores mais antigos ou que não suportam CSS por exemplo. Não pode conter elementos ou atributos destinados a marcação de frames.
Frameset	deve ser usada quando o documento contém molduras (frames).

Atenção:

Sobre frames é importante dizer que o seu uso está sendo abandonado por ter restrições à acessibilidade.

Head

A seção *head* contém elementos com informações adicionais sobre o documento. Um elemento obrigatório é o título do documento (`<title></title>`). O conteúdo do elemento *title* será colocado na barra superior da janela do navegador. Os elementos podem aparecer no cabeçalho, alguns recomendados pela W3C, por exemplo, para informar qual o conjunto de caracteres usados na marcação.

Body

A seção *body* contém os elementos que marcam os conteúdos que serão visualizados no navegador. Tudo o que você vê no navegador está entre os elementos `<body></body>`.

Principais Elementos (X)HTML

Agora você irá conhecer os principais elementos do (X)HTML. A ideia não é apresentar todos, mas aqueles que são mais utilizados e que dizem respeito a parte estrutural do documento e seu conteúdo.

Elemento	Finalidade
<code><h1></code> título de nível 1 <code></h1></code> <code><h2></code> título de nível 2 <code></h2></code> ... <code><h6></code> título de nível 6 <code></h6></code>	abertura e fechamento de um cabeçalho. São 6 níveis: h1 (o maior) até h6 (o menor).
<code><p></code> texto <code></p></code>	inicia um parágrafo pulando uma linha.
<code>
</code>	quebra de linha

<code><hr/></code>	cria uma linha divisória horizontal.
<code><div> texto </div></code>	abertura e fechamento de uma divisão de página cujos elementos que estão dentro dela obedecerão às mesmas definições.
<code> texto do link </code>	Estabelece um texto ou imagem como link. Atributo
<code></code>	inserção de imagem. Atributos: src ⇒ caminho e nome da imagem width ⇒ largura da imagem em pixels height ⇒ altura da imagem em pixels alt ⇒ texto alternativo a imagem (importante para acessibilidade).

Atributos

Como já foi citado acima, os elementos podem conter atributos que referenciam características ou dados complementares do elemento. Cada elemento pode conter atributos diferentes. Um exemplo, é o atributo `align` que pode ser aplicado a vários elementos e tem por objetivo o alinhamento do conteúdo do elemento. As opções são *left*, *center*, *right*. Ele pode ser aplicado, por exemplo, aos elementos *p*, *h1*, *h2*... *h6*, *hr*, *div*, *table*.

Ex.:

```
<p align='center'>texto </p>
```

Links

Sobre a criação de links existem possibilidades diferentes de criação dependendo o destino do link:

- **links internos:** são aqueles que apontam para uma página dentro do próprio site. É necessário informar apenas o local e nome do arquivo dentro do site.

```
<a href="produtos.html">Página de produtos</a>
```

- **links externos:** são aqueles que apontam para fora do próprio site. Neste caso devemos usar o endereço completo da página destino(URL). Observe que foi usado o atributo `target='_blank'` para indicar que o endereço deve ser aberto em uma nova janela.

```
<a href="http://www.w3c.org" target="_blank">W3C</a>
```

- **link para e-mail:** são aqueles que apontam para um endereço de e-mail. Neste caso usamos a `mailto:` antes de indicar o e-mail.

```
<a href="mailto:testes@provedor.com.br">Email Teste</a>
```

Tabelas

Em (X)HTML utilizamos bastante tabelas, por isso é importante reconhecer toda a sua estrutura. Vejamos a tabela mostrada na figura A.8, como será o (X)HTML desta tabela?

linha 1 coluna 1	linha 1 coluna 2
linha 2 coluna 1	linha 2 coluna 2
linha 3 coluna 1	linha 3 coluna 2

Figura A.8 - Tabela
Fonte: do autor

A figura A.9 apresenta os elementos (X)HTML da tabela visualizada na figura A.8. Agora vejamos os detalhes:

- O elemento *table* indica a criação de uma tabela. Abre com `<table>` e fecha com `</table>`. Abre na linha 10 e fecha na linha 23.
- para cada linha da tabela (são 3) existe um elemento *tr*. `<tr>` abre linha e `</tr>` fecha a linha.
- dentro de cada linha está o elemento da coluna (*td*). Neste caso a tabela possui duas colunas, então dentro de cada linha existem dois elementos *td*. `<td>` abre a coluna e `</td>` fecha coluna. É importante destacar que os dados da tabela sempre estarão dentro do elemento *td*.

Dicas:

`tr` ⇒ table row (linha da tabela) | `td` ⇒ table data (dados da tabela)

```

10 <table width="600" border="1" align="center" cellpadding="2" cellspacing="0">
11 <tr>
12 <td>linha 1 coluna 1</td>
13 <td>linha 1 coluna 2</td>
14 </tr>
15 <tr>
16 <td>linha 2 coluna 1</td>
17 <td>linha 2 coluna 2</td>
18 </tr>
19 <tr>
20 <td>linha 3 coluna 1</td>
21 <td>linha 3 coluna 2</td>
22 </tr>
23 </table>

```

Figura A.9 - (X)HTML da tabela
Fonte: do autor

Vejamos novamente com algumas anotações:

Diagrama de anotações sobre o código HTML da tabela:

- Largura da tabela. Pode ser em pixels ou %**: aponta para `width="600"`.
- Largura da borda**: aponta para `border="1"`.
- Alinhamento da tabela. Pode ser left, right ou center**: aponta para `align="center"`.
- Espaçamento entre conteúdo da célula e o seu contorno**: aponta para `cellpadding="2"`.
- Espaçamento entre as células**: aponta para `cellspacing="0"`.
- Segunda coluna da terceira linha**: aponta para o elemento `<td>linha 3 coluna 2</td>`.

Figura A.10 - Estrutura do (X)HTML da tabela
Fonte: do autor

Vamos ver em mais detalhes agora os atributos `cellpadding` e `cellspacing` do elemento `table`. Veja a figura A.11:

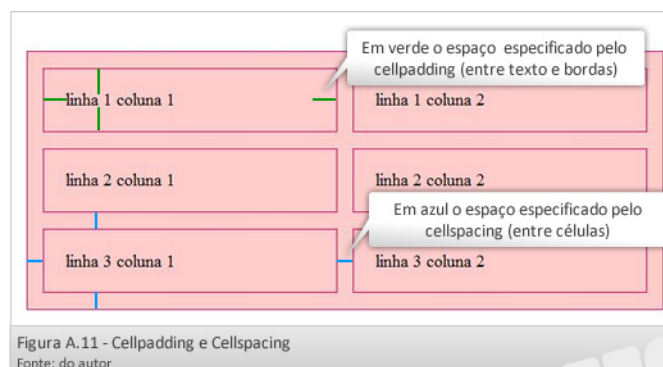
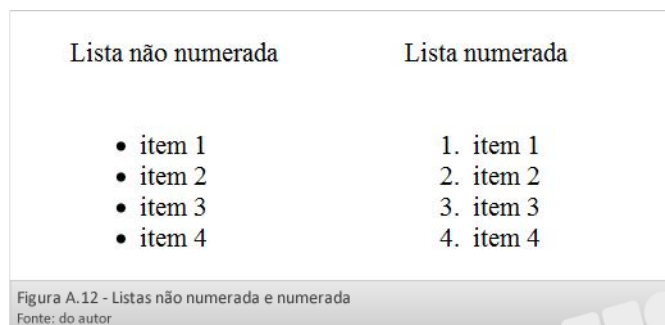


Figura A.11 - Cellpadding e Cellspacing
Fonte: do autor

Listas

Vamos trabalhar um pouco com listas. Podemos criar listas não numeradas ou numeradas. Veja no exemplo da figura A.12.



Para criar uma lista não numerada usamos o elemento ``. Para cada item da lista usaremos o elemento ``. Veja na figura A.13 as marcações para criação da lista não numerada.

```

1 <ul>
2   <li>item 1</li>
3   <li>item 2</li>
4   <li>item 3</li>
5   <li>item 4</li>
6 </ul>

```

Figura A.13 - (X) HTML da lista não numerada
Fonte: do autor

Para criar uma lista numerada usamos o elemento ``. Para cada item da lista usaremos o elemento ``. Veja na figura A.14 as marcações para criação da lista numerada.

```

1 <ol>
2   <li>item 1</li>
3   <li>item 2</li>
4   <li>item 3</li>
5   <li>item 4</li>
6 </ol>

```

Figura A.14 - (X) HTML da lista numerada
Fonte: do autor

Dicas:

ul ⇒ unordered list | ol ⇒ ordered list | li ⇒ list item

Formulários em (X)HTML

Para que possamos interagir com o usuário e criar soluções que usem as linguagens do lado cliente e servidor, é fundamental trabalhar com formulários (X)HTML. Por isso, vamos ver os principais elementos de formulário. Para começar vejamos um exemplo de formulário na figura A.15. Um formulário tem como objetivo, geralmente, possibilitar a entrada de dados. Neste exemplo da figura há apenas uma caixa de texto e um botão.

The figure shows a simple web form. It consists of a label 'E-mail:' followed by a text input field. To the right of the input field is a button labeled 'ok'. Below the form is a caption: 'Figura A.15 - Exemplo Formulário' and 'Fonte: do autor'.

Vejamos o (X)HTML do formulário acima na figura A.16. Todo formulário inicia com o elemento

form. Todos os elementos vinculados ao formulário ficam entre `<form></form>`. Os elementos HTML basicamente usados para marcação de formulários são:

- *form* que é o “container” para os demais elementos;
- *input* que pode ser de variados tipos como: *text*, *submit*, *button*, *radio button* e *checkbox*;
- *textarea* para entrada de textos *multi-linhas*, tais como comentários;
- *select* também conhecido como *drop-down-list*, ou menu de lista;

```

1 <form name="form1" method="post" action="">
2
3   <label for="email">E-mail:</label>
4
5   <input name="email" type="text" id="email"
6     size="40" maxlength="50">
7
8   <input type="button" name="botao" id="botao"
9     value="ok">
10
11 </form>

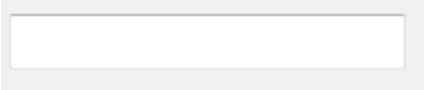
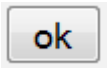
```

Figura A.16 - (X)HTML do formulário da figura A.15
Fonte: do autor

Todos os elementos de formulário devem possuir os atributos *id* e *name*. Eles serão necessários para manipulação dos valores tanto em programação do lado cliente (*Javascript*) quando em programação do lado servidor (PHP). O atributo *id* é o identificador do elemento, ou seja, sempre será um valor único para cada elemento do documento. Geralmente, o *id* e *name* possuirão o mesmo valor (exceção é para o tipo *radio*).

No exemplo da figura A.16, vimos o elemento *label* que é usado para especificar etiquetas para os controles. O atributo *for* associa a etiqueta de um campo de formulário. O valor do atributo *for* deve ser o valor do atributo *id* ou *name* do elemento que se deseja referenciar. No nosso exemplo o *label* é referente a caixa de texto chamada *email*.

A tabela abaixo apresenta as variações para os elementos *input*, *select* e *textarea*.

Elemento	Código	
<code>input type="text"</code>	<code><input type="text" name="texto" id="texto" /></code>	Caixa de texto: 
<code>input type="password"</code>	<code><input type="password" id="texto2" name="texto2" value="senha" /></code>	
<code>input type="hidden"</code>	<code><input type="hidden" id="oculto" name="oculto" value="1" /></code>	Campo oculto (não aparece). Mas possui <i>id</i> e valor. Usado para passar alguma informação que não precise ser visualizada.
<code>input type="button"</code>	<code><input type="button" name="botao" id="botao" value="ok"></code>	Botão: 

input type="submit"	<pre><input type="submit" name="botao2" id="botao2" value="Enviar" /></pre>	Botão do tipo submit: envia o formulário executando a ação do form 
input type="reset"	<pre><input type="reset" name="botao3" id="botao3" value="Limpar" /></pre>	Botão do tipo reset: limpa os dados do form, voltando ao estado inicial. 
input type="checkbox"	<pre><input type="checkbox" id="status" name="status" value="S" /></pre>	Checkbox: quando selecionado conterá o valor do atributo value. 
input type="radio"	<pre><input type="radio" name="sexo" value="F" id="sexo_0" /> Feminino
 <input type="radio" name="sexo" value="M" id="sexo_1" /> Masculino</pre>	Botões de rádio: devem possuir o mesmo name para que ao marcar uma opção a outra seja desmarcada. 
input type="file"	<pre><input type="file" name="arquivo" id="arquivo" /></pre>	Arquivo: Seleção de um arquivo 
select	<pre><select name="UF" id="UF"> <option value="RS">Rio Grande do Sul</ option> <option value="SC">Santa Catarina</option> <option value="PR">Paraná</ option> </select></pre>	Menu de lista: para cada opção possui um elemento option. Quando selecionado conterá o valor do atributo value 
textarea	<pre><textarea name="texto3" id="texto3" cols="20" rows="5"></textarea></pre>	Área de texto: rows: número de linhas cols: largura em termos de colunas 

Atributos do elemento form:

action = url

Este atributo indica o endereço de envio quando o formulário for processado. Geralmente uma outra página do site é especificada.

method = get/post

Este atributo indica sob qual forma será enviada as respostas: “POST” é o valor que corresponde ao envio de dados armazenados no corpo do pedido, enquanto “GET” corresponde ao envio dos dados codificados no URL e separados do endereço por um ponto de interrogação. O método POST é recomendado por ser uma forma mais segura de envio dos dados. No caso do método GET (envio dos dados através do URL), o URL assemelhar-se-á à uma cadeia do tipo:

<http://www.dominioteste.com.br?campo1=val1&campo2=val2>

enctype = content-type

Este atributo especifica o tipo de conteúdo usado para submeter o formulário no servidor (quando o valor de *method* for igual a “*post*”). O valor por padrão é “*application/x-www-form-urlencoded*”. O valor “*multipart/form-data*” deve ser usado em combinação com o elemento *input type="file"*

Síntese

Bem, agora já conhecemos a estrutura básica da linguagem e seus principais elementos. De agora em diante podemos começar a criar documentos (X)HTML. É importante lembrar que precisamos seguir com rigor a especificação da linguagem para que nossos documentos sejam considerados válidos. Vale lembrar também que existe bastante material disponível sobre (X)HTML, por isso, agora é exercitar e pesquisar sobre outros elementos e atributos.

Atividades

1. Como vimos, o XHTML é mais rigoroso que o HTML quanto a estrutura dos elementos. Dada a tabela abaixo, apresente a estrutura correta considerando as restrições do XHTML.

Errado	Correto
<code><DIV><P>Aqui está errado!</P></DIV></code>	
<code><div><p>texto em negrito </p></div></code>	
<code><p>um parágrafo <p> outro parágrafo</code>	
<code>
</code>	
<code><hr></code>	
<code><td width=300></code>	
<code><input checked/></code>	

2. Dado o código html abaixo, encontre e corrija os erros existentes.

Linha	Código
1	<code><html></code>
2	<code><title>Atividade - LPW</title></code>
3	<code><head></code>
4	<code></head></code>
5	<code><body></code>
6	<code>IFSUL</code>
7	<code></code>
8	<code>
</code>
9	<code><table width="250" border="1" cellspacing="0" cellpadding="3"></code>
10	<code><tr></code>
11	<code><td>Código</code>
12	<code><td>Descrição</code>
13	<code><td>Preço</code>
14	<code></tr></code>
15	<code><tr></code>
16	<code><td>1</code>
17	<code><td>Impressora</code>
18	<code><td>R\$ 300,00</code>
19	<code></tr></code>
20	<code></table/></code>
21	<code></body></code>

3. Apresente o código para uma página HTML que contenha os seguintes elementos:

- Estrutura básica do documento (cabeçalho e corpo descritos corretamente)
- Título geral do documento (para exibição na barra superior do navegador)
- Título 1: Minha página;
- Um parágrafo de texto livre
- Título 2: Tabela de Calorias
- Tabela conforme abaixo:

Refrigerantes e energéticos	
Coca-cola	R\$ 2,50
Coca-cola light	R\$ 3,00
Fanta	R\$ 2,00

4. Apresente o código (X)HTML que corresponda a apresentação da página com o formato conforme a figura abaixo:

Nome:

E-mail:

☐ Aluno
☐ Professor
☐ Funcionário

Mensagem:

☐ Enviar cópia por e-mail

Figura A.17 - Figura do exercício 4
Fonte: do autor

5. Faça uma pesquisa sobre os atributos que podem ser aplicados aos elementos abaixo relacionados e apresente exemplos de uso:

img

a

input

ul

tr

6. Tomando por referência as páginas mostradas nas figuras abaixo crie os seguintes documentos (X)HTML:

- index.html
- links.html
- contato.html
- disciplinas.html

Obs.: os balões amarelos na imagens são apenas lembretes

