

1 Web design

1.1. O que é Web design?

Web design é, em inglês, o nome da arte praticada pelos seres artrópodes da ordem *Aracneae*, ou aracnídeos, mais popularmente conhecidos como as *aranhas*.

Mas isto era há 10 anos atrás. Hoje, Web design é a arte de tecer uma outra teia, muito mais desorganizada e caótica, e que não é governada por nenhuma aranha. Web design é a concepção e projeto da interface interativa do serviço Web, formado por "páginas".

Alguns anos depois das aranhas, a administração da Web cabia à programadores que precisavam codificar toda a informação usando uma linguagem: o HTML, e depois armazená-las em uma área especial de uma máquina Unix, onde rodava um servidor Web – programa que permitia o acesso remoto às informações das páginas. Hoje, a arte de criar páginas Web continua a exigir mais e mais conhecimentos de programação ... visual! Saber HTML hoje é menos importante que ter noções de design.

O design de páginas para a Web hoje se assemelha mais à editoração eletrônica que à programação. Os caminhos do Web design, porém, são bem diferentes daqueles seguidos na criação para mídia impressa. O Web designer deve conhecer não só as possibilidades do meio onde publicará a sua informação, como suas limitações. A arte do Web design consiste em aproveitar ao máximo os recursos oferecidos pela Web, garantindo a melhor apresentação, navegabilidade e interatividade de um Web site. A idéia é atrair os visitantes e estimulá-los a voltar outras vezes. Às vezes é necessário sacrificar a qualidade da apresentação ou deixar de usar algum recurso útil devido a lentidão da rede ou incompatibilidades de browsers. É importante que o Web designer aprenda a traçar uma linha de equilíbrio entre os impedimentos tecnológicos e as possibilidades criativas desta nova mídia, para tirar o maior proveito do seu potencial.

O bom Web design começa fora do computador. Planejar a estrutura da teia e sua identidade com base nos objetivos à que se destina (comercio, informação, promoção, intranet, captura de insetos) antes de iniciar a implementação, preserva o conceito por trás do design da interface e diminui a limitação da tecnologia disponível.

Como já mencionamos acima, hoje é possível desenvolver todo um site sem escrever uma só linha de HTML. Obviamente, omitimos alguns detalhes nesta afirmação (senão, por que estaríamos ensinando HTML avançado?). O estado atual das ferramentas de *Web publishing* limita muito o que se pode fazer sem programar. HTML é um padrão em constante desenvolvimento e as ferramentas levam no mínimo seis meses para incorporar as extensões e novos padrões aos seus produtos. Para criar páginas com os mais modernos recursos interativos, com uma apresentação sofisticada, acessível e fácil de manter quase sempre ainda é preciso escrever alguma coisa em HTML. É a única forma de não se tornar escravo das ferramentas.

1.1.1. *A internet e a world wide web*

Para entender como criar e publicar páginas para a Web, precisamos primeiro saber um pouco como funciona a Teia, e a Internet, que está por trás de tudo. Nos parágrafos seguintes relembremos um pouco a sua história.

A Internet é uma rede antiga. Tem mais de 30 anos de idade. Uma das pessoas que teve um papel decisivo na sua criação foi Fidel Castro. Os Estados Unidos tinham passado por uma possibilidade real de ataque nuclear depois da instalação de mísseis russos em Cuba e a interligação das bases militares em rede foi uma estratégia militar para proteger a comunicação em caso de ataque nuclear. Ela ligava máquinas diferentes entre si através de linhas redundantes de maneira que, mesmo que uma ou várias bases fossem reduzidas a pó, as outras estações ainda conseguiriam se comunicar entre si. Então, com medo que Fidel e seus charutos nucleares paralisassem o sistema de defesa do país foi criada a rede ARPANET (*Advanced Research Projects Agency Network*). A rede interligou vários computadores em algumas universidades e centros de pesquisa envolvidos com projetos militares. Na época, computador ainda era coisa rara. Computador em rede, mais ainda.

Os primeiros usuários (cientistas) usavam a rede para trocar mensagens de correio eletrônico e se conectar remotamente em computadores distantes. O serviço tornou-se tão útil, que as universidades envolvidas começaram a conectar seus departamentos, mesmo os que não tinham a ver com o projeto. Na década de 80, foi a vez da NSFNET, a rede nacional de pesquisa dos Estados Unidos, que conectou os cinco grandes centros regionais de supercomputação e passou também a fazer parte da ARPANET. A NSFNET se tornou espinha dorsal das duas redes, depois chamada de Internet.

A NSFNET levou consigo uma legião de pequenas redes que a ela estavam conectadas. E assim a Internet foi crescendo e crescendo, sem nenhuma organização central. Qualquer centro que achasse seus serviços convenientes podia se amarrar ao nó mais próximo simplesmente pagando as despesas de uma linha dedicada de dados. Em pouco tempo, a Internet já interligava os maiores centros de pesquisa do mundo.

A expansão foi trazendo novos serviços e outras redes de pesquisa, como a Usenet, a Bitnet, EARN e redes de BBSs. O crescimento já era impressionante, mas a popularidade ainda era limitada. Grande parte do tráfego era acadêmico. Operar um sistema de correio eletrônico em geral significava saber operar um jurássico terminal IBM ou máquina Unix, e ter alguma familiaridade com suas interfaces hostis. Para transferir um arquivo era necessário saber usar um programa especial (FTP), mandar comandos para ele e entender os hieróglifos que ele retornava.

O maior problema era a desorganização generalizada da Internet. A Internet era um tremendo caos. Você sabia que poderia transferir um arquivo de qualquer lugar do mundo. O problema era saber se ele existia e onde estava!

A Internet começou a crescer de forma explosiva em 1992, com o surgimento da *World Wide Web*, um projeto do laboratório CERN em Genebra (eles não estudavam aranhas, mas

partículas subatômicas). A Web conseguiu finalmente organizar um pouco as informações da Internet através do hipertexto. No ano seguinte, chegou o *Mosaic* para Windows – programa desenvolvido por um grupo de estagiários do NCSA (*National Center for Supercomputing Applications*) da Universidade de Illinois a partir de uma versão anterior para o sistema Unix.

O Mosaic, oferecendo pela primeira vez uma interface gráfica multimídia para a Web, trouxe a grande massa de usuários de PC e Macintosh para dentro da rede. Os provedores de acesso e informação comerciais se multiplicaram, oferecendo às pessoas comuns o mesmo acesso que antes só tinham as grandes organizações e o meio acadêmico.

Os criadores do Mosaic pouco depois de deixar a universidade criaram uma empresa: a Netscape, que foi provavelmente a empresa que teve a maior influência nos rumos seguidos pela Web na sua evolução até os dias de hoje.

1.1.2. *A web como um meio de comunicação*

Apesar de ter surgido como um serviço de uma rede de computadores, a Web é hoje muito mais que isto e para explorá-la, nem computador é necessário mais. Tecnologias recentes como o *Network Computer* (NC) - que é um simples terminal para a Web, browsers que vêm embutidos em telefones celulares, e a rede *WebTV* mostram que a *World Wide Web* está destinada a preencher todos os espaços da mídia de difusão, não se limitando àqueles que possuem um computador.

A Web, dessa forma, possui um potencial inigualável na história das telecomunicações. É capaz de servir de interface à todos os serviços da Internet e ainda aos tradicionais serviços de voz (telefone), televisão, rádio e mídias impressas. Pode integrar tudo e interagir com tudo. Ainda estamos dando os primeiros passos neste novo terreno, e cada dia nos traz mais certeza de que trata-se de um caminho sem volta.

Diferente dos meios tradicionais de comunicação de massa, a *World Wide Web* é uma mídia democrática. O usuário não precisa possuir uma estação difusora, uma concessão, uma gráfica ou qualquer coisa do tipo para poder publicar sua informação e influenciar sua audiência. Todos podem receber as informações de todos. Qualquer um pode prover informação. O poder da informação está nas mãos de todos os que puderem ter um espaço na Teia, e não mais apenas com as aranhas ou com aqueles que possuem os meios de difusão tradicionais.

1.2. *A World Wide Web e a Internet*

A *World Wide Web* é o nome do mais popular dos *serviços* da Internet. Por esse motivo, é freqüentemente confundida com a própria Internet. Mas Web e Internet não são a mesma coisa, e precisamos conhecer bem a diferença entre as duas antes que possamos começar a desenvolver páginas e aplicações para a Web.

Internet é o nome dado ao conjunto de computadores, provedores de acesso, satélites, cabos e serviços que formam uma *rede mundial* baseada em uma coleção de *protocolos de comunicação* conhecidos como TCP/IP.

1.2.1. Protocolos

É através de protocolos de comunicação que um computador pode se comunicar com outro através de uma linha telefônica ou placa de rede sem que o usuário precise se preocupar em saber qual o meio físico que está sendo utilizado. O sistema Windows possui protocolos que permitem facilmente interligar computadores rodando Windows entre si. Os mesmos protocolos podem não servir para fazer com que uma máquina Windows se comunique com uma máquina Unix ou Macintosh, pois essas máquinas possuem arquiteturas diferentes. TCP/IP é uma suite de protocolos padrão que foi adotado como “língua oficial” da Internet. Para fazer parte da Internet, um computador precisa saber se comunicar em TCP/IP. Todas as operações de rede são traduzidas para TCP/IP antes que possam funcionar na Internet.

1.2.2. Endereços Internet

Um dos protocolos mais importantes da suite TCP/IP é o protocolo de rede IP - *Internet Protocol*. Ele define a *forma de endereçamento* que permite a localização de um computador na Internet, através de um conjunto de dígitos chamado de *endereço IP*. Qualquer máquina acessível através da Internet tem um endereço IP *exclusivo*. Esse endereço pode ser *temporário* ou *permanente*. Quando você se conecta a um provedor via linha telefônica, ele atribui um número IP temporário à sua máquina que permitirá que ela faça parte da Internet enquanto durar a sua sessão no provedor. Só assim é possível receber informações em um browser ou enviar e-mail. Computadores que hospedam páginas Web e que oferecem outros serviços pela Internet precisam de um endereço IP *fixo*, para que você possa localizá-los a qualquer hora. Por exemplo, 200.231.191.10 é o endereço IP da máquina onde está localizado o servidor Web do IBPINET em São Paulo. Você pode localizá-lo digitando `http://200.231.191.10/` no campo de endereços do seu navegador.

1.2.3. Serviço de Nomes (DNS – Domain Name Service)

Embora cada computador seja identificado de forma exclusiva através de um endereço IP, não é dessa forma que costumamos localizá-los na Internet. Um dos serviços fundamentais ao funcionamento da Internet é o *serviço de nomes de domínio*. Esse serviço é oferecido por várias máquinas espalhadas pela Internet e que guardam tabelas que associam o *nome* de uma máquina ou de uma rede a um endereço IP. Quando você digita o nome de uma máquina no seu browser (por exemplo, `www.ibpinet.net`), o browser primeiro tenta localizá-la consultando uma outra máquina (cujo endereço IP o browser já conhece) que oferece o serviço de nomes. Essa máquina consulta outros serviços de nomes espalhados pela Internet e em pouco tempo

devolve o endereço IP correspondente ao nome solicitado (*www.ibpinet.net* devolverá 200.231.191.10).

Se o sistema de nomes falhar, o browser não conseguirá o número IP que precisa e assim não localizará a máquina correspondente (mesmo que ela não esteja fora do ar).

1.2.4. *Portas e serviços da Internet*

A Internet existe há mais de três décadas. Na maior parte desse período ela era restrita aos meios acadêmicos e militares e oferecia poucos serviços. Os principais *serviços* utilizados na rede eram a *transferência de arquivos* entre computadores (usando aplicações que se comunicavam através do protocolo FTP - *File Transfer Protocol*), o *correio eletrônico* e a *emulação de terminal*, que permitia o acesso a computadores remotos. Esses serviços eram oferecidos em algumas máquinas onde rodavam programas servidores, permanentemente no ar aguardando a conexão de um cliente em uma de suas *portas de comunicação*.

Uma mesma máquina pode oferecer vários serviços, desde que em portas diferentes. Imagine que o endereço IP de uma máquina seja como o endereço de um prédio de escritórios. Localizando o prédio, você procura por um determinado serviço que é prestado por uma empresa. Pode haver várias empresas no prédio. Cada uma tem uma sala identificada por um número. O número da sala é análogo à porta de serviços de uma máquina. Para facilitar a vida dos clientes, várias portas, identificadas por um número, foram padronizadas, ou seja, em computadores diferentes, você geralmente encontra os mesmos serviços localizados em portas com os mesmos números.

Para ter acesso a um serviço é preciso ter um cliente apropriado que saiba conversar na língua (protocolo) do servidor (programa que oferece o serviço na porta buscada pelo cliente). Como as portas são padronizadas, um cliente muitas vezes só precisa saber o nome ou endereço IP da máquina que tem determinado serviço, pois o número da porta ele supõe que seja o número padrão.

Se você usa o *Internet Explorer* para ter acesso ao site do IBPINET ou o *Outlook Express* para ler seu e-mail no IBPINET, é possível que você esteja se conectando à mesma máquina. O *Outlook Express* se conectará à porta 110 para verificar suas mensagens. Na hora de enviar, utilizará os serviços da porta 25. O *Internet Explorer* buscará a *home page* do IBPINET na porta 80, mas se você decidir fazer compras e utilizar o servidor seguro do IBPINET, se o browser utilizará a porta 443.

1.2.5. *A Web entre os serviços da Internet*

O serviço de emulação de terminal remoto - *Telnet*, requer que o usuário conheça o sistema remoto, tenha permissão de acesso e saiba utilizá-lo. O acesso é orientado a caracter e pode ser feito em MS-DOS. No início da popularização da Internet, vários serviços eram oferecidos apenas via *Telnet*, que funcionava como uma espécie de cliente universal. O usuário, ligado à Internet, poderia pesquisar o banco de dados da Nasa, bater papo com usuários

remotos e pesquisar repositórios de informações pelo mundo afora. A World Wide Web surgiu inicialmente como mais um desses serviços, que poderia ser utilizado através de um cliente Telnet apontando para o endereço `info.cern.ch`. O serviço ainda existe, por razões históricas. É possível acessá-lo via linha de comando no DOS ou Unix usando:

```
telnet info.cern.ch
```

É um acesso orientado a caracter e na época concorria com outros serviços de informações mais populares como o *WAIS*, o *Gopher* e o *Archie*, que também tentavam organizar as informações da Internet.

O acesso direto ao servidor, usando um cliente conectado à sua porta de comunicação é sempre mais rápido e eficiente que o acesso via Telnet. Os primeiros clientes Web eram orientados a caracter (não exibiam fontes nem imagens) e rodavam apenas em ambientes Unix. Com o surgimento do *X-Mosaic*, o primeiro browser gráfico rodando em ambiente *X-Window* (Unix), a Web começou a se tornar popular (pelo menos entre os acadêmicos). O crescimento explosivo da Internet aconteceu quando finalmente o Mosaic passou a ser acessível ao usuário comum não-acadêmico através do *Win-Mosaic* e posteriormente do *Mac-Mosaic*. O Mosaic, desenvolvido por Marc Andreessen quando estagiava no NCSA (*National Center for Supercomputing Applications*) abriu caminho para vários outros clientes Web, hoje chamados de browsers, e que passaram a ser o primeiro (e às vezes único) contato de muitos usuários com a Internet. O próprio Andreessen, ao deixar o NCSA, fundou a *Netscape Communications*, que dominou a Web por vários anos.

Os browsers de hoje não são mais apenas clientes Web. Eles lêem páginas locais, enviam e-mail, permitem que o usuário leia grupos de notícias, e-mail, execute aplicações locais, acesse aplicações remotas e diversos outros serviços da Internet. O browser moderno é um cliente universal para toda a Internet, embora esse acesso ocorra através da World Wide Web.

1.3. A plataforma Web

A World Wide Web é um serviço TCP/IP baseado no *protocolo de nível de aplicação HTTP* (*HyperText Transfer Protocol*) – *Protocolo de Transferência de Hipertexto*. A plataforma Web é o meio virtual formado pelos servidores HTTP (servidores Web que mantêm sites), clientes HTTP (browsers) e protocolo HTTP (a língua comum entre o cliente e o servidor).

1.3.1. Hipertexto

Hipertexto é uma forma não linear de publicação de informações onde palavras que aparecem no decorrer do texto podem levar a outras seções de um documento, outros documentos ou até outros sistemas de informação, fugindo da estrutura linear original de um texto simples. O hipertexto baseia-se em ligações entre dois pontos chamados de *âncoras*. As ligações entre as âncoras são chamadas de *vínculos* (*links*). Vínculos de hipertexto são

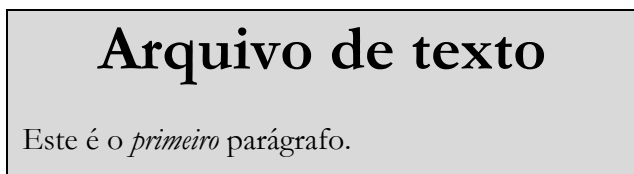
implementados em textos publicados na Web usando uma linguagem declarativa chamada *HTML - HyperText Markup Language*.

1.3.2. HTML

HTML é usada para *marcar* um arquivo de *texto simples* (texto simples é texto sem formatação alguma, visualizável em *qualquer* editor de textos). Se um arquivo de texto simples receber uma extensão de nome de arquivo “.html” ou “.htm”, um navegador como o Internet Explorer irá tentar interpretá-lo como HTML. Dentro do texto, pode-se definir descritores (ou comandos HTML) entre os símbolos “<” e “>”:

```
<h1>Arquivo de texto</h1>
<p>Este é o <i>primeiro</i> parágrafo.</p>
```

Os descritores só serão visíveis quando o arquivo for visualizado em um editor de textos (como o Bloco de Notas do Windows). Ao ser visualizado em um programa capaz de entender HTML, apenas o texto aparece, com uma aparência determinada pelos descritores:



O texto com marcadores é chamado *código-fonte HTML*. O código-fonte é usado para produzir a página visualizada o browser que é chamada de *página HTML* ou *página Web*.

O browser, por ser capaz de exibir diversos tipos de informação, depende totalmente da *extensão do arquivo* para saber o que fazer com ele. Se a extensão “.htm” ou “.html” não estiver presente ou se o arquivo tiver a extensão “.txt”, o browser exibirá o código-fonte.

Além da formatação da página, o HTML é responsável também pela inclusão de imagens e definição dos links que permitem a navegação em hipertexto.

1.3.3. Servidor HTTP

O serviço HTTP funciona de forma semelhante ao serviço *FTP - File Transfer Protocol* (protocolo de comunicação usado na Web para operações de transferência de arquivos). Ambos oferecem aos seus clientes um *sistema de arquivos virtual* onde podem localizar *recursos* (arquivos, programas, etc.) e transferi-los de um computador para outro. O sistema virtual pode ter uma *hierarquia* própria e totalmente diferente do *sistema de arquivos real do computador*, ao qual está vinculado. Geralmente um servidor tem acesso a uma área restrita da máquina e só permite a visualização dos arquivos lá contidos. O sistema de arquivos virtual usa uma notação diferente daquela usada pelo sistema real. Por exemplo, considere o seguinte sistema de diretórios no Windows:

```
C:\
```

```

C:\Windows
C:\Documentos
C:\Documentos\Web\
C:\Documentos\Web\Imagens
C:\Documentos\Web\Videos

```

Suponha que um servidor HTTP foi instalado nessa máquina. Na instalação, ele é configurado para administrar um sistema de diretórios a partir de um certo diretório. Suponha que esse diretório é C:\Documentos\Web\. Para o servidor, isto é seu diretório raiz. No sistema de diretórios virtual, o diretório raiz de um servidor é chamado de / (barra). O sistema de arquivos virtual (a parte que um browser poderá ter acesso) é:

```

/                (C:\Documentos\Web\)
/Imagens         (C:\Documentos\Web\Imagens)
/Videos          (C:\Documentos\Web\Videos)

```

Um browser jamais terá acesso ao diretório Windows, por exemplo. A principal função de um servidor Web é, portanto, *administrar* um sistema de arquivos e diretórios virtual e *atender à requisições dos clientes* HTTP (os browsers), que, na maior parte das vezes, enviam comandos HTTP pedindo que o servidor devolva um ou mais arquivos localizados nesses diretórios. Os pedidos são feitos através de uma sintaxe especial chamada de URI.

1.3.4. URIs (URLs ¹)

Todas as comunicações na plataforma Web utilizam uma sintaxe de endereçamento chamada *URI - Uniform Resource Identifier* - para localizar os recursos que são transferidos. O serviço HTTP depende da URI que é usada para localizar *qualquer coisa* na Internet. Contém duas informações essenciais: 1) COMO transferir o objeto (o protocolo); 2) ONDE encontrá-lo (o endereço da máquina e o caminho virtual). URIs tipicamente são constituídas de três partes:

- mecanismo (protocolo) usado para ter acesso aos recursos (geralmente HTTP)
- nome da máquina (precedido de //) onde o serviço remoto é oferecido (e a porta, se o serviço não estiver em uma porta padrão) ou outro nome através do qual o serviço possa ser localizado (sem //).
- nome do recurso (arquivo, programa) na forma de um caminho (no sistema de arquivos virtual do servidor) onde se possa encontrá-lo dentro da máquina.

Sintaxe típica:

```
protocolo://maquina:porta/caminho/recurso
```

¹ URIs também são frequentemente chamadas de URLs (Uniform Resource Locators). A URL é um tipo particular de URI mas, para a nossa discussão, essa distinção é irrelevante. A documentação HTML (especificação) sempre refere-se à essa sintaxe como URI.

As URIs mais comuns são os endereços da Web, que utilizam o mecanismo HTTP para realizar a transferência de dados:

`http://www.maquina.com.br/caminho/para/minha/página/texto.html`

Veja algumas outras URLs:

- `ftp://usuario:senha@maquina.com/pub/arquivo.doc`
Acesso a servidor FTP que exige usuário e senha para fazer download de arquivo.doc
- `nnntp://news.com.br/comp.lang.java`
Acesso a servidor de newsgroups para ler o grupo comp.lang.java
- `news:comp.lang.java`
Acesso ao grupo comp.lang.java através de servidor default (definido localmente)
- `http://www.ibpinet.net/`
Acesso à página default disponível no diretório raiz do servidor Web de www.ibpinet.net
- `http://www.algumlugar.com:8081/textos/`
Acesso à página default disponível no diretório textos do servidor Web que roda na porta 8081 da máquina www.algumlugar.net
- `http://www.busca.com/progbusca.exe?opcoes=abc&pesquisa=dracula`
Passagem de parâmetros de pesquisa para programa de busca progbusca.exe que terá sua execução iniciada pelo servidor HTTP que roda na porta 80 (default) de www.busca.com.
- `http://www.ibpinet.net/helder/dante/pt/inferno/notas_4.html#cesar`
Acesso à uma seção da página HTML notas_4.html identificada como “cesar”, localizada no subdiretório virtual /helder/dante/pt/inferno/ do servidor Web de www.ibpinet.net.
- `mailto:helder@ibpinet.net`
Acesso à janela de envio de e-mail do cliente de correio eletrônico local.

1.3.5. Browser

O browser é um programa que serve de interface universal a todos os serviços que podem ser oferecidos via Web. É para a plataforma Web o que o sistema operacional (Windows, Linux, Mac) é para o computador. A principal função de um browser é *ler* e exibir o conteúdo de uma página Web. A maior parte dos browsers também é capaz de exibir vários outros tipos de informação como diversos formatos de imagens, vídeos, executar sons e rodar programas.

Um browser geralmente é usado como *cliente HTTP* – aplicação de rede que envia requisições a um servidor HTTP e recebe os dados (uma página HTML, uma imagem, um programa) para exibição, execução ou *download*. Browsers também podem ser usados *off-line* como aplicação local do sistema operacional para navegar em sistemas de hipertexto

construídos com arquivos HTML (sem precisar de servidor HTTP). Nesse caso, não se comportam como clientes HTTP (já que não estão realizando operações em rede) mas apenas como *visualizadores de mídia interativa* capazes de visualizar HTML, imagens, sons, programas, etc.

Como os browsers precisam interpretar vários tipos de código (código de imagens GIF, JPEG, código de programas Java e Flash, códigos de texto HTML ou texto simples) é preciso que ele saiba identificar os dados que recebe do servidor. Isto não é a mesma coisa que identificar um arquivo carregado do disco local, onde ele pode identificar o tipo através da extensão. Quando os dados chegam através da rede, a extensão não significa nada. O servidor precisa informar ao browser o que ele está enviando. Na Web, isto é feito através de uma sintaxe padrão para definir tipos chamada *MIME - Multipart Internet Mail Extensions*.

1.3.6. Tipos MIME

MIME é uma sintaxe universal para identificar tipos de dados originalmente utilizada para permitir o envio de arquivos anexados via e-mail. O servidor Web possui, internamente, tabelas que relacionam os tipos de dados (na sintaxe MIME) com a extensão dos arquivos por ele gerenciados. Quando ele envia um conjunto de bytes para o browser, envia antes um *cabeçalho* (semelhante ao cabeçalho de e-mail) informando o *número de bytes* enviados e o *tipo MIME* dos dados para que o browser saiba o que fazer com a informação. A sintaxe MIME tem a seguinte forma:

tipo/subtipo

O *tipo* classifica um conjunto de bytes como imagens, textos, vídeos, programas (aplicações), etc. O *subtipo* informa características particulares de cada tipo. Não basta saber que o arquivo é uma imagem, é preciso saber qual o *formato*, pois os códigos usados para produzir imagens de mesma aparência gráfica podem diferir bastante entre si. Tanto no servidor como no browser há tabelas que relacionam extensões de arquivo a tipos MIME:

image/jpeg	.jpe, .jpg, .jpeg
image/png	.png
image/gif	.gif
text/html	.html, .htm, .jsp, .asp, .shtml
text/plain	.txt
x-application/java	.class

1.4. Tecnologias de Apresentação

As tecnologias utilizadas na plataforma Web podem ser classificadas de acordo com sua finalidade em *tecnologias de apresentação* e *tecnologias interativas*. As tecnologias de apresentação são aquelas que se destinam unicamente à formatação e estruturação das páginas Web. Podem ser usadas também para construir a interface de aplicações Web no browser. Os principais padrões

em uso atualmente são HTML, CSS, XML e XSL. As tecnologias interativas são as que permitem o desenvolvimento de aplicações e páginas com alto nível de interatividade com o usuário. Em geral consistem da combinação de uma linguagem de programação com uma arquitetura ou modelo que possibilita a sua integração com uma página HTML ou servidor HTTP. Podem, portanto, executar do lado do servidor (como CGI, ASP, ADO, Servlets, ISAPI, JSP, PHP, Cold Fusion e LiveWire) ou do lado do cliente (como JavaScript, DHTML, Java Applets, ActiveX e VBScript).

1.4.1. HTML

HTML - *HyperText Markup Language* é a linguagem universal da Web. É através dela que a informação disponível nas páginas da WWW pode ser acessada por máquinas de arquiteturas e sistemas operacionais diferentes. Não é uma linguagem de programação com a qual se possa construir algoritmos, mas uma linguagem declarativa que serve para organizar informações em um arquivo de textos que será visualizado em um browser. Define uma coleção de elementos para marcação (definição de estrutura) de texto. Se você, no passado, já usou um editor de textos como *WordStar* ou *Carta Certa*, deverá se “sentir em casa” com HTML. Como foi mencionado anteriormente, um *arquivo HTML* é um arquivo de *texto simples* recheado de marcadores que se destacam do texto pelos caracteres especiais "<" e ">".

Existem várias linguagens para formatação de textos. Qualquer texto que aparece formatado (com fontes, cores, tamanhos) em um computador tem uma linguagem de formatação por trás. A maioria são linguagens proprietárias que só funcionam em softwares específicos (textos em formato *Word*, por exemplo). Existem alguns formatos, porém, que se tornaram padrões, servindo basicamente para realizar conversões entre os formatos proprietários (SGML e RTF, PostScript, TeX, PDF). HTML é um formato *público* (não pertence a um fabricante específico), e é *leve* (não produz arquivos enormes como o *Word*) sendo por essas e outras razões adequado à difusão de informações que serão visualizadas em máquinas diferentes.

Com HTML é possível publicar documentos estruturados *on-line*, recuperar informações através de vínculos de hipertexto, projetar uma interface interativa com formulários para acesso a serviços remotos como buscas e comércio eletrônico, e incluir imagens, vídeos, sons, animações e outras aplicações interativas dentro de documentos visíveis no browser.

1.4.2. Histórico do HTML

HTML foi desenvolvida originalmente por Tim Berners-Lee no CERN - Laboratório Europeu de Física de Partículas. Sua popularidade cresceu junto com a popularização da Web, através do *NCSA Mosaic*. Devido ao surgimento de vários browsers que utilizavam HTML para navegar no sistema de informações proporcionado pela Web, grupos de trabalho foram formados com a intenção de padronizar especificações para o HTML.

HTML 2.0, concluída em 1995, foi a primeira versão recomendada pelo IETF - *Internet Engineering Task Force* e se tornou um padrão da Internet. HTML 2.0 era uma linguagem simples que dizia como um browser deveria estruturar uma página, mas não como os títulos, parágrafos e listas deveriam aparecer graficamente. Durante o desenvolvimento do HTML 3.0, a Web estava em franca expansão e os esforços de padronização não puderam acompanhar as tendências do mercado, que exigiam maiores recursos de apresentação gráfica ao HTML 2.0. O HTML 3.0 acabou não sendo aprovado e anos depois, aprovou-se uma recomendação chamada HTML 3.2 em 1997, que introduzia recursos de apresentação gráfica no HTML.

Infelizmente a maioria dos recursos gráficos do HTML 3.2 foram incorporações de extensões proprietárias da *Netscape* e *Microsoft*, criadas sem levar em conta a filosofia do HTML de garantir a compatibilidade da linguagem em plataformas diferentes. Isto acabou atrasando o desenvolvimento de ferramentas de desenvolvimento eficientes, pois era impossível validar HTML para plataformas que não suportavam certos recursos gráficos mais sofisticados.

HTML foi desenvolvida originalmente para que *qualquer dispositivo* pudesse ter acesso à informação da Web. Isto inclui PCs com monitores gráficos de diversas resoluções, terminais orientados a caracter, telefones celulares, dispositivos geradores de voz, etc. HTML 3.2 tinha elementos que prejudicavam essa meta. Finalmente, depois de muita discussão, as empresas entraram em um acordo e desenvolveram o HTML 4.0, que estende o HTML com mais recursos visando um acesso mais universal à informação da Web, como recursos de acessibilidade à pessoas com deficiências, suporte a convenções internacionais (outras línguas, outros alfabetos), separação da estrutura, conteúdo e apresentação, recursos interativos do lado do cliente e otimização em tabelas e formulários.

1.4.3. HTML 4 e CSS

Vários elementos do HTML 3.2 foram considerados *deprecados* (candidatos a se tornarem obsoletos) pelo HTML 4. São todos elementos que permitem definir cores, fontes, alinhamento, imagens de fundo e outras características da apresentação da página que dependem da plataforma onde a informação é visualizada. HTML nunca realizou bem o trabalho de formatação gráfica de uma página. Foi criada inicialmente para apenas dar estrutura a um conteúdo. Nunca previu formas de posicionar imagens e texto de forma absoluta em uma página e as soluções desenvolvidas pelos Web designers, por não serem soluções previstas na especificação, têm causado problemas de acesso em vários sites e impedido o acesso de dispositivos mais limitados como a *WebTV* e *WebPhone* de terem acesso total à Web. A solução do HTML 4.0 foi separar a *estrutura* da *apresentação*, deixando que a linguagem HTML voltasse às suas origens (nos tempos do HTML 2) para definir apenas a função do texto marcado (o *que* é título, parágrafo, etc.) Uma outra linguagem foi criada para se preocupar com a aparência (*como* o título e parágrafo serão exibidos na tela). A principal linguagem usada para esse fim é CSS - *Cascading Style Sheets*, que permite a criação de *folhas de estilo* aplicáveis a várias páginas de um site. Se um dispositivo limitado não consegue exibir os estilos definidos no CSS, ele pelo

menos consegue entender a estrutura do texto e imagens de forma que mesmo usuários com menos recursos podem ter acesso à informação.

Escrever HTML não é difícil, apenas toma tempo, por isso usar um editor apropriado pode tornar o processo de criação de páginas mais produtivo. Um arquivo de texto simples com descritores HTML, quando carregado em um browser, tem os descritores interpretados e as suas informações formatadas na tela de acordo com a estrutura prevista pelos marcadores e uma folha de estilos, geralmente definida pelo próprio browser. A folha de estilos pode ser definida pelo programador usando uma linguagem como CSS e vinculada à página para mudar sua aparência.

1.4.4. XML e XSL

XML - *eXtensible Markup Language* e XSL - *eXtensible Style Language* são as novas criações do W3C - *World Wide Web Consortium* (consórcio das empresas que desenvolvem os padrões para a WWW). Não pretendem substituir o HTML mas, em vez disso, oferecer meios de estender e ampliar as possibilidades da Web. XML é uma especificação ou *meta-linguagem* que define uma sintaxe que pode ser usada para criar novas linguagens semelhantes a HTML. A própria linguagem HTML pode ser vista como um tipo especial de XML. Com XML você pode criar sua própria linguagem de marcação “MinhaML”, definir seus próprios marcadores e esquemas para aplicações específicas, por exemplo, poderia conter algo como:

```
<compra id="xyz"><data>26/12/1999</data>...</compra> ...
```

Depois você pode usar CSS ou XSL para definir a aparência dos seus marcadores em um browser XML. Para que um browser XML seja capaz de compreender a linguagem que você criou, é preciso definir um *dicionário* e uma *gramática* (usando as regras da especificação XML) para ela. A gramática para a análise da sua linguagem deve ficar armazenada em um documento chamado DTD - *Document Type Definition*. Carregando o DTD, o browser XML aprenderá a nova linguagem e será capaz de formatar a informação que você estruturou com seus marcadores, e apresentar as informações na tela de acordo com as regras de estilo definidas no CSS ou em um XSL criado por você.

As tecnologias XML e XSL não serão abordadas neste curso mas você pode obter maiores informações sobre as duas tecnologias, além de ter acesso a guias de referência e tutoriais através do site do World Wide Web Consortium (W3C) em <http://www.w3.org/XML/>.

1.5. Ferramentas de desenvolvimento HTML

Podemos criar uma página Web simplesmente usando um editor de textos (como o bloco de notas do Windows) e um browser para visualização. Embora esta seja uma boa forma de aprender, é pouco produtiva para desenvolver sites complexos que usam tabelas, frames e design sofisticado. Alguns editores são bastante visuais e mostram todo o processo de criação

do site, fazem busca e substituição em todo o site, utilizam *templates*, permitem edição direta do código e fazem a previsão sem a necessidade de um browser. É o caso do *Microsoft FrontPage*, do *Macromedia DreamWeaver*, do *Corel WebMaster*, e do *Adobe GoLive*. Outros, simplesmente servem para economizar batidas de teclas e evitar erros de sintaxe na hora de escrever os descritores HTML. Editores de código HTML como o *Hot Dog*, *Hot Metal*, *HomeSite*, *BareBones*, etc. realizam esta função. Nas seções a seguir, discutiremos um pouco de cada um.

1.5.1. Editores gráficos (WYSIWYG)

Os editores mais fáceis de usar, que dispensam totalmente o uso de HTML são os editores WYSIWYG. Esse nome esquisito é uma sigla bastante usada nos primeiros tempos da editoração eletrônica para caracterizar programas que representavam na tela do computador uma página da forma como seria impressa. Naquele tempo, era comum a existência de processadores de texto ou formatadores de texto que usavam comandos para descrever como o texto iria aparecer na hora da impressão. Para se ter uma idéia do resultado final, era necessário ou imprimir ou rodar um programa a parte que fizesse um *preview* da página criada. Com o advento da editoração eletrônica, surgiram programas como o *Aldus PageMaker* no *Macintosh* e *Ventura Publisher* no PC que mostravam na tela a própria página, e não um monte de códigos de controle misturados ao texto. WYSIWYG quer dizer *What You See Is What You Get*, ou, “o que você vê é o que você obtém”, sugerindo que o que o autor visualiza na tela é uma representação bastante fiel do resultado final que obterá na impressora.

WYSIWYG na Web é menos fiel que na impressão. Para trabalhos impressos, tem-se uma representação da página em uma determinada impressora, com determinadas fontes, cores, etc. Na Web é impossível saber se a pessoa que vai ver a sua página tem as mesmas fontes, cores, versões de browser que aquela onde você fez o seu teste “WYSIWYG”. Portanto, os editores gráficos de páginas Web são WYSISWIG ou What You See is *Sometimes* What You Get.

A multiplicidade de browsers e plataformas que existem na Internet exige do Web designer mais que saber fazer páginas que ficam boas em uma única plataforma e browser. É necessário testar o site em sistemas diferentes, levando em conta o público-alvo e muitas vezes é necessário recorrer à codificação HTML para resolver algum problema não previsto pelos editores.

1.5.2. Editores de texto

Pode-se usar qualquer editor de texto que tenha a capacidade de salvar um arquivo de texto “puro”, em formato ISO-Latin-1 ou ASCII para criar páginas HTML. Se o seu editor só conseguir salvar ASCII 7-bit, será mais difícil o uso de acentos (você terá que usar códigos especiais). Alguns exemplos de editores que podem ser usados são o *Bloco de Notas* do Windows, o *WordPad* (desde que se salve a página como “texto”), o *EditPad*, o *WinEdit* e outros editores shareware/freeware populares disponíveis na Internet.

1.5.3. Editores de HTML

Editores de HTML são como os editores de texto mas possuem uma série de atalhos para agilizar a entrada de código HTML. Os atalhos também evitam que se cometam erros de sintaxe. Os editores HTML mais sofisticados possuem um sistema de *preview* embutido, permitem a definição de gabaritos a serem aplicados a um conjunto de páginas, suportam macros, busca e substituição em diversos arquivos e filtros diversos.

A maioria dos editores HTML estão disponíveis na Internet pelo sistema de *freeware* ou *shareware*. O site das duas vacas, ou TUCOWS (<http://www.tucows.com>) possui uma lista crescente destes programas. Você também encontra editores em <http://www.shareware.com>. Entre os comerciais, um dos mais populares é o *Allaire HomeSite*.

1.5.4. Browsers e servidores

Qualquer web designer que queira desenvolver páginas seriamente hoje em dia deve ter as versões dos browsers mais populares entre o público. Não basta testar somente na última versão. Geralmente a grande massa de usuários que navega na Internet não possui a última versão do seu browser.

Os browsers são a janela do usuário para a Web. Dependendo da versão ou do tipo de browser que o seu visitante está usando, e qual o seu computador, a sua página pode aparecer para ele como uma experiência agradável que o fará voltar muitas e muitas vezes; ou como um pesadelo horrível, que ele jamais esquecerá. É trabalho do Web designer resolver ao máximo esses problemas de compatibilidade de forma a garantir que pelo menos a informação essencial seja acessível (isto inclui a identidade visual e a navegação do site). É essencial testar as páginas em browsers e plataformas diferentes antes de publicá-las.

Os browsers mais populares hoje continuam sendo o *Netscape Communicator* e *Microsoft Internet Explorer*. As plataformas de navegação mais populares ainda são PC e Mac. Mas o que é usado por todo o mundo pode não ser o mais importante no seu caso: se seu *público-alvo* é formado em grande parte por usuários de máquinas Sun é essencial testar suas páginas nessas plataformas também.

Há ainda uma outra questão. Mesmo que você saiba que 99% do seu público usa browsers Netscape ou Microsoft, é importante saber as *versões* que eles usam. A grande maioria dos usuários *não possui* as últimas versões dos seus browsers (os provedores continuam distribuindo versões antigas). Há muitos recursos novos nas novas versões do *Communicator* e do *Internet Explorer* que não são suportados nas versões anteriores. Se você usá-los sem prestar atenção ao efeito que causam em versões antigas, uma grande parte do público que visitar suas páginas poderão nunca mais voltar.

1.5.5. *Por que aprender HTML?*

Se é possível fazer todo um site, com *scripts* interativos, imagens, animações, Java, etc. sem escrever uma linha sequer de HTML, por que perder tempo aprendendo mais hieroglifos que não vão servir para nada?

A resposta à pergunta do parágrafo anterior depende do que você pretende fazer e do controle que você pretende ter sobre a sua obra. Se o seu objetivo é fazer páginas práticas e rápidas, como uma *home page* pessoal ou para um sistema Intranet simples, pode realizar todo o trabalho sem sequer tomar conhecimento do HTML, usando as ferramentas que existem por aí. Mas se você deseja fazer coisas mais sofisticadas, como ter mais controle sobre o *design* da sua página, criar páginas que interajam com outros programas, como servidores de bancos de dados, e usar recursos novos como *Dynamic HTML*, *JavaScript*, canais e folhas de estilo, você provavelmente não vai escapar de ter que editar sua página pelo menos uma ou duas vezes para fazer alguns ajustes, ou para analisar o código para poder criar rotinas JavaScript, DHTML, etc.

A notícia boa é que HTML é fácil. Não é linguagem de programação, não tem recursão, ponteiros ou nada do tipo. É somente marcação de texto. É trivial para aqueles que usam computadores desde o tempo do MS-DOS e WordStar – programas bem mais complexos que o HTML.