

*Escola de  
Verão*

*Jan. 98*



*Graduação e Pós-Graduação*

# Introdução à Internet

Prof. Nilton Alves Jr.



Denise Coutinho de A. Costa  
denise@cat.cbpf.br

Eduardo Fahr Pessoa  
pessoa@cat.cbpf.br

Fernanda Santoro Jannuzzi  
fernanda@cat.cbpf.br

Helio Sergio Nigri  
hnigri@cat.cbpf.br

João Marques Ferreira Jr.  
joaom@cat.cbpf.br

Leonardo Ferreira Carneiro  
leo@cat.cbpf.br

Nilton Alves Jr.  
naj@cat.cbpf.br

## Resumo

Esta Nota Técnica é uma revisão das Notas de Aula do curso de mesmo título da Escola de Verão 98 – CBPF. O curso e agora este trabalho, destinam-se a alunos e profissionais da área técnico/científica que desejam adquirir/aumentar seus conhecimentos relativos à Internet. Esta Nota Técnica também se propõe a ser material de consulta para trabalhos outros que envolvam históricos e estatísticas sobre a Internet e a Internet Brasil.

Este trabalho tem como objetivo principal abordar os principais aspectos, inclusive técnicos, da infra-estrutura da “Era da Informação”, para que sejam utilizados de maneira otimizada os recursos disponibilizados no ambiente Web.

Os objetivos secundários são: i) mostrar o desenvolvimento das redes, suas origens e sua evolução; ii) promover discussões de maneira que o senso crítico sobre as informações seja ampliado; iii) apresentar os principais protocolos, suas características técnicas, limitações, vantagens e eventuais configurações; iv) trabalhar com os principais serviços disponíveis na rede, utilizando as características/opções mais vantajosas dos softwares mais divulgados; v) transmitir a importância da utilização da rede, principalmente na área de pesquisa e desenvolvimento; vi) criar páginas Web passivas utilizando HTML e ativas (ou pseudo-ativas) utilizando programas em C e Java Script e vii) utilizar o ambiente Web para aplicações científicas e videoconferências



# Índice

<b>1. HISTÓRICO DA INTERNET</b>	<b>5</b>
1.1. INTERNET NO BRASIL	11
1.1.1. Domínios Registrados	13
1.1.2. Hosts por Domínio	15
1.2. ATO NORMATIVO PARA A ATRIBUIÇÃO DE DOMÍNIO NA INTERNET NO BRASIL	17
<b>2. ENTENDENDO A INTERNET</b>	<b>19</b>
2.1. ADMINISTRAÇÃO DA INTERNET	20
2.2. CONECTIVIDADE NA INTERNET	20
<b>3. FUNCIONAMENTO DA INTERNET</b>	<b>22</b>
3.1. PROTOCOLO TCP/IP	22
3.1.1. O Modelo OSI	22
3.1.2. As Camadas do TCP/IP	23
3.1.3. Relações entre o modelo OSI e TCP/IP	24
3.1.4. Subredes	25
3.1.5. Redes na Internet	25
3.1.6. Roteadores e Bridges	26
3.1.7. Componentes de uma LAN	26
3.2. ENDEREÇAMENTO NA INTERNET	27
<b>4. APLICAÇÕES</b>	<b>29</b>
4.1. FILE TRANSFER PROTOCOL - FTP	29
4.1.1. Comandos do FTP	30
4.2. TELNET	31
4.2.1. Comandos do Telnet	31
4.3. MAIL	32
<b>5. AMBIENTE WEB</b>	<b>33</b>
5.1. NAVEGADORES	34
5.2. PROCURA DE INFORMAÇÃO	35
<b>6. CRIAÇÃO DE PÁGINAS WEB</b>	<b>37</b>
6.1. HYPERTEXT MARKUP LANGUAGE (HTML)	37
6.1.1. Básico	37
6.1.2. Títulos	38
6.1.3. Formatação do texto	38
6.1.4. Imagens	39
6.1.5. Listas	39
6.1.6. Links	39
6.1.7. Tabelas	40
6.1.8. Formulários	41
6.2. CGI EM C	42
<b>7. INTRODUÇÃO À LINGUAGEM JAVASCRIPT</b>	<b>44</b>
7.1. AONDE INSERIR O SCRIPT?	44
7.2. CONSIDERAÇÕES INICIAIS	46
7.2.1 Variáveis	46
7.2.2. Operadores	46
7.2.3. Estruturas de controle	47
7.2.4. Comandos	48
7.3. OBJETOS	49
7.3.1. Objetos HTML	49
7.3.2. Objetos internos.	51
7.3.3. Objetos do navegador	53
7.4. FUNÇÕES	54
7.4.1. Funções internas	55
7.5. EVENT HANDLERS	55

---

7.6 EXEMPLO DE APLICAÇÃO EM JAVASCRIPT	56
<b>8. VIDEOCONFERÊNCIA</b>	<b>58</b>
8.1. PRINCÍPIOS BÁSICOS	59
8.2. SOFTWARES DE VIDEOCONFERÊNCIA	60
8.2.1 <i>Session Directory - SDR</i>	60
8.2.2 <i>Visual Audio Tool - VAT</i>	61
8.2.3 <i>Video Conferencing Tool - VIC</i>	61
8.2.4 <i>Whiteboard - WB</i>	61
<b>9. CONCLUSÃO</b>	<b>62</b>
<b>REFERÊNCIAS</b>	<b>63</b>
<b>LEITURAS ALTERNATIVAS</b>	<b>64</b>

# INTRODUÇÃO À INTERNET

## 1. Histórico da Internet

Neste primeiro capítulo, coletamos informações relativas ao histórico da Internet, em diversos *sites* relacionados nas Referências [1], [2] e [3]. Tivemos a intenção de fazer uma grande compilação de informações e dados visando facilitar qualquer tipo de trabalho de pesquisa neste assunto. Em seguida mostramos a evolução da Internet no Brasil, utilizando dados estatísticos representados em tabelas e gráficos.

### **Anos 40 (Pós-Guerra)**

- Notáveis cientistas perceberam que estávamos entrando na era da informação. Sonhavam com um banco de dados universal acessível por qualquer pessoa. Um desses cientistas antecipou um sistema eletrônico de compartilhamento de informação. O sistema era constituído de um “memex”, uma máquina conceitual que podia armazenar vastas quantidades de informação em que um usuário tinha a capacidade de criar *trails*, que eram *links* com textos e ilustrações relativas. Estes *trails* podiam então ser armazenados e usados no futuro.

### **1957**

- URSS estava temporariamente a frente na corrida do espaço quando lançou o Sputnik, o primeiro satélite artificial da Terra. Em resposta, os EUA criaram a ARPA (*Advanced Research Projects Agency*) dentro do Departamento de Defesa (DoD) para se estabelecer na liderança em ciência e tecnologia militar.

### **1962**

- Rand Paul Baran, de uma agência de governo, foi contratado pela força aérea dos EUA para fazer um estudo de como poderia manter o comando e o controle de seus mísseis e aviões de bombardeio depois de um ataque nuclear. Isto era para ser uma rede de pesquisa nuclear descentralizada que sobrevivesse à uma catástrofe nuclear de maneira que se qualquer cidade dos EUA fosse atacada, os militares teriam ainda o controle de suas armas nucleares para um possível contra-ataque.

### **1965**

- A ARPA patrocina o estudo em *Cooperative Network of Time-Sharing Computers*. A motivação original foi projeto de um sistema de comunicação. O laboratório MIT e um computador de Santa Mônica são diretamente ligados sem *packet switches*.

### **1967**

- A ARPANET foi criada e os computadores no Departamento de Defesa foram ligados transversalmente pelo país.

### **1969**

- Foi construída uma rede física ligando 4 nós: UCLA, SRI (em Stanford), Universidade de Santa Bárbara e Universidade de Utah. Esta rede foi ligada via circuitos de 50Kbps.

### **1970**

- Os *hosts* da ARPANET começam a usar o *Network Control Protocol* (NCP).

**1971**

- É inventado um programa de *e-mail* para enviar mensagens através de uma rede distribuída.

**1972**

- A ARPA foi renomeada DARPA.
- A ARPANET continuava usando NCP para transferir dados. Isto permitia a comunicação entre os *hosts* da mesma rede.

**1973**

- Primeiras conexões internacionais da ARPANET.
- Definição e implementação dos protocolos TCP/IP (protocolos não proprietários) que permitiu a comunicação entre computadores com arquiteturas dispares.
- É delineada a idéia do padrão ETHERNET.

**1974**

- Era possível a comunicação de todos os computadores do mundo, mas faltavam ainda as infra-estruturas de rede e a compatibilização de muitos fabricantes para que seus computadores pudessem se comunicar com computadores de outros fabricantes.
- Primeiro uso do vocábulo INTERNET por Vint Cerf and Bob Kahn no artigo sobre *Transmission Control Protocol*.

**1976**

- A ETHERNET foi desenvolvida permitindo mover dados rapidamente através de cabo coaxial.
- UUCP foi desenvolvido em laboratório da AT&T e distribuído.
- O Departamento de Defesa começou a experimentar o TCP/IP e ficou decidido o seu uso na ARPANET.

**1979**

- Criada a USENET juntamente com os seus grupos de *News*, permitindo que as pessoas possam exprimir abertamente suas idéias e convicções, podendo discutir em grupo problemas técnicos ou científicos.

**1981**

- Surge uma rede cooperativa, BITNET, na cidade universitária de Nova York com conexão à Yale.
- A BITNET realiza as tarefas de *eletronic mail* e servidores do tipo *List Server* para distribuir informação, além de arquivos de transferência.
- A CSNET (*backbone* criado pela *National Science Foundation*) foi criada com o objetivo de fornecer serviços de rede (especialmente *e-mail*) aos cientistas de universidades que não possuíam acesso algum a ARPANET. Foi proposto um plano para uma inter-rede com conexão entre a CSNET e a ARPANET.
- A CSNET ficou conhecida como *Computer Science Network*.

**1982**

- O DCA e a ARPA estabelecem a *Transmission Control Protocol* (TCP) e *Internet Protocol* (IP) como *Protocol Suite*, conhecido como TCP/IP, para a ARPANET. Isto resulta em uma das primeiras definições da INTERNET como sendo um conjunto de redes conectadas.
- É criada a EUNET (Rede Européia de UNIX) para fornecer *e-mail* e serviços da USENET.
- Primeiras conexões entre Holanda, Dinamarca, Suécia e UK.
- Especificação do *External Gateway Protocol* (RFC827)

**1983**

- Foi criada a *Internet Activities Board*(IAB)

- Em primeiro de janeiro toda máquina conectada ao ARPANET, passou a usar o TCP/IP. Este núcleo do *Internet Protocol* substituiu o NCP inteiramente.
- A Universidade de Wisconsin criou o *Domain Name System* (DNS). Isto permitiu que pacotes fossem direcionados para um *domain name* que poderia ser relacionado pelo servidor de DNS a um endereço IP correspondente. Isto facilitou a comunicação entre *hosts* de diferentes redes.
- ARPANET ficou dividida em duas redes: ARPANET e MILNET que ficaram integrados ao *Defense Data Network*, criado no ano anterior. A MILNET era uma rede de computadores de instituições militares enquanto a ARPANET servia para dar suporte aos trabalhos de pesquisas avançadas.
- Estações de trabalho surgiram. Muitas com a versão Unix de Berkley, que incluíam software de rede IP.
- Estabelecido pela IBM a *European Academic Research Network* (EARN), muito parecida com a BITNET.

### 1984

- Ficou estabelecido a JUNET (*Japan Unix Network*), usando UUCP.
- Estabelecido a JANET (rede acadêmica associada) na UK usando o *Coloured Book Protocols*, previamente SECnet.
- Introduzido o *Moderated Newsgroups* na USENET.

### 1985

- A *National Science Foundation* (NSF) começou a desenvolver novas linhas as quais seriam finalizadas em 1988 e criou a NSFNET, interligando todos os supercomputadores dos maiores centros americanos de pesquisa.

### 1986

- As redes NSFNET e ARPANET se conectaram entre si passando a ser as duas espinhas dorsais (*backbone*) de uma nova rede que junto com os demais computadores ligados à elas assumiram o nome de Internet.
- *Mail Exchanger*(MX) não permite que o IP de *hosts* de redes tenham endereços de domínio.
- Foi criada a CSFNET, que estabeleceu cinco ótimos centros de computadores para fortalecer alto poder computacional para todos. Isto permite uma explosão de conexões, especialmente de Universidades.
- O 1º FREENET apareceu sobre o presságio da *Society for Public Access Computing* (SoPAC). Mais tarde o gerenciamento do mesmo é assumido pela *National Public Telecomputing Network* (NTCP) em 1989.
- O *Network News Transfer Protocol* (NNTP) é projetado para intensificar a performance da USENET News através do TCP/IP.
- BARNET (*Bay Area Regional Research Network*) foi estabelecida usando autos elos de velocidade. Ficou operacional em 1987.
- A IETF (*Internet Engineering Task Force*) foi criada para servir como um fórum de coordenação técnica dos contratantes do DARPA trabalhando no ARPANET, na *Defense Data Network* (DDN) e o sistema *Internet Core Gateway*.

### 1987

- A BITNET e a CSNET foram unidas para formar a *Corporation for Research and Educational Networking* (CREN), outro trabalho da *National Science Foundation*.
- A NSFNET passou a ser mantida com apoio das organizações IBM, MCI (empresa de telecomunicações) e Merit (instituição responsável pela rede de computadores de instituições educacionais de Michigan), que formaram uma associação conhecida como *Advanced Network and Services* (ANS).
- Elos de *e-mail* estabelecidos entre a Alemanha e a China usando protocolos do CSNET.
- Os EUA liberaram a rede para uso comercial.

## 1988

- Pouco depois da conclusão do *backbone* do NFSNET, o tráfego cresceu tão rapidamente que começaram imediatamente com projetos de atualizar a rede outra vez.
- O Merit e seus companheiros formaram uma corporação sem fins lucrativos chamada *Advanced Network Systems* (ANS) a qual conduzia pesquisa dentro da rede de alta velocidade. A NSF rapidamente adotou uma nova rede e por volta do final de 1991, todos os seus sites foram conectados por este novo *backbone*.
- O DARPA formou o *Computer Emergency Response Team* (CERT) em resposta as necessidades mostradas durante um incidente.
- DOD escolhe adotar o modelo OSI e avaliar o uso do protocolo TCP/IP . O *US Governemment OSI Profile* (GOSIP) define o conjunto de protocolos para serem suportados pelos produtos adquiridos pelo governo.
- Foi criada a *California Education and Reasearch Federation Network* (CERFNET)
- Foi desenvolvido o *Internet Relay Chat* (IRC).
- A Fidonet conecta-se a rede habilitando a troca de *e-mail* e *news*.
- Países se conectando a NSFNET: Canadá, Finlândia, França, Dinamarca, Noruega, Islândia, Suécia.

## 1989

- Os provedores de serviços europeus formaram o *Reseaux IP Europeens* (RIPE) para assegurar a coordenação técnica e administrativa necessária para permitir a operação da PAN-EUROPEEN IP NETWORK .
- Primeiros retransmissores entre um carregador de *e-mail* comercial e a Internet: MCI Mail através da *Corporation for the National Research Initiative* (CNRI) e a Compuserver através de Ohio State University.
- Países se conectando a NSFNET: Austrália, Alemanha, Israel, Itália, Japão, México, Holanda, Nova Zelândia, Porto Rico e UK.

## 1990

- O Departamento de Defesa dispensou a ARPANET que foi substituída pelo *backbone* da NSFNET. As linhas originais da ARPANET foram tiradas de linhas, isto é, o *backbone* foi desativado criando-se em seu lugar o *backbone* Defense Research Internet (DRI).
- O CERN, em Geneva, implementa o primeiro sistema de hipertexto para fornecer acesso eficiente de informação para os membros da Comunidade Internacional de Física de Altas Energias.
- O ARPANET deixa de existir.
- Foi criado o *Eletronic Fronhie Foundation* (EFF).
- O Archie, sistema de pesquisa na Internet, foi liberado pela Megill.
- Hytelnet foi liberada pela Universidade de Sakatchewan.
- O *World surge on-line* (world.std.com), tornando-se o 1º provedor comercial de acesso *dial-up* à Internet.
- O *ISO Development Enviroment* (ISODE) foi desenvolvido para prover um acesso de migração do DOD para OSI. O software ISODE permite que a aplicação OSI opere através do TCP/IP.
- 10 redes regionais assim como o *backbone* nacional canadense formaram a CA\*net com conexão direta à NSFNET.
- A primeira máquina operada remotamente através da Internet, a Internet Toaster, controlada via SNMP, faz sua estréia no Interop.
- Países conectados ao NSFNET: Argentina, Áustria, Bélgica, Brasil, Chile, Grécia, Índia, Irlanda, Coréia do Sul, Espanha, Suíça.

## 1991

- A CSNET foi interrompida tendo realizado seu prematuro e importante papel na provisão de serviço acadêmico de rede. Um característica chave do CREN é que seu custo operacional é inteiramente obtido através dos direitos pagos pelos membros de sua organização.

- O NSF estabeleceu uma nova rede chamada *The National Research Education Network* (MREN). O propósito desta rede é conduzir alta velocidade nas pesquisas de rede. Não era para ser usada como uma rede comercial nem era para ser usada para enviar um dos muitos dados que a Internet agora transfere.
- O *Wide Area Information Server* (WAIS) foi liberado pelo *Thinking Machines Corporation*.
- O GOPHER foi liberado pela Universidade de Minn.
- World Wide Web ou WWW foi liberado pelo CERN.
- Foi criada a *Pretty Good Privacy* (PGP).
- *US High Performance Computing Act* (Gore1) fundou o *National Research and Education Network* (NREN).
- Começo do serviço do JANET IP (JIPS) na qual sinalizou a mudança através do software de *Coloured Book* para TCP/IP dentro da rede acadêmica de UK.
- Países conectados ao NSFNET: Croácia, República Checa, Hong Kong, Hungria, Polônia, Portugal, Singapura, África do Sul, Taiwan, Tunísia.
- Na transição de 1991/1992 a ASNET passou a ser o *backbone* principal da Internet; nessa mesma época iniciou-se o desenvolvimento de um *backbone* europeu (EBONE), interligando alguns países da Europa à Internet.

## 1992

- A *Internet Society* (ISOC) é patenteadada.
- Primeiro MBONE *audio multicast* e video multicast.
- Foi reconstituída a *Internet Architecture Board* (IAB) e tornou-se parte da *Internet Society*.
- Verônica, uma ferramenta de pesquisa GOPHERSPACE, é fundada pela Universidade de Nevada.
- *World Bank* surge on-line.
- É criado o 1º ISP do Japão, *Internet Initiative Japan* (IIJ).
- É inventado o termo "*surfing the internet*".
- Países conectados ao NSFNET: Camarão, Chile, Equador, Estônia, Kuwait, Luxemburgo, Latvia, Malásia, Eslováquia, Eslovênia, Venezuela, Tailândia.
- Nos EUA surgiram as primeiras empresas provedoras de acesso comercial à Internet. Sua explosão veio a ocorrer com o surgimento da Web.

## 1993

- O INTERNIC foi criado pela NSF para fornecer serviços específicos da Internet: diretório e serviços de dados pela AT&T, serviços de registros pela Network Solutions Inc. e serviços de informação pela General Atomics/CERNet.
- A NCSA e a Universidade de Illinois desenvolveram uma interface gráfica para o WWW chamada "Mosaic for X". O Mosaic faz sucesso na Internet.
- Começa a ser transmitido a Internet Talk Radio.
- As Nações Unidas aparecem on-line.
- A mídia e o comércio realmente tomam conhecimento da Internet.
- A partir de 1993 a Internet deixou de ser uma instituição de natureza apenas acadêmica e passou a ser explorada comercialmente, tanto para a construção de novos *backbones* por empresas privadas (PSI, UUnet, Sprint, ...) como para o fornecimento de serviços diversos, abertura essa a nível mundial
- Países conectados ao NSFNET: Bulgária, Costa Rica, Egito, Fiji, Gana, Guam, Indonésia, Kazakhstan, Kenia, Liechtenstein, Peru, Romênia, Federação Russa, Turquia, Ucrânia, UAE, Ilhas Virgens.

## 1994

- Nenhuma grande mudança foi feita na parte física da rede. O que aconteceu de mais significativo foi o crescimento. Muitas novas redes foram ligadas ao *backbone* NSF. Centenas de milhares de novos *hosts* foram acrescentados na Internet durante este período.
- A Pizza Hut oferece seus serviços *on-line*.

- Primeiro *cyberbank*.
- É instalado na NSFNET o *backbone Asynchronous Transmission Mode* (ATM) de 145 Mbps.
- A ARPANET/Internet celebra 25 anos de aniversário.
- Comunidades começam a se ligar diretamente à Internet (Lexington e Cambridge, Mass., USA).
- *Shopping centers* chegam à Internet.
- Transmissão do 1º *Cyberstation*, RT-FM, pelo INTEROP em Las Vegas
- O *National Institute for Standards and Technology* (NIST) sugere que o GOSIP deveria incorporar o TCP/IP e deixar cair o requerimento “*OSI-only*”.
- *Trans-European Research and Education Network Association* (TERENA) é formado pela união de RARE e EARN com representantes de 38 países assim como CERN e ECMWF. O alvo do TERENA é promover e participar do desenvolvimento de uma alta qualidade de informação internacional e infra-estrutura de telecomunicação para o benefício da pesquisa e comunicação.
- Países conectados a NSFNET: Armênia, Argélia, Bermudas, Burkina Faso, China, Colômbia, Polónia Francesa, Jamaica, Líbano, Lituânia, Macal, Marrocos, Nova Caledônia, Nicarágua, Nigéria, Panamá, Filipinas, Senegal, Lanka Dosri, Swaziland, Uruguai, Uzbekistan.

## 1995

- A *National Science Foundation* anunciou que em 30 de Abril deste ano não poderia mais permitir nenhum longo acesso direto ao *backbone* do NSF. O *National Science Foundation* contratou 4 companhias que poderiam ser provedores de acesso ao *backbone* do NSF. Essas companhias poderiam então vender conexões para grupos, organizações e companhias.
- A NSFNET volta a ser uma rede de pesquisa. O tráfego principal do *backbone* US agora é roteado através da interconexão dos provedores de rede.
- A nova NSFNET nasce como NSF estabelecendo o *very high speed Backbone Network Service* (vBNS) ligado a grandes centros computacionais: NCAR, NCSA, SDSC, CTC e PSC.
- *Real Audio*, uma tecnologia do áudio, deixa a rede de escuta perto do *real-time*.
- Radio HK foi a estação de rádio a começar a transmitir sua programação via Internet.
- O WWW supera o FTP-DATA com o serviço de maior tráfego no NSFNET.
- Tradicionais sistemas *dial-up on-line* (Compuserver, America On-line, Prodigy) começam a fornecer acesso a Internet.
- Milhares de usuários da rede em Mineápolis St.Paul (USA) perderam acesso a rede depois que transeuntes começam uma fogueira sob uma ponte na Universidade de Minn causando o derretimento dos cabos de fibra-ótica.
- Registro de *domain names* não está longe de ser grátis.
- Domínios registrados: Etiopia (ET), Cote d’Ivoire (CI), Ilhas de Cozinhos (CK), Ilhas de Cayman (KY), Anguilla (AL), Gibraltar (GI), Vaticano (VA), Kiribati (KI), Kyrgyzstan (KG), Madagascar (MG), Mauritius (MU), Micronésia (FM), Mônaco (MC), Mongólia (MN), Nepal (NP), Nigéria (NG), Samoa Ocidental (WS), San Marino (SM), Tasmânia (TZ), Tonga (PARA), Uganda (UG), Vanuatu (VU).
- Tecnologias do ano: WWW e mecanismos de busca.
- Tecnologias emergindo: Código móvel (Java e JavaScript), Ambientes Virtuais (VRML) e ferramentas de colaboração.
- Criação da Microsoft Network (MSN).

## 1996

- A Internet conta com cerca de 9.500.000 *hosts* e mais de 30 milhões de usuários.
- A maioria dos tráfegos feitos na Internet é carregado por *backbones* de ISPs independentes, incluindo MCI, AT&T, Sprint, Uunet, BBN planet, ANS e outros.
- A *Internet Society*, o grupo que controla a Internet, está tentando imaginar um novo TCP/IP capaz de ter bilhões de endereços, em vez do sistema limitado de hoje. O problema era que ambos os sistemas endereçados eram capazes de trabalhar no mesmo tempo durante o período de transição.

- O *Controlversial US Communications Devency Act* (CDA) torna-se lei nos US para proibir a distribuição de materiais indecentes através da NET. Poucos meses depois o júri impôs um mandado contra sua coação. A Suprema Corte controla esta inconstitucional em 1997.
- O InterNIC deixa cair seu *name service* como resultado de não ter pago sua taxa específica de domínio.
- Vários ISPs estendidos sofrem o chamado *service outages*, trazendo uma pergunta: se ele poderá manusear a produção do número de usuários. AOL (19 horas), NETcom (13 horas), AT&T WordNET (28 horas – *e-mail* unicamente).
- *New Yorks' Public Access Networks Corp* (PANIX), está saindo depois de repetidos ataques feitos por um *cracker* usando métodos resumidos em uma revista de rackers.
- A *Internet Ad Hoc Commitee* anuncia planos para somar sete *Top Level Domains* (gTLD): .firme, .loja, .Web, .artes, .rec, .informação, .nom. O IAHC passa a ser conhecido como um grupo compreendido de registradores de domínio global.
- Um *cancelbot* é liberado pela USENET limpando mais de 25,000 mensagens fora.
- Na guerra dos *browsers* www, brigam principalmente Netscape e Microsoft que atacam em uma nova era desenvolvendo um software por onde liberações são feitas trimestralmente com a ajuda de impacientes usuários da Internet para testar próximas(beta) versões.
- Domínios registrados: Qatar (QA), Vientine (LA), Djibouti (DJ), Niger (NE), República Africana Central (CF), Mauretania (MF), Oman (OM), Ilha do Norfolk (NF), Tuvalu (TV), Polinésia Francesa (PF), Síria (SY), Aruba (AW), Cambodia (KH), Guiana Francesa (GF), Eritrea (ER), Cape Verde (CV), Burundi (BI), Benin (BJ), Bosnia-hercegovina(BA), Andorra(AD), Guadalupe (GP), Guerney (GG), Isle de Man (IM), Jersey (JE), Lao (LA), Maldives (MV), Ilhas de Marshall (MH), Mauritania (SR), Ilhas de Mariana do Norte (MP), Ruanda (RW), Togo (TG), Yemen (YE), Zaire (ZR).

## 1997

- 71.618 listas de discussão registradas no Liszt.
- O *American Registry for Internet Numbers* (ARIN) é estabelecido para tratar da administração e registro de números de IP das áreas geográficas atualmente controladas pela *Network Solutions* (INTERNIC) a partir de março de 1998.
- Domínios registrados: Ilhas de Falklans (FK), Timor Oriental (TP), Congo (CG), Christmas Island (CX), Gambia (GM), Guínea-Bissau (GW), Haiti (HT), Iraque (IQ), Líbia (LY), Malawi (MW), Martirique (MQ), Montserrat (DONA), Myanmar (MM), Ilha da Reunião Francesa (RE), Seychelles (SC), Sierra Leone (SL), Sudan (SD), Turkmenistan (TM), Turks e Ilhas de Caíco (TC).

### 1.1. Internet no Brasil<sup>1</sup>

A Internet chegou no Brasil em 1988, por iniciativa da comunidade acadêmica de São Paulo (Fundação de Amparo à Pesquisa do Estado de São Paulo - FAPESP) e de Rio de Janeiro (Universidade Federal do Rio de Janeiro - UFRJ) e também do Laboratório Nacional de Computação Científica – LNCC/CNPq.

Em Setembro de 1989 o governo brasileiro através do então Ministério da Ciência e Tecnologia, criou a Rede Nacional de Pesquisa (RNP). Esta agência passou a organizar a rede nacional unificada, trabalhando com o protocolo TCP/IP e está conectada com o exterior servindo como ponto de ligação entre as rede regionais, acadêmicas ou comerciais, e a Internet. Os objetivos de criar esta instituição eram de iniciar e coordenar a disponibilização de serviços de acesso à Internet no Brasil. Como ponto de partida foi criado um *backbone* RNP, interligando instituições educacionais à Internet.

<sup>1</sup> Para escrevermos esta seção visitamos os *sites* [4], [5] e [6].

Esse *backbone* inicialmente interligava 11 estados a partir dos Pontos de Presença (*Point of Presence* - POP) em suas capitais. A fim de integrar instituições de outras cidades à Internet, foram criados alguns *backbones* regionais ligados a esses pontos. Como exemplo desses *backbones* temos no Rio de Janeiro a Rede Rio e em São Paulo a *Academic Network at São Paulo* (ANSP).

Em dezembro/94 foi iniciada a exploração comercial da Internet a partir de um projeto piloto da Embratel, onde foram permitidos acessos à Internet inicialmente através de linhas discadas e posteriormente (abril/1995) através de acessos dedicados via RENPAC ou linhas E1.

Em paralelo a isso, a partir de abril/1995 foi desenvolvido pela RNP um processo de implantação comercial da Internet no Brasil com uma série de etapas, entre as quais a ampliação do *backbone* RNP no que se refere a velocidade e número de POPs, afim de suportar o tráfego comercial de futuras redes conectadas a esses POPs. Esse *backbone* a partir de então passou a se chamar Internet/BR.

O serviço da Internet Brasil foi aberto ao usuário não-acadêmico. Através de POPs, operados por instituições do governo, empresas passaram a se conectar à espinha dorsal da rede, ou *backbone*. Estas empresas, os chamados provedores de acesso, estavam liberadas para oferecer à rede aos usuários finais. Os provedores de acesso estão conectados aos POPs através de uma linha telefônica exclusiva, com uma velocidade mínima de 64 Kbps (64.000 bits por segundo).

Uma primeira etapa de expansão do *backbone* RNP foi concluída em dezembro/95 restando ainda a criação de POPs em mais estados. Além disso, algumas empresas (IBM, UNISYS, Banco Rural) anunciavam em 1996 a inauguração de *backbones* próprios.

O *backbone* da Internet Brasil é mostrado no desenho abaixo. Como se pode notar, todos os estados brasileiros possuem conexão à rede através de POPs. Existem também outras conexões ao exterior que não passam pelo *backbone* da Internet Brasil, como as do Banco Rural, Telebrás e da IBM Brasil.

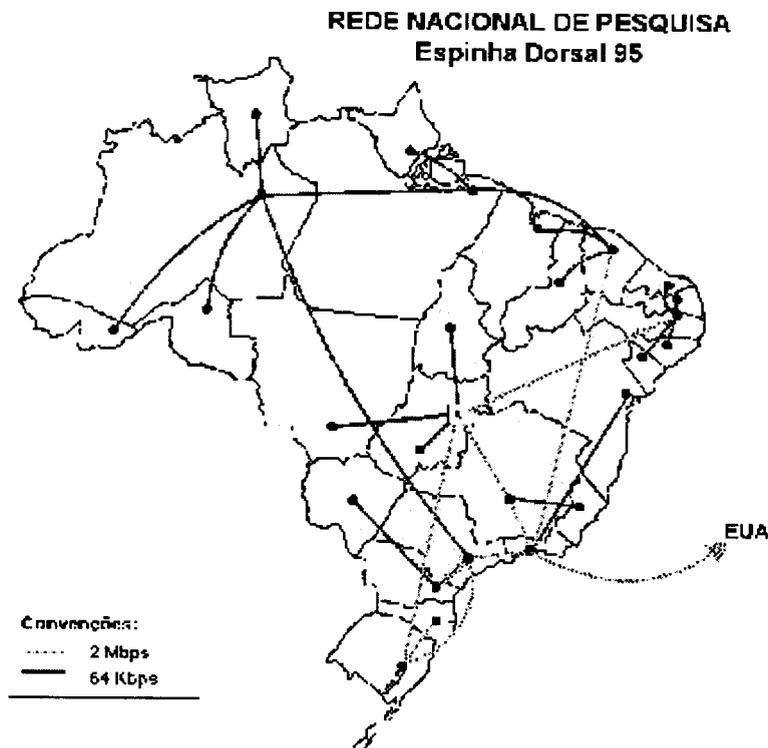


Imagem: Comitê Gestor Internet Brasil

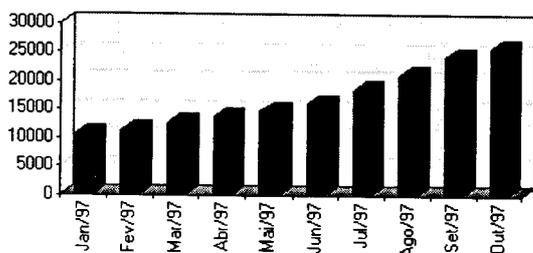
### 1.1.1. Domínios Registrados

Os dados sobre o número de domínios registrados são coletados mensalmente pelo *Comitê Gestor Internet Brasil* [6]. As informações brutas são coletadas através do programa *host*, e depois processadas através de filtros para se obter as informações finais.

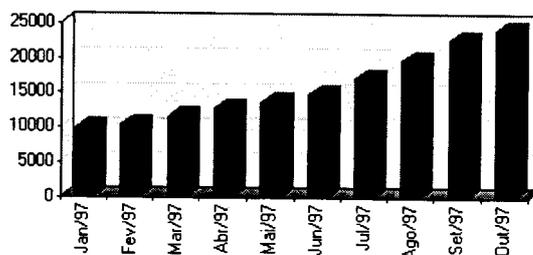
Na tabela abaixo são apresentadas estatísticas que mostram o crescimento do número de registros de domínios de janeiro à outubro de 1997. A seguir são mostrados os gráficos relativos.

Domínios	.com	.edu	.g12	.gov	.mil	.net	.org	.br
31/01/97	9.829	278	70	148	7	15	333	10.680
28/02/97	10.365	284	71	158	7	15	362	11.262
31/03/97	11.574	295	81	169	7	15	412	12.553
30/04/97	12.668	303	87	173	8	16	461	13.716
31/05/97	13.641	307	93	179	8	16	499	14.743
30/06/97	14.769	319	98	182	8	16	557	15.949
31/07/97	17.159	325	125	189	9	17	672	18.496
31/08/97	19.615	338	142	194	9	18	740	21.056
30/09/97	22.595	353	162	205	9	18	841	24.183
31/10/97	24.153	368	168	214	9	19	928	25.859
Em Outubro	6,90%	4,25%	3,70%	4,39%	0,00%	5,56%	10,34%	6,93%

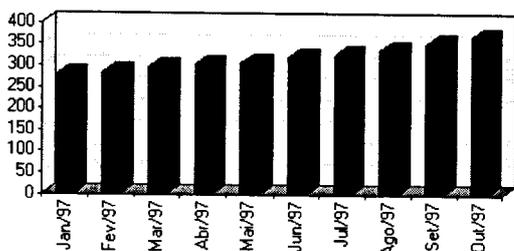
.br



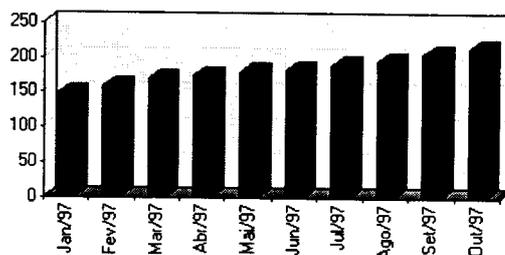
com.br



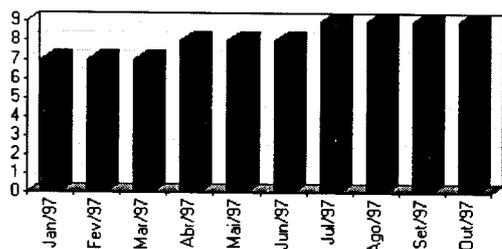
edu.br



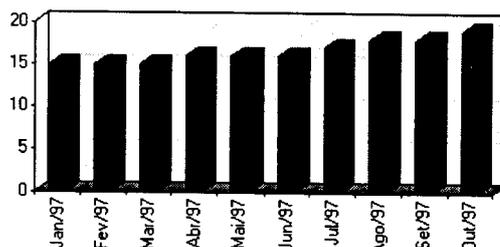
gov.br



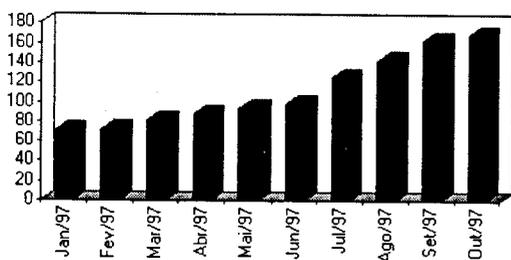
mil.br



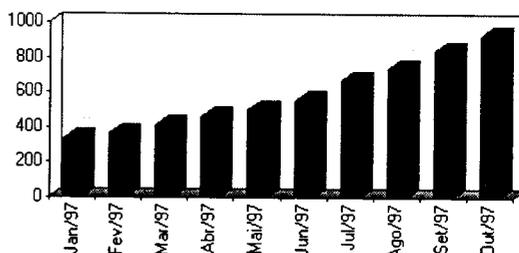
net.br



g12.br



org.br



É importante observar que principal aumento do número de domínios, ocorreu na área comercial (93,40%), seguido da área organizacional (3,59%) e educacional (1,42%). As outras áreas apresentaram um crescimento menor que 1%.

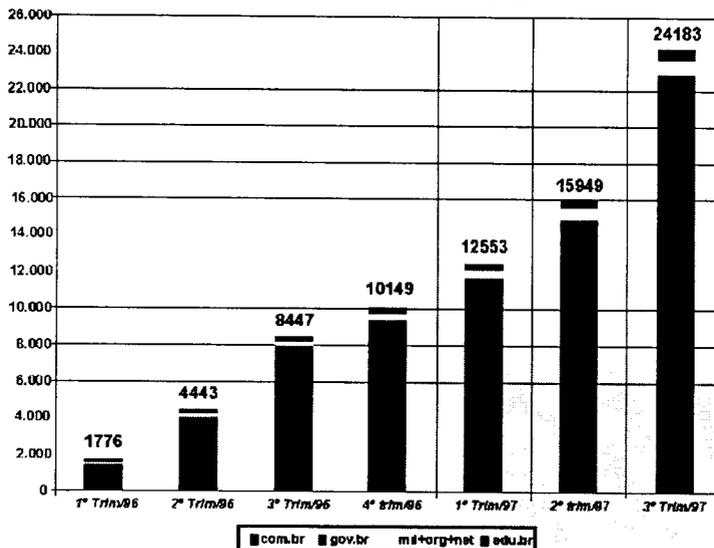
**Nota1:** Os valores apresentados foram coletados pelo GT-ER. Foram considerados apenas os domínios que possuem servidores de DNS ativos.

**Nota2:** A contagem está sujeita a alterações ocasionadas pela eventual dificuldade de conexões com os diversos *backbones*. A fim de minimizar este tipo de problema, são realizadas várias contagens e o maior resultado de cada sub-domínio é apresentado.

**Nota3:** Na Internet Brasil não existe o domínio edu.br. Os domínios das instituições acadêmicas ficam logo abaixo do domínio .br. Assim todos os domínios da forma "dom.br", exceto os domínios com.br, gov.br, org.br, mil.br, net.br e g12.br são os domínios apresentados como domínios acadêmicos e referenciados aqui como edu.br.



### Número de Domínios Internet no Brasil



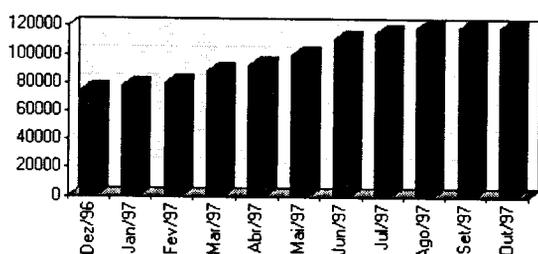
#### 1.1.2. Hosts por Domínio

Os dados sobre o número de *hosts* por domínios também são coletados mensalmente pelo Comitê Gestor Internet Brasil. As informações brutas são coletadas através do programa *host*, e depois processadas através de filtros para se obter as informações finais.

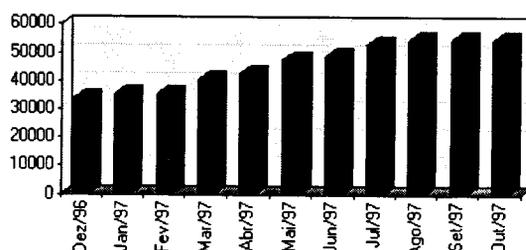
Na tabela abaixo são apresentados estatísticas que mostram o crescimento do número de *hosts* nos domínios de janeiro à outubro de 1997. A seguir são mostrados os gráficos relativos.

<i>Hosts</i>	com.br	edu.br	g12.br	gov.br	mil.br	net.br	org.br	.br
31/01/97	35.125	33.029	145	7428	153	493	969	77.342
28/02/97	35.201	35.008	147	7.853	153	493	1.044	79.899
31/03/97	40.522	37.191	175	8.134	153	513	1.094	87.782
30/04/97	42.422	38.890	214	8.618	153	527	1.181	92.005
31/05/97	47.439	40.402	229	9.054	163	547	1.332	99.166
30/06/97	48.863	50.047	248	9.316	164	555	1.431	110.624
31/07/97	52.863	50.047	297	9.957	164	575	1.555	115.458
31/08/97	54.447	51.502	304	10.195	165	602	1.634	118.849
30/09/97	54.447	51.502	391	10.195	168	602	1.634	118.939
31/10/97	54.447	51.502	444	10.722	172	602	1.634	119.523
Em Outubro	0,00%	0,00%	13,55%	5,17%	2,38%	0,00%	0,00%	0,49%

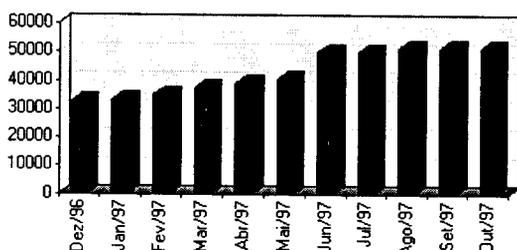
.br



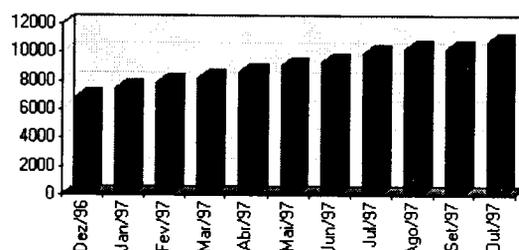
com.br



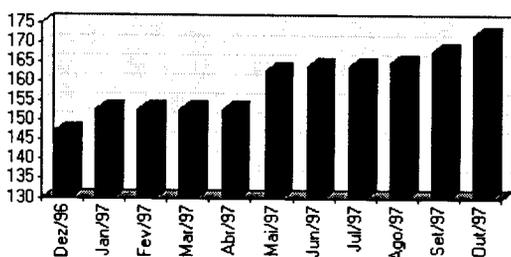
edu.br



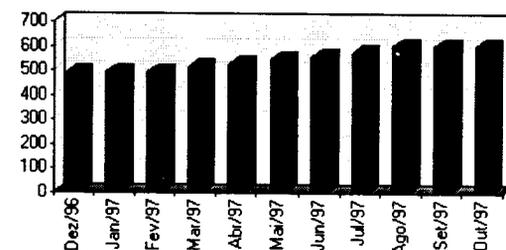
gov.br



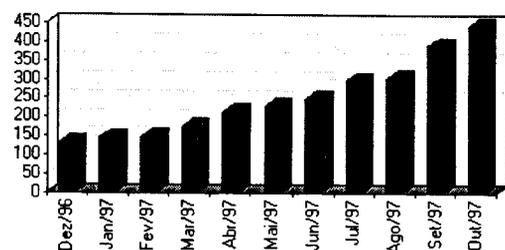
mil.br



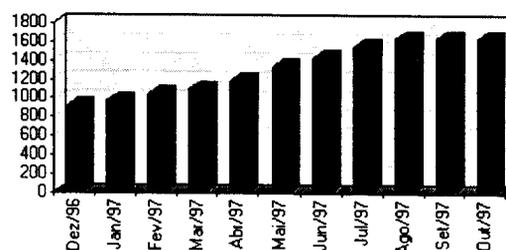
net.br



g12.br

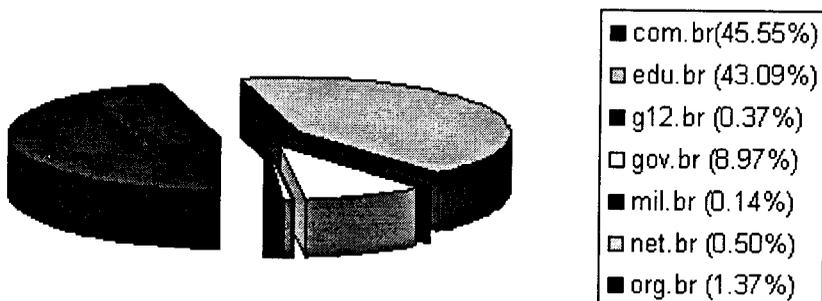


org.br

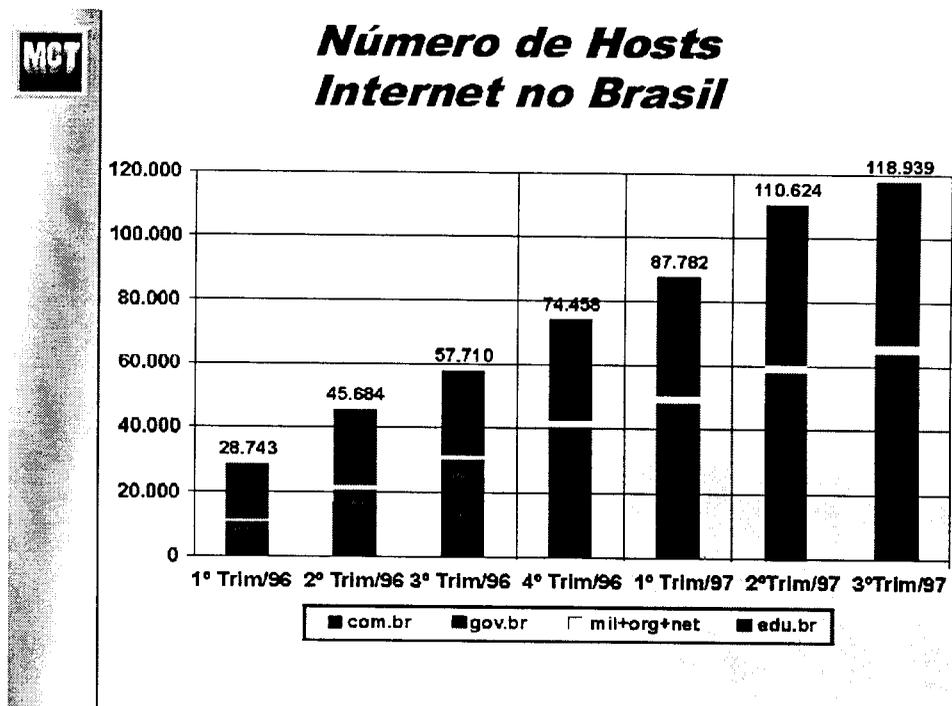


Porcentagem em relação ao total do nº de *hosts*

Outubro 1997



- Obs1:** A contagem está sujeita a alterações ocasionadas pela eventual dificuldade de conexões com os diversos *backbones*. A fim de minimizar este tipo de problema, são realizadas várias contagens e o maior resultado de cada sub-domínio é apresentado.
- Obs2:** Na Internet Brasil não existe o domínio edu.br. Os domínios das instituições acadêmicas ficam logo abaixo do domínio .br. Assim todos os domínios da forma dom.br, exceto os domínios com.br, gov.br, org.br, mil.br, net.br e g12.br são os domínios apresentados como domínios acadêmicos e referenciados aqui como edu.br.
- Obs 3:** A aparente estagnação do domínio edu não retrata de fato a realidade, apenas está retratando a dificuldade de obtenção destes dados. Lembramos que estes dados são apenas o resultado de amostras coletadas através de consultas DNS. Assim, como não existe o domínio .edu, como já citado na Nota2, temos que percorrer todas as subárvores de domínios abaixo do domínio .br para se obter os dados do domínio acadêmico. Devido as dificuldades de conexão com todos estes *sites* e as políticas de segurança adotadas, alguns dados estão deixando de ser coletados.



## 1.2. Ato Normativo para a Atribuição de Domínio na Internet no Brasil

- 1. DO REGISTRO** - O Comitê Gestor Internet Brasil - CG é o órgão responsável pelo registro, no país, de Nomes de Domínios na rede eletrônica Internet, observadas as condições abaixo:
- I. O registro adota como critério o princípio de que o direito ao nome do domínio é conferido ao primeiro requerente da inscrição. Entretanto, esta concessão poderá ser cancelada pelo Comitê Gestor da Internet Brasil, nos termos do disposto no item 5.
  - II. O Comitê Gestor da Internet Brasil não se responsabiliza pela escolha do Nome de Domínio requerido e utilizado pelo usuário, bem como por quaisquer danos decorrentes do seu uso indevido.
  - III. Impõe-se ao requerente o uso regular do Nome de Domínio requerido, na forma constante do documento de Solicitação de Registro de Nome de Domínio.
  - IV. É permitido o registro de Nome de Domínio tão-somente para pessoas jurídicas constituídas no país e devidamente inscritas no Cadastro Geral de Contribuintes - CGC do Ministério da Fazenda.
  - V. É assegurado a cada pessoa jurídica, legalmente habilitada, o registro exclusivo de um Nome de Domínio, salvo expressa autorização do Comitê Gestor da Internet Brasil.
  - VI. Incumbe ao Comitê Gestor da Internet Brasil a manutenção de registros de nomes nas categorias de domínios reservadas para entidades governamentais (.gov.br), não-governamentais (.org.br), comerciais (.com.br), militares (.mil.br), educacionais (".edu".br) e empresas de telecomunicações (.net.br), ou qualquer outro sub-domínio que venha a ser criado pelo CG sob o espaço reservado ao Brasil (.br) pelo InterNic.
  - VII. Refoge à inteira responsabilidade do Comitê Gestor da Internet Brasil o gerenciamento de novas divisões e sub-domínios criados pelo requerente sob o Nome de Domínio registrado em qualquer uma das categorias relacionadas no item 1.VI.

### 2. DAS TAXAS DE INSCRIÇÃO E MANUTENÇÃO DO SERVIÇO

- I. Nos casos de inscrição, em se tratando de nova solicitação de registro de Nome de Domínio, é devida uma taxa a ser fixada por instrumento próprio, pelo Comitê Gestor da Internet Brasil e,
- II. No que se refere a manutenção do registro do Nome de Domínio, para cada período de vigência de 12 meses, é devido uma taxa a ser estipulada por instrumento próprio pelo Comitê Gestor da Internet Brasil e paga antecipadamente pelo requerente. No caso de nova solicitação de registro, será acrescida a taxa de inscrição.

- ### 3. DAS RESPONSABILIDADES DO REQUERENTE
- Obriga-se o requerente a responder por quaisquer ações judiciais ou extra-judiciais que resultem de violação de direitos ou prejuízos causados a outrem, o que exime, por completo, o Comitê Gestor da Internet Brasil de quaisquer ônus decorrente de condutas danosas.

- ### 4. DAS MODIFICAÇÕES
- Reserva-se o Comitê Gestor da Internet Brasil a faculdade de, periodicamente, e, observado o prazo de 30 dias da divulgação da nova regra no endereço Internet <http://www.cg.org.br>, modificar os termos do presente ato normativo, bem como alterar os valores das taxas de inscrição e manutenção do sistema de registro, preservando-se as hipóteses de direitos adquiridos.

- ### 5. DO CANCELAMENTO
- Extingue-se o direito de uso de um Nome de Domínio registrado na Internet sob o domínio .br:

- I. pela renúncia expressa do respectivo titular, por meio de documentação hábil;
- II. pela expiração do prazo de proteção de 12 meses, contados a partir da data de concessão do Nome de Domínio, nos termos propostos pelo CG. Inexistindo renovação, o registro será cancelado dentro do prazo de 30 dias;
- III. pela caducidade em razão do não uso regular do Nome de Domínio, no período contínuo de 180 dias;

IV. por ordem judicial;

V. pela inobservância das regras estatuídas pelo Comitê Gestor, por intermédio de notificação por escrito, no prazo de 30 dias, cabendo a redução do lapso temporal, nas hipóteses de cumprimento de decisão judicial;

VI. Nas hipóteses acima mencionadas, não assiste ao requerente qualquer ressarcimento ou indenização pelo cancelamento do registro.

**DA PUBLICAÇÃO** - A divulgação das normas regulamentares do presente instrumento efetuar-se-á por intermédio do endereço eletrônico do servidor Web do CG na Internet: <http://www.cg.org.br>.

## 2. Entendendo a Internet

A Internet é realmente uma rede de redes de computadores que trocam informações entre si. Estes computadores podem ser de qualquer tipo, arquitetura, marca ou modelo. Podem ser microcomputadores ou computadores de grande porte. Podem usar qualquer processador e portanto qualquer sistema operacional. Podem usar qualquer software que permita comunicação entre servidores e clientes. Estes computadores estão interligados por linha comum de telefone, linhas privadas de comunicação, canais de satélite, cabos submarinos e outros meios de comunicação. Esta é uma e talvez a principal característica da Internet: a independência de hardware e software [7][8].

Na realidade, a palavra internet é proveniente da expressão *internetwork* (comunicação entre redes). Uma maneira simples de visualizar a Internet é considerar uma nuvem com computadores conectados a ela. Esta nuvem é dinâmica e cresce a medida que crescem as redes. Estas redes se comunicam através do protocolo TCP/IP (*Transmission Control Protocol/Internet Protocol*) que falaremos mais tarde.

O crescimento da Internet foi assustador nos últimos 10 anos. Em 1984 haviam 1.000 *hosts* – servidores de recursos. Já em 1989 haviam 100.000, em 1992 haviam 1.000.000, em 1994 3.200.000 com 20 milhões de usuários aproximadamente. Hoje, em 1997 cerca de 10.000.000 de *hosts* e mais de 30 milhões de usuários. Esta explosão está muito ligada ao uso comercial da Internet, onde negócios que vão desde a simples compra de um rádio, carro ou mesmo apartamento até complexas aplicações no mercado de títulos/capitais podem ser feitas de qualquer computador conectado à Internet.

É importante frisar que a Internet não tem um dono específico. Isso ocorre pelo fato que esta conexão só é feita localmente com as políticas locais de acordo. Não há administração, embora existam organizações que se dedicam a definir padrões, normas e regras para utilização e disponibilização de Internet. Essa organização pode ser meio confusa à primeira vista, porém, vem dando bastante certo e com isso contribuindo para o aumento significativo da rede.

Podemos caracterizar bem o objeto de nosso estudo através de uma lista de pontos observados que retratam por completo a essência do que a Internet é ou não é [11]:

- A Internet é um meio para comunicação efetiva entre pessoas ou processos.
- A Internet é um mecanismo de recuperação de informação e suporte à pesquisa.
- A Internet é flexível em relação ao preço e características.
- A Internet é simultaneamente uma entidade local e internacional, permitindo interação entre usuários separados por uma parede ou por um oceano.
- A Internet não especifica nem hardware nem software.
- A Internet não é uma única rede, mas um grupo de redes logicamente (não fisicamente) arrumadas hierarquicamente.
- A Internet não é propriedade de nenhum governo, corporação ou universidade.
- A Internet não é a mesma em todos os lugares (não homogênea) mas amplamente diferente (heterogênea).

- A Internet não é restrita somente a pesquisa, mas também a área comercial ou outros usos apropriados, apesar de alguns servidores serem dedicados.
- A Internet não é usada por um tipo específico de usuário, mas qualquer profissional, estudante ou pessoa comum.

No item 2.1. citaremos os tipos de instituições que possuem caráter regulador e no item 2.2 classificaremos o tipo de acesso e de conexão.

## 2.1. Administração da Internet

Tanto a administração quanto a operação da Internet são descentralizadas, apenas alguns serviços tais como definição de padrões e pesquisas e ainda a distribuição dos endereços são administrados por instituições regulamentadoras. As principais instituições são:

- **The Internet Society (ISOC)** – procura orientar a pesquisa e utilização através de fóruns, debates e publicações.
- **The Internet Architecture Board (IAB)** – fundada em 1983 e integrada ao ISOC em 1992, coordena os grupos IETF e IRTF descritos abaixo, na pesquisa e desenvolvimento envolvidos no funcionamento da Internet.
- **The Internet Research Task Force (IRTF)** – grupo de pesquisadores que se dedicam a projetos de longo prazo referentes ao funcionamento da Internet.
- **The Internet Engineering Task Force (IETF)** – grupo de pesquisadores responsáveis pelo desenvolvimento de padrões a serem divulgados através de Request for Comments (RFC) que no princípio tinham a intenção de serem propostas e tornaram-se padrões oficiais da Internet.
- **The Internet Network Information Center (InterNIC)** – composto de 3 instituições (AT&T, PSI e General Atomics) que organizam a distribuição dos endereços e registros de domínios e também das RFC's.
- **The Internet Assigned Numbers Authority (IANA)** – mantido pelo Instituto de Ciência e Informação da Universidade do Sul da Califórnia, controla a distribuição dos identificadores para serviços a serem oferecidos pela Internet.

No Brasil, o principal órgão de administração da rede é o Comitê Gestor Internet, criado em 1995 por iniciativa do Ministério das Comunicações e da Ciência e Tecnologia e seu principal objetivo é coordenar a implantação do acesso à Internet no Brasil. À nível de redes, a Rede Nacional de Pacotes (RNP) administra o *backbone* Internet/BR e as redes ligadas a este *backbone* são administradas por instituições locais como a FAPESP em São Paulo e FAPERJ no Rio de Janeiro.

## 2.2. Conectividade na Internet

Podemos classificar o acesso à Internet através do tipo de aplicações ou quanto a forma de conexão, tendo cada um duas categorias [12]. No primeiro tipo, podemos ter Acesso Completo quando o computador tem endereço Internet e portanto é um *host*, e Acesso Limitado quando o computador não tem endereço Internet e portanto precisa estar ligado à um outro que seja *host*, normalmente chamado de provedor.

No segundo tipo, relativo à forma de conexão, também temos duas categorias. A primeira, Conexão Permanente, inclui a ligação entre computadores quando é feita através de circuitos dedicados de comunicação e é usada somente por computadores que tenham Acesso Direto pois devem possuir endereço e nome de domínios fixos. A segunda categoria, Conexão Temporária, pode ser usada por computadores com Acesso Completo e também com Acesso Limitado pois é feita através de linhas telefônicas discadas.

A integração destas duas classificações nos leva as seguintes formas de acesso à Internet:

- Acesso Dedicado: via Conexão Permanente, com Acesso Completo à Internet, execução de aplicações clientes servidoras;
- Acesso Discado de Protocolo: via Conexão Temporária, com Acesso Completo à Internet, execução de aplicações clientes;
- Acesso Discado de Terminal: via Conexão Temporária, com Acesso limitado à Internet, via emulação de terminal e/ou transferência de informações via protocolos não TCP/IP;
- Acesso Discado UUCP: via conexão Temporária, com Acesso Limitado à Internet, via protocolos do pacote *Unix-to-Unix Copy Protocol* (UUCP) do Unix, com acesso apenas aos serviços padrão da rede USENET (*mail* e *Network News*).

## 3. Funcionamento da Internet

Vamos agora iniciar o estudo do Funcionamento da Internet. São quatro os itens que abordaremos em seguida [13][14]: Protocolo, Endereçamento, Roteamento e Domínios.

### 3.1. Protocolo TCP/IP

O TCP/IP – *Transmission Control Protocol/Internet Protocol* – é um conjunto de protocolos de comunicação, tais como o TCP, IP, ICMP, ARP, FTP, Telnet e NFS. Um protocolo nada mais é que um sistema de comunicação de dados, que permite que vários sistemas de uma rede interajam entre si. As informações enviadas pela Internet são dependentes do TCP/IP, fazendo com que ele seja utilizado como um protocolo primário de rede na Internet.

Este conjunto de protocolos tem como principal característica a capacidade de permitir a comunicação entre máquinas que diferem, por exemplo, no sistema operacional, na CPU, nas interfaces de rede, etc. A sua arquitetura é similar a do modelo OSI (*Open Systems Interconnection*), criada pela ISO (*International Organization of Standards*), em 1978.

#### 3.1.1. O Modelo OSI

O modelo OSI trata-se de uma estrutura que compatibiliza sistemas de comunicação de dados diferentes, sendo dividido em sete camadas que representam diferentes níveis de abstração e funções definidas.

Cada uma destas camadas fornecem serviços às camadas imediatamente acima ou recebem serviços das imediatamente abaixo. A seguir, temos uma tabela mostrando a disposição delas.

Número das Camadas	Funcionalidade
7	Aplicações
6	Apresentação
5	Sessão
4	Transporte
3	Rede
2	Link de Dados
1	Física

A primeira camada é denominada Camada Física. Ela controla o *hardware* da máquina, isto é, todas as especificações do meio físico, como interfaces mecânicas e elétricas, cabos, conectores, etc.

A segunda camada é a de Link de Dados, que trata do envio de informações. Este envio é realizado através do “empacotamento” e endereçamento destas informações. Além disso, a camada de Link de Dados está envolvida com o controle de fluxo de transmissão nas linhas de comunicação. O “empacotamento” dos bits recebidos e transmitidos é feito em unidades lógicas chamadas *frames* (*Ethernet*, *Token Ring*, etc.).

A Camada de Rede é a terceira camada. Ela tem a função de garantir a chegada das informações, através da criação de um mecanismo de endereçamento uniforme, de maneira que possamos interconectar redes de quadros diferentes.

A camada posterior a de Rede é a de Transporte. A confiabilidade na entrega de dados é responsabilidade dela, que a faz através de um controle de erros. Esta camada é a camada central.

A quinta camada é chamada de Camada de Sessão. Ela permite a comunicação de duas vias ao longo de uma conexão, como por exemplo a transmissão de um arquivo entre duas máquinas.

A sexta é a Camada de Apresentação, que controla a representação dos dados, sendo esta representação a mesma entre os dois lados envolvidos.

A última camada é a de Aplicações, que permite o acesso a rede para os usuários. É nela que se encontram os protocolos e funções (serviços remotos como manuseio de arquivos, correio eletrônico, acesso a banco de dados, etc.) necessárias para que as aplicações dos usuários realizem tarefas de comunicações.

### 3.1.2. As Camadas do TCP/IP

Assim como o modelo OSI, o TCP/IP também é constituído de camadas, sendo que cada uma delas tem uma função específica. Abaixo, mostramos cada uma delas.

**A Camada da Internet** : esta camada é uma *linchpin* que mantém toda a arquitetura do TCP/IP junta. Sua função é permitir que usuários enviem pacotes de uma rede qualquer e que eles viajem independentemente para os seus destinos (potencialmente em uma rede diferente). Eles podem até chegar em uma ordem diferente daquela em que foram enviados, sendo neste caso função das camadas superiores rearrumá-los, se for de desejo do usuário. A função desta camada é análoga a do sistema de *e-mail* (correio eletrônico). Uma pessoa pode colocar uma seqüência de cartas internacionais em uma caixa de *e-mail* em um país, e com um pouco de sorte, a maioria delas irão ser enviadas para os endereços corretos do país de destino. Provavelmente, estas cartas irão viajar uma a uma em direção a portas (*gateways*) de *e-mail* internacionais ao longo do caminho, mas elas serão transparentes para os usuários. Além disso, a fim de que cada país (cada rede) tenha seus próprios selos, prefere-se certos tamanhos de envelope, e que as regras de entrega sejam escondidas dos usuários. A Camada da Internet define um formato de pacote oficial e protocolo chamado IP (*Internet Protocol*). O trabalho da Camada da Internet é entregar pacotes IP onde eles supostamente irão. O roteamento de pacotes é claramente a principal característica desta camada, assim como evitar congestionamentos na rede.

**A Camada de Transporte** : esta é a camada superior a da Internet no TCP/IP. Ela é designada para permitir pares de entidades na fonte e que usuários de destino vençam dificuldades em uma conversa. Dois protocolos *host-to-host* são definidos aqui. São o TCP e o UDP (*User Datagram Protocol*), que serão explicados posteriormente.

**A Camada de Aplicações** : o TCP/IP não tem Camada de Sessão e Apresentação, já que não há necessidade de qualquer uma das duas existirem. Este ponto de vista mostra-se correto ao realizarmos experiências com o Modelo OSI. Estas camadas são de pequena utilidade na maioria das aplicações. No topo da Camada de Transporte existe a Camada de Aplicações. Ela contém todos os protocolos de alta velocidade. Os primeiros incluem terminal visual (Telnet), transferidor de arquivos (FTP), e *e-mail* (SMTP). O protocolo de terminal visual permite ao usuário de uma máquina qualquer se conectar a uma outra máquina distante e trabalhar nela. Já o protocolo de transferência de arquivos possibilita uma forma de mover dados eficientemente de uma máquina à

outra. Por último, o *e-mail* foi originalmente somente um tipo de transferidor de arquivos, mas depois um protocolo especializado foi desenvolvido para isto. Muitos outros protocolos foram adicionados ao longo dos anos, como por exemplo o DNS (*Domain Name Service*), usado para mapear nomes de usuários nos seus endereços de rede; NNTP, o protocolo usado para mover artigos novos pela rede, e o HTTP, que é o protocolo usado para buscar páginas no *World Wide Web*, além de muitos outros.

**A Camada Usuário-Rede (*Host-to-Network Layer*)** : abaixo da Camada da Internet há uma grande lacuna. O TCP/IP não diz muito sobre o que acontece aqui, exceto apontar que o usuário tem que se conectar com a rede usando algum protocolo de forma que possa assim enviar pacotes IP através dele. Este protocolo não é definido e varia de usuário para usuário e de rede para rede. Livros e artigos em geral sobre TCP/IP raramente falam a respeito dele.

### 3.1.3. Relações entre o modelo OSI e TCP/IP

Apesar da arquitetura do protocolo TCP/IP ser similar a do modelo OSI, eles não são completamente iguais. O TCP/IP não faz tão bem distinções entre camadas de topo de uma “pilha” de protocolos como o faz o modelo OSI. As três camadas de topo do modelo OSI (Camada de Aplicações, Camada de Apresentação e Camada de Sessão) são equivalentes aos protocolos de processo da Internet (*Internet Process Protocol*), que tem como exemplos o Telnet, FTP, SMTP, NFS, SNMP, e DNS.

A camada de transporte do modelo OSI é responsável pela transferência de dados de confiança, que na “pilha” de protocolos da Internet corresponde aos protocolos *host-to-host*. Exemplos deles, como já comentamos no item anterior, são o próprio TCP e o UDP. O primeiro é utilizado como um tradutor de mensagens vindas de protocolos de camadas superiores, além de prover o reconhecimento de dados necessário e o controle da conexão orientada do fluxo entre sistemas remotos. Já o UDP (*User Datagram Protocol*), apesar de semelhante, não é igual ao TCP, uma vez que ele não tem conexão orientada e muito menos passa o reconhecimento de dados.

O protocolo da Internet (IP) é responsável pelas comunicações (em que não ocorre conexão entre nós que estejam sendo postos em comunicação) entre sistemas. Ele tem a mesma funcionalidade da camada de rede do modelo OSI, que é responsável pelo envio de informações ao longo da rede. No caso do IP, além de permitir este movimento de informações, ele também auxilia na recepção de mensagens entre sistemas. Porém, ele não garante a transferência destas mensagens. O IP também pode fragmentar as mensagens em pedaços e então as remontar na direção do seu destino. Se os fragmentos chegarem fora de ordem, o IP reagrupa os pacotes na seqüência correta. O IP requer então que um endereço seja apontado para todo dispositivo da rede.

Este endereço é denominado endereço IP (*IP address*), sendo organizado como uma série de quatro octetos. Cada um destes octetos definem um único endereço, com parte dele representando uma rede (e opcionalmente uma subrede) e parte representando um nó particular na rede.

Abaixo, mostramos duas pequenas tabelas. A primeira compara as camadas do modelo OSI com as do TCP/IP, enquanto a segunda relaciona o modelo OSI com protocolos da Internet, sendo um deles o próprio TCP/IP. Assim, temos uma melhor idéia das relações entre ambos, resumindo assim o assunto principal deste tópico.

Modelo OSI	TCP/IP
Camada de Aplicações	Camada de Aplicações
Camada de Apresentação	Não existe
Camada de Sessão	
Camada de Transporte	Camada de Transporte
Camada de Rede	Camada da Internet
Camada de Link de Dados	Camada Usuário-Rede
Camada Física	

Modelo OSI	INTERNET	
Camada de Aplicações	Telnet/FTP/SMTP	NFS/SNMP/DNS
Camada de Apresentação		
Camada de Sessão	TCP	UDP
Camada de Transporte		
Camada de Rede	IP	
Camada de Link de Dados		
Camada Física		

### 3.1.4. Subredes

Criar uma subrede é um processo de divisão de uma rede lógica muito grande em pequenas redes físicas. As razões que levam à criação de uma subrede podem ser limitações elétricas da tecnologia de rede ou a necessidade de locais distantes serem conectados com uma linha de alta velocidade, entre outros.

As redes resultantes são pequenos buracos da rede inteira, que são mais fáceis de se gerenciar. Estas pequenas subredes comunicam-se entre si através de portas (*gateways*) e roteadores. Também, se não houver uma organização, poderá haver diversas subredes que estarão fisicamente na mesma rede. Isto pode ser feito logicamente dividindo-se as funções da rede em grupos de trabalho.

### 3.1.5. Redes na Internet

O *design* e a configuração de uma rede na Internet é similar à configuração de uma rede de computadores qualquer. Inclui-se aí vários tipos de nós, como por exemplo estações de trabalho, servidores, impressoras, *mainframes*, roteadores, *bridges*, portas, servidores de impressão e terminais. A Internet requer um único endereço IP para cada um dos seus dispositivos, sendo que um dispositivo, dependendo da sua função, pode ter mais de um endereço. Contudo, pelo menos um endereço é necessário para a comunicação com os outros dispositivos. Uma rede TCP/IP pode ser constituída de diversos sistemas conectados com uma área de rede local (LAN – *Local Area Network*), ou centenas de sistemas com conexões para milhares de sistemas na Internet. Cada organização pode criar o tipo de rede apropriada para as suas necessidades.

### 3.1.6. Roteadores e *Bridges*

Informações entre redes são compartilhadas através de roteadores e *bridges*. A utilização apropriada de um dos dois é determinada pelos requerimentos da rede, os protocolos envolvidos, a capacidade da rede e as demandas usuais. Uma escolha adequada dos componentes pode auxiliar uma rede a operar eficientemente, permitindo um futuro crescimento, e também assegura uma reabilitação contínua.

Um roteador envia mensagens entre redes do mesmo tipo, sendo inteligente o bastante para saber se a mensagem recebida deve ser transmitida para outra rede, ou para outro roteador. Roteadores operam na Camada de Rede e são geralmente relacionados com um protocolo como o IP ou o IPX (*Internet Packet Exchange*). A maioria dos roteadores que roteam tráfego IPX também roteam tráfego IP tão bem como.

O roteador é usado para conectar várias redes locais e grandes áreas de rede, provendo assim uma maneira de compartilhamento de dados entre redes.

Se uma rede usa muitos protocolos diferentes e outra rede somente usa IP, e queremos pô-las em comunicação, é necessário um roteador que utilize apenas endereços IP. O roteador assim previne que mensagens sejam postas em uma rede que não pode gerenciá-las.

Por outro lado, as *bridges*, assim como os roteadores, podem ser usadas para conectar redes locais com grandes áreas de rede. Elas compartilham informações indiferentes ao protocolo usado. Uma *bridge* também permite que duas redes que estejam conectadas entre si tenham diversos protocolos diferentes simultaneamente. As mensagens transmitidas por uma *bridge* geralmente não têm qualquer informação de roteamento adicional. As mensagens são deixadas sem qualquer distúrbio.

Adicionalmente, uma *bridge* transmite mensagens somente para endereços de rede no outro lado dela, mas pode transmitir todos os protocolos de rede.

### 3.1.7. Componentes de uma LAN

Como já vimos anteriormente, uma rede TCP/IP constitui-se de vários sistemas que estão conectados com uma área de rede local; no caso, o que chamamos de *Local Area Network* - LAN. Abaixo, mostramos as funções de seus componentes básicos, com o intuito de resumir as características principais deste tipo de rede [15].

**1. *Bridge*** : Um dispositivo que conecta dois ou mais segmentos de rede de um mesmo tipo de meio físico. Uma *bridge* examina os campos de endereços do *hardware* de um pacote de rede e os filtra baseando-se nos endereços de um segmento de rede ao outro e vice versa.

**2. *Roteador*** : Trata-se de um dispositivo que contém duas ou mais interfaces de rede. O roteador examina o endereço do protocolo de *software* (IP), seleciona uma rota apropriada e transmite um pacote entre redes separadas. Ele também transmite, geralmente, pacotes pertencentes a uma família individual de protocolos.

**3. *Gateway*** : Um dispositivo que interliga duas ou mais redes de comunicação, baseado em diferentes conjuntos de protocolos. Um porta efetua qualquer conversão de protocolo necessária.

**4. *Switch*** : É um dispositivo de função similar a de uma *bridge*, mas que executa uma conexão e desconexão dinâmica lógica entre dois segmentos de cabo sem intervenção do operador.

Um *switch* é um dispositivo de alta velocidade onde os múltiplos dados de *paths* podem ser determinados e utilizados simultaneamente.

**5. Concentrador** : O dispositivo central completo em que todos os usuários de uma instalação de par trançado Ethernet são conectados.

**6. Hub** : É um dispositivo central completo que tem vários tipos de pacotes de rede que podem fluir. O *hub* é muitas vezes um dispositivo com várias aberturas (*multislotted device*) que contém placas (*boards*) separadas que podem prover a funcionalidade de um repetidor, de uma *bridge*, de um *switch*, de um roteador, de uma porta, ou de um concentrador. Além do mais, o *hub* pode prover funções múltiplas entre segmentos de cabo e redes. Assim sendo, o *hub* cresceu em importância a medida que incorporava múltiplas funções.

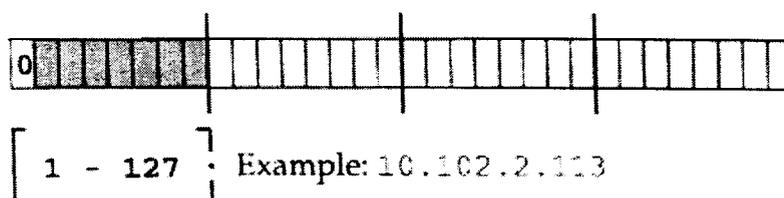
Podemos concluir que, com as diversas aplicações que vem emergindo atualmente, faz-se necessário diferentes tipos de sistemas de computadores que possam então compartilhar informações e programas, sendo que uma maneira eficiente e segura de realizar este compartilhamento é o uso dos protocolos da Internet (IP), sendo a entrega destas informações assegurada pelo TCP.

### 3.2. Endereçamento na Internet

O endereçamento de equipamentos (*devices*) na Internet é baseado em um identificador único que independe da tecnologia de rede envolvida ou sistema operacional da estação. Existe um outro modo de endereçamento conhecido como endereçamento em *multicasting*, onde um endereço IP está associado a um grupo de computadores. Este endereço, chamado endereço IP como visto anteriormente, é constituído por 32 bits, distribuídos em quatro campos de 8 bits (octetos). Cada octeto é representado por um número decimal entre 0 e 255, separados por ponto. Como exemplo podemos citar o servidor MESONPI cujo IP é 152.84.253.35, ou ainda a estação servidora do CBPF que chama-se "cbpfsu1" com 152.84.253.2 como IP.

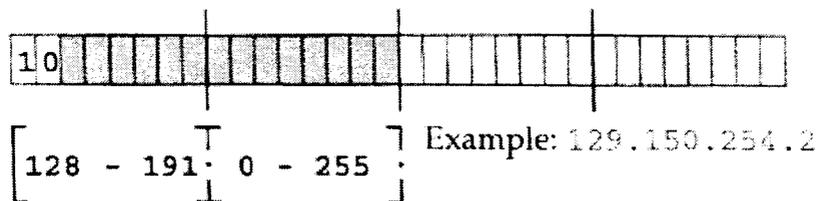
Cada endereço IP determina a rede, a subrede e o dispositivo. O valor do primeiro octeto determina qual parte do endereço IP é relativo ao número de rede e de subrede e qual identifica o dispositivo na subrede. Os números de rede são divididos em quatro classes: Classe A, Classe B, Classe C e Classe D. Passaremos a descrever cada uma das classes a seguir.

Classe A: se o primeiro bit for 0, então os próximos 7 bits serão destinados ao número de rede e os 24 bits (3 octetos) restantes, aos números de dispositivo. Assim, teremos 127 classes de rede do tipo A e mais de 16 milhões de dispositivos.

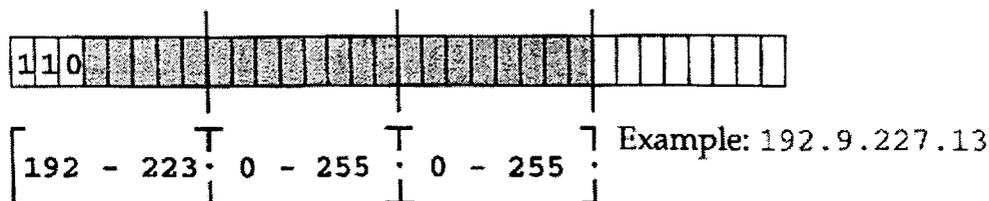


Obs: A classe 127, chamada loopback, é utilizada para testes onde o host envia pacotes para si próprio. ex.: 127.0.0.0

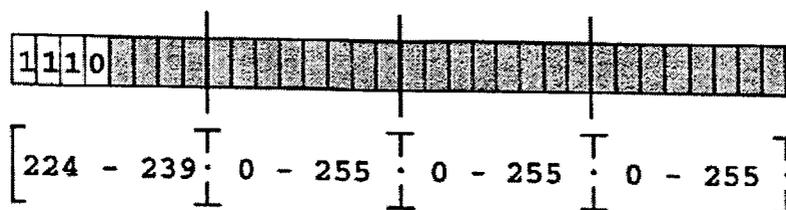
Classe B: se os dois primeiros bits forem 10, então os próximos 14 bits serão destinados ao número da rede e os 16 bits (2 octetos) restantes aos números de dispositivos. Assim, teremos 16.384 redes de classe B e mais de 65 mil dispositivos



Classe C: se os três primeiros bits forem 110, então os próximos 21 bits serão destinados ao número de rede e os 8 bits (1 octeto) restantes aos números de dispositivos. Assim, teremos 2.097.152 redes classe C e até 254 dispositivos



Classe D: se os quatro primeiros bits forem 1110, então o valor do primeiro octeto pode variar entre 224 e 239 e dizemos que este número é um endereço *multicast*. Os próximos 28 bits compõem um número de identificação de grupo para um específico grupo *multicast*.



Podemos concluir das definições de classes acima que um endereço IP *multicast* é um endereço destinado a um ou mais *hosts* ou dispositivos, enquanto um endereço Classe A, B ou C especifica o endereço de um *host* ou dispositivo individual.

## 4. Aplicações

A maior vantagem em trabalhar conectado em uma rede de computadores é a habilidade de compartilhar recursos e informações remotamente em algum *host*. Abordaremos em seguida três dos recursos mais utilizados na Internet: FTP, Telnet e Mail [10]. O ambiente Web será abordado em profundidade no próximo capítulo.

### 4.1. *File Transfer Protocol* - FTP

É o serviço padrão da Internet que copia ou faz transferência de arquivos entre computadores que estejam conectados em uma rede TCP/IP. O FTP permite ao usuário transferir tanto arquivos ASCII (texto) quanto arquivos binários (codificados).

Este serviço se baseia no estabelecimento de uma sessão padrão limitada entre o cliente FTP local e o servidor FTP do equipamento remoto. Durante uma sessão FTP, o usuário conecta-se com o outro computador usando o cliente FTP. A partir deste ponto, ele, o cliente, pode mover-se ao longo da árvore de diretórios, do conteúdo da lista de diretórios, copiar arquivos do computador remoto para o seu computador e transferir arquivos do seu computador para o sistema remoto.

Esta sessão é autenticada de forma parecida à do serviço Telnet, que abordaremos em seguida, possuindo apenas comandos referentes à manipulação de diretórios e arquivos, permitindo ao usuário pesquisar a estrutura de arquivos do equipamento remoto antes de fazer a transferência de arquivos propriamente dita. É importante ressaltar que clientes FTP possuem grupos de comandos diferentes dependendo do sistema operacional utilizado. Entretanto, estes grupos variam pouco de sistema operacional para sistema operacional, evitando assim maiores problemas de compatibilidade.

O uso mais comum do serviço FTP na Internet é a obtenção de programas ou informações partindo de servidores de domínio público ou comercial. Este serviço é conhecido como FTP Anônimo (*Anonymous FTP*). Para utilizá-lo, o usuário deve iniciar uma sessão FTP para o sistema remoto e usar como *user name* a palavra *anonymous*, seguida da *password*, que será, em geral, o seu endereço de *e-mail*. Esta conta é especial, pois possui uma autenticação flexível do correio eletrônico do usuário somente para controle estatístico ou posterior comunicação. A sessão assim estabelecida tem acesso somente aos arquivos que puderem ser consultados ou transferidos para o computador do usuário e isto é definido pelo *host* servidor.

Obs: Muitos *sites* não concedem FTP Anônimo. Conceder a usuários do tipo *guest* permissão para se conectarem ao seu computador envolve alguns riscos. Nos casos em que o FTP Anônimo não é aceito, o comando FTP envia uma mensagem similar àquela quando o *login* falha – “*User anonymous unknown*”. Os *sites* que permitem FTP Anônimo geralmente colocam o usuário em uma árvore de diretórios restrita que tem apenas acesso à leitura. Se você tem permissão para colocar arquivos no computador remoto, você usualmente pode apenas colocá-los em um diretório.

Os conjuntos de arquivos referentes a programas ou assuntos disponibilizados em um servidor FTP Anônimo recebem o nome de repositório. Porém, por medidas de segurança e facilidade de acesso, existe um programa chamado de *FTP mirroring*, que faz cópias de um dado repositório em mais de um equipamento da Internet.

Um grande problema do FTP Anônimo é a dificuldade que se tem para encontrar arquivos localizados na Internet, uma vez que, por ela ser um local muito extenso, demoramos muito para encontrar os arquivos de nosso interesse. Pensando nisso, foi criado o *Archie*, um

programa examinador de base de dados que contacta sites do FTP Anônimo ao redor do mundo e obtém do *site* FTP uma lista completa de todos os seus arquivos/diretórios. O *Archie* então pega esta informação e a indexa em sua própria base de dados. Os servidores *Archie* podem ser executados de duas formas: você pode executar um programa cliente local que se conecta com um servidor *Archie*, ou pode se conectar com o servidor diretamente através do Telnet.

#### 4.1.1. Comandos do FTP

Abaixo, temos uma tabela com alguns dos comandos existentes no sistema operacional *Linux* e que são comuns a maioria dos sistemas operacionais. Um ponto importante sobre comandos FTP é que existem *hosts* remotos que podem não suportar todos os comandos deste tipo.

Comandos	Funções
<i>ascii</i>	Coloca o FTP em modo ASCII (quando for transferir arquivos de tipo texto).
<i>binary</i>	Coloca o FTP em modo binário (geralmente quando os arquivos a serem transferidos não forem de tipo texto. Ex.: <i>.zip</i> , <i>.com</i> , <i>etc.</i> ).
<i>cd</i>	Este comando permite a mudança de diretório na estação remota.
<i>cdup</i>	Retorna ao diretório anterior em que você se encontrava.
<i>delete</i>	Apaga um arquivo remoto.
<i>dir</i>	Mostra o conteúdo do diretório corrente.
<i>disconnect</i>	Termina uma sessão FTP.
<i>get</i>	Transfere um arquivo para o computador do usuário.
<i>hash</i>	Inibe ou não a exibição do caracter “#” na hora da transferência. Isto permite ao usuário ver como está o andamento do processo.
<i>help</i>	Lista os comandos do FTP.
<i>lcd</i>	Este comando permite a mudança de diretório na estação local.
<i>mdelete</i>	Apaga um grupo de arquivos.
<i>mdir</i>	Mostra o conteúdo de múltiplos diretórios correntes.
<i>mget</i>	Transfere para o computador do usuário um grupo de arquivos.
<i>mkdir</i>	Cria um diretórios na máquina remota.
<i>mput</i>	Realiza a transferência de um grupo de arquivos de computador do usuário para outro que está sendo acessado.
<i>open/close</i>	Abre/Fecha uma sessão FTP.
<i>prompt</i>	Força uma indução interativa entre múltiplos comandos.
<i>put</i>	Transfere um arquivo do computador do usuário para o que está sendo acessado.
<i>pwd</i>	Diz em que diretório se encontra o usuário.
<i>quit/bye</i>	Encerra o FTP.
<i>recv</i>	Recebe arquivos remotos.
<i>rename</i>	Renomea arquivos remotos.
<i>rmdir</i>	Remove diretórios na máquina remota.
<i>status</i>	Mostra o <i>status</i> corrente.
<i>verbose</i>	Ativa/Desativa o código na mensagem de erro

**Obs:** A maioria dos servidores FTP baseados no sistema operacional UNIX, faz distinção entre letras maiúsculas e minúsculas (*case sensitive*), enquanto que aqueles baseados no sistema Windows NT, não. Desse modo, o usuário deve certificar-se que está utilizando a caixa (maiúscula ou minúscula) adequada quando transferir recursos a partir dessas instalações FTP.

## 4.2. Telnet

O comando Telnet é uma ferramenta básica dentro do sistema operacional UNIX. É uma interface de usuário para um sistema remoto usando o protocolo Telnet. Através deste comando, um usuário da rede pode se conectar em outro computador conectado na Internet e, então, utilizar os recursos desta máquina, através da execução dos seus comandos.

Temos que os equipamentos remotos devem possuir um sistema operacional multitarefa (executa mais de uma aplicação simultaneamente, compartilhando o tempo de CPU), que contenha mecanismos de autorização de acesso via sistema de contas, justificando assim a classificação do serviço Telnet como um serviço tipo *Remote Login* da Internet.

A execução do *logon* em um computador qualquer conectado na Internet via Telnet pode ser feita conhecendo-se um nome de usuário (*user name*) válido e uma senha (*password*) na máquina remota. Isto é feito através de um programa cliente Telnet, que permite que o usuário interaja com o serviço Internet. Esta interação ocorre selecionado-se o equipamento onde se deseja executar uma dada aplicação. O servidor Telnet é acionado, enviando então um *prompt* para o estabelecimento da sessão, pedindo o nome do usuário e a senha necessários. Uma vez iniciada a sessão, o usuário poderá utilizar qualquer aplicação desse equipamento autorizado para esta sessão.

Existem sistemas que oferecem a variante de um *logon* como *guest*. Contudo, ao conectar-se dessa forma, a sua área de rede pode ser vista por qualquer um que venha a se conectar também como *guest*, enquanto que um *logon* específico, com um *user name* e uma *password* válidos, garante a segurança necessária para você trabalhar na sua área própria.

O Telnet é um serviço que pode ser muito útil, como por exemplo, ao disponibilizar serviços da Internet que não estejam disponíveis localmente, através da execução do lado cliente desses serviços em outros equipamentos. No entanto, a sua aplicação mais útil é a de permitir acesso remoto a qualquer *host* da Internet que disponibilize este serviço. Acesso remoto é talvez a maior facilidade da Internet.

### 4.2.1. Comandos do Telnet

Assim como no FTP, não existe um padrão único para definir o grupo de comandos do serviço Telnet. Os comandos podem variar de sistema para sistema. Porém, existem alguns comandos considerados básicos, facilmente encontrados nas implementações UNIX do Telnet. Abaixo, temos uma pequena tabela que mostra quais são estes comandos e as suas respectivas funções:

Comandos	Funções
<i>Open</i>	Estabelece uma conexão Telnet com um <i>host</i> remoto.
<i>Close</i>	Termina uma conexão Telnet.
<i>Quit</i>	Fecha o Telnet corrente (se houver) e termina o Telnet.
<i>z</i>	Suspende o Telnet de forma que comandos possam ser executados no computador local.
<i>?</i>	Obtém ajuda no uso dos comandos Telnet.

**Obs:** O comando *z* não é reconhecido em todas as implementações do serviço Telnet – nem mesmo em todas as implementações UNIX. Você deve consultar a documentação da sua versão particular do Telnet para determinar se o comando *z* é aceito e, se for, como reiniciar o Telnet após o comando *z*.

### 4.3. Mail

Historicamente o Correio Eletrônico talvez seja a aplicação na Internet mais utilizada pelos usuários e é provavelmente a de mais fácil utilização. Assim como é possível colocar uma simples carta em uma caixa postal regular, o usuário pode escrever, endereçar e enviar eletronicamente um *e-mail* (forma compacta de *eletronic mail*) para qualquer pessoa, desde que esta pessoa possua um endereço eletrônico. O serviço *e-mail* é um método de enviar mensagens de um usuário em um computador para um outro usuário, chamado usuário receptor, em um *host* destinatário.

Existem várias vantagens em utilizar este serviço. Podemos citar:

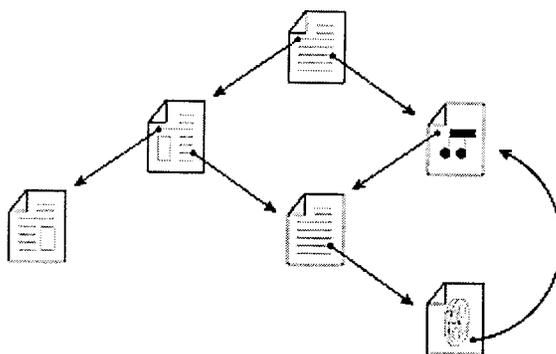
- É possível enviar relatórios, dados e documentos que chegarão em seu destino em poucos minutos e até mesmo segundos.
- Não obrigatoriamente o usuário receptor será interrompido por alguma mensagem. Este usuário poderá ler a mensagem em qualquer momento posterior conveniente.
- Não é necessário nenhuma comunicação direta com o usuário receptor, nem mesmo é necessário que ele esteja conectado na rede naquele momento.
- É possível enviar arquivos (*files*) ou documentos desenvolvidos em qualquer outro aplicativo (editores de texto, planilhas, banco de dados, editores gráficos, etc.).

Basicamente existem dois tipos de programas utilizados neste recurso. O primeiro deles, chamado *user agent*, que é utilizado pelos usuários para construir e enviar *outgoing mail* ou receber, ler e organizar *incoming mail*. O segundo deles, *mail delivery*, não é utilizado diretamente pelo usuário, e serve para pegar a mensagem trabalhada pelo *user agent*, e enviá-la ao *host* destino. Este tipo de programa geralmente é controlado pelo sistema operacional do *host* originário e destinatário sem controle do usuário. Existe uma variedade grande dos dois tipos de softwares. Como exemplos de *user agent* podemos citar Mailbox (Netscape), Outlook (Microsoft), Eldora e Pegasus.

## 5. Ambiente Web

A World Wide Web (Web ou WWW ou ainda W3) oferece uma interface gráfica na qual é fácil "navegar" na Internet observando vários tipos de documentos nos milhares de *hosts* disponíveis na Internet. Esses documentos, assim como os *links* existentes entre eles, compõem uma "teia" (*Web*) de informações [16].

Essa "teia" permite que você salte ou faça um *hyperlink* de uma página da Web para outras páginas na "teia". Pense na Web como uma grande biblioteca. Os *sites* da Web são como os livros, e as páginas da Web são como páginas específicas dos livros. Um conjunto de páginas da Web é denominado um *site* da Web. Sua viagem começa em um *site* específico da Web. Estas páginas podem conter notícias, imagens, filmes, sons, mundos em 3D -- praticamente tudo. Essas páginas podem estar localizadas em computadores de qualquer lugar do mundo. Uma vez conectado à Web, você dispõe do mesmo acesso as informações no mundo inteiro; não há restrições nem taxas adicionais de longa distância.



A World Wide Web está mudando o modo como as pessoas se comunicam no mundo todo. Essa nova mídia global vem aumentando de popularidade mais rapidamente que qualquer outro meio de comunicação em toda a história. Nos dois últimos anos, a Web cresceu e passou a incluir uma enorme variedade de informações -- tudo, de ações nas bolsas de valores a oportunidades profissionais, de serviços, de mensagens a notícias, lançamentos da indústria cinematográfica, revistas literárias e jogos. Os tipos de informações variam das mais irrelevantes até as mais importantes a nível mundial. Muitas vezes as pessoas falam em "surfear" pela Web e visitar novos *sites*. "Surfear" significa seguir os *hyperlinks* para páginas e assuntos sobre os quais você jamais ouviu falar, encontrar novas pessoas, visitar novos lugares e aprender coisas do mundo inteiro.

Lembre-se, a Internet não se resume a informações de corporações. Por ser tão fácil publicar na Web, muitas pessoas estabeleceram páginas principais pessoais, páginas a respeito de si mesmas e seus interesses, com suas próprias fotos e muito mais. Alguns têm até ponteiros (*links*) para o que estão vestindo no escritório naquele dia ou seu animal de estimação.

A página principal é o ponto de partida de um site da Web. Equivale a folha de rosto ou ao Índice Geral de um livro. Cada página da Web, incluindo a página principal do site da Web, possui um endereço exclusivo denominado um *Universal Resource Locator* (URL). Um "navegador" é uma ferramenta de software que você usa para ver as páginas da Web.

## 5.1. Navegadores

Um navegador (*browser*) é um programa de computador que interpreta documentos HTML, permitindo a visualização de textos e imagens na ordem especificada por ele. O primeiro navegador utilizado foi o *Mosaic*, desenvolvido pela NCSA, que servia apenas para visualizar as páginas. Ele não durou muito tempo, pois empresas como Microsoft e Netscape iniciaram suas atividades nesse ramo, fabricando navegadores mais avançados, que cabaram assim superando o *Mosaic*.

As versões mais novas dos navegadores incluem não apenas os navegadores em si, mas também ferramentas como um editor de *e-mails*, um editor de páginas HTML, *bookmarks* (que servem para guardar o endereço de páginas muito visitadas), recursos multimídia, entre outros. Tornaram-se verdadeiros pacotes com utilitários para Internet. Como se isso não bastasse, os navegadores são capazes de utilizar *plug-ins*, programas desenvolvidos por outras empresas para expandir a capacidade do *browser*. Um exemplo de *plug-in* é o Real Player, que permite assistir à programas de TV e escutar estações de rádio.

Para realizar a visualização de uma página, o navegador, baseado no endereço URL (que deve ser indicado pelo usuário), procura na Internet pelo *site*. Ao encontrá-lo, o navegador conecta-se com o sistema remoto, buscando e interpretando as informações contidas no documento HTML especificado. Após a interpretação da página, ela aparece na tela do usuário.

Atualmente, são dois os navegadores mais utilizados: o Navigator, fabricado pela Netscape; e o Internet Explorer, da Microsoft. Abaixo, mostramos uma pequena tabela em que são comparadas as principais características destes navegadores, sendo mostradas as versões 4.0 do Explorer e a versão 4.03 do Navigator, para assim termos uma idéia da funcionalidade de cada um, uma vez que tratam-se dos principais *browsers* existentes no mercado hoje em dia.

Características	Internet Explorer	Navigator
<b>Velocidade</b> HTML (em segundos) Java (índice CaffeineMark)	9,8 985	12,2 755
<b>Interface</b>	Permite personalizar a barra de ferramentas e tem mais opções de impressão que o Navigator.	Oferece uma interface única entre os vários módulos e admite a personalização da barra de links.
<b>Bookmarks</b>	Cada <i>bookmark</i> é um arquivo separado; este recurso funciona integrado com o de <i>webcasting</i> e com a navegação <i>offline</i> .	Utiliza um arquivo único (HTML) para todos os <i>bookmarks</i> e possibilita a sua edição
<b>Navegação offline</b>	Permite navegar no cache (área de armazenamento temporário) de arquivos.	Páginas específicas podem ser carregadas para exibição <i>offline</i> .
<b>Webcasting</b>	Vários serviços já estão disponíveis em português.	Tudo está em inglês, incluindo o programa.
<b>E-mail</b> Múltiplas contas		Sim, mas mudar de conta é

Filtros	Sim Seleciona mensagens com base no cabeçalho.	difícil. Seleciona com base no cabeçalho e no conteúdo.
<b>Teleconferência</b> Telefonia via rede Conferência por vídeo Compartilha aplicativos Recurso de <i>whiteboard</i>	Sim Sim Sim Sim	Sim Não Não Sim
<b>Segurança</b>	Emprega criptografia e admite ajustes diferentes para o micro, rede corporativa e a Internet.	Suporta os principais padrões de criptografia.
<b>Tamanho do arquivo de instalação (MB)</b>	25,1	14,7
<b>Tempo estimado de download (velocidade média de transferência de dados de 16 Kbps)</b>	3h30	2h05

## 5.2. Procura de Informação

Este serviço é sem dúvida o coração da Internet. Sem ele como iríamos achar as informações que desejamos? Hoje em dia existem muitos *sites* de busca na Internet. Conseguem-se achar praticamente tudo o que desejarmos, basta saber como e onde procurar. Os *sites* de busca contém um enorme banco de dados, no qual ficam armazenados os endereços Web e descrições sobre estes endereços. Quando mandamos procurar alguma coisa, um programa varre o banco de dados em busca do que pedimos e retorna os resultados como uma página HTML. Esta página contém links para todos os endereços encontrados que se enquadram nos critérios de busca.

Normalmente, ao procurarmos por informações, digitamos apenas uma palavra relacionada ao assunto de interesse. Podemos também combinar duas ou mais palavras, mas para isso deve-se ler as sintaxes de busca do *site*. Veja abaixo alguns exemplos que são utilizados no ALTAVISTA.

- Quando quisermos que uma palavra específica apareça em todos os documentos retornados pela busca, devemos colocar o sinal de + antes da palavra. O sinal de - serve para excluir uma determinada palavra dos resultados da busca.
- Se você quer que os resultados da sua busca possuam exatamente uma frase, escreva-a entre aspas. Por exemplo, “**Netscape Communicator**” retorna endereços que contenham em sua descrição a palavra **Netscape** seguida de **Communicator**.
- Você pode usar a sintaxe **text:texto\_para\_procurar** para retornar páginas que contenham em qualquer parte do documento a *string* **texto\_para\_procurar**. Por exemplo, **text:vaca louca** retorna as páginas que contém em qualquer parte do documento o texto **vaca louca**.

No Brasil, um *site* muito popular é o “CADÊ?” (<http://www.cade.com.br>). No exterior os mais usados são o “YAHOO” (<http://www.yahoo.com>), o “LYCOS” (<http://www.lycos.com>) e o

“ALTAVISTA” (<http://www.altavista.digital.com>). Este último, já tem sua versão brasileira também (<http://www.altavista.com.br>). É muito importante divulgar sua página na Internet, caso contrário, ninguém vai ficar sabendo que ela existe. Os *sites* de busca oferecem um cadastramento de endereços URL. Após o cadastro, o endereço e a descrição da sua página ficam armazenados no banco de dados. Quando alguém procurar por algo que se enquadre nas descrições da sua página, ela será apresentada. Alguns *sites*, como por exemplo “ALTA VISTA”, possuem um sistema automático de busca de novas páginas que alimenta a sua base de dados.

## 6. Criação de Páginas Web

Esta seção se propõe a mostrar os principais aspectos da criação de uma página Web (*home page*). Mostraremos aqui dois aspectos: linguagem e aplicações externas.

A linguagem base destas páginas (HTML) é descrita abordando os marcadores básicos e também aqueles mais utilizados de uma maneira geral. Esta linguagem pode utilizar programas executáveis remotamente utilizando um certo padrão de interfaceamento (CGI), o que torna dinâmico o uso da Web.

No próximo capítulo abordaremos a linguagem JavaScript que colocou este dinamismo além do servidor, agora a nível de usuário.

### 6.1. Hypertext Markup Language (HTML)

Esta linguagem foi originariamente desenvolvida por Tim Berners-Lee quando trabalhava no CERN, e popularizada pelo navegador Mosaic desenvolvido no NCSA. Durante o começo dos anos 90, junto com o crescimento explosivo da Web, esta linguagem foi extendida de várias maneiras o que motivou aos técnicos a trabalharem no sentido de unificarem as especificações. A versão HTML 1.0, que surgiu em 1991 era basicamente a linguagem utilizada pelos navegadores Web. A capacidade de mostrar imagens só ficou disponível em junho de 1993 na versão HTML+. A versão HTML 2.0 foi finalizada em dezembro de 1994, oficialmente sob controle de um *Internet Engineering Task Force (IETF) Working Group*[17]. A versão HTML 3.0 e 3.2 foram definidas em 1995 e janeiro de 1997, respectivamente. Recentemente, dezembro de 1997, a versão HTML 4.0 foi estabelecida e trazia além dos aspectos texto, multimídia e *hiperlink* das versões anteriores, mais opções multimídia, linguagens de *scripts*, mais opções de impressão, etc.

Abordaremos aqui oito tipos de marcadores amplamente usados pelo HTML [21]: 1. Básico; 2. Títulos; 3. Formatação; 4. Imagens; 5. Listas; 6. *Links*; 7. Tabelas e 8. Formulários.

#### 6.1.1. Básico

Todo documento HTML deve conter esta estrutura básica:

```
<HTML>
<HEAD>
<TITLE> Título da página </TITLE>
<HEAD>
<BODY>

</BODY>
</HTML>
```

Esta é a aparência de um novo documento HTML que tem as declarações básicas que toda página deve ter. Você pode criar um *template* de um documento HTML contendo essas declarações básicas. Assim, sempre que for criar uma nova página você abre o *template* e não precisa escrever todos estes marcadores. Todo o conteúdo da página deve ficar entre os marcadores <BODY> e </BODY> (com exceção dos *scripts*, que estudaremos mais a frente) que tem alguns atributos especiais:

- **background="imagem.gif"** - coloca uma imagem de fundo, igual ao *wallpaper* do *Windows*.
- **bgcolor="#000000"** - escolhe a cor de fundo da página

- `text="#000000"` - escolhe a cor do texto
- `link="#000000"` - escolhe a cor do texto de um *link*
- `vlink="#000000"` - escolhe a cor do texto de um *link* já visitado
- `alink="#000000"` - escolhe a cor do *link* que está ativo

Nota 1: As imagens utilizadas devem ter o formato de compactação GIF ou JPG.

Nota 2: As cores devem ser escritas em base hexadecimal.

Nota 3: Os dois primeiros dígitos são da cor vermelha, os outros dois da verde e os dois restantes da azul (RRGGBB – *Red/Green/Blue*).

Há um marcador bem especial que devemos considerar que serve para inserir comentários no código. Para inserir um comentário utilizamos o marcador `<!--` para iniciar e o `-->` para terminar. Tudo que estiver entre eles é um comentário e não irá aparecer na tela.

## 6.1.2. Títulos

Existem 6 estilos de título ou de texto que variam o tamanho da fonte. São eles:

<code>&lt;H1&gt;Título 1&lt;/H1&gt;</code>	<b>Título 1</b>
<code>&lt;H2&gt;Título 2&lt;/H2&gt;</code>	<b>Título 2</b>
<code>&lt;H3&gt;Título 3&lt;/H3&gt;</code>	<b>Título 3</b>
<code>&lt;H4&gt;Título 4&lt;/H4&gt;</code>	<b>Título 4</b>
<code>&lt;H5&gt;Título 5&lt;/H5&gt;</code>	<b>Título 5</b>
<code>&lt;H6&gt;Título 6&lt;/H6&gt;</code>	<b>Título 6</b>

## 6.1.3. Formatação do texto

Existem alguns marcadores que nos ajudam a formatar o texto.

<code>&lt;P&gt;Texto...&lt;/P&gt;</code>	Cria um parágrafo. Contém um atributo especial. <b>align</b> =["posição"] – alinha o texto de acordo com os seguintes valores: <b>left</b> - à esquerda <b>right</b> - à direita <b>center</b> – centralizado <b>justify</b> – justificado
<code>&lt;BR&gt;</code>	Força uma nova linha ou pula uma linha.
<code>&lt;PRE&gt;Texto...&lt;/PRE&gt;</code>	Mostra o texto na tela do jeito que foi escrito no documento conservando quebras de linha, parágrafos, tabulação, etc.
<code>&lt;B&gt;texto...&lt;/B&gt;</code>	<b>Negrito.</b>
<code>&lt;I&gt;texto...&lt;/I&gt;</code>	<i>Itálico.</i>
<code>&lt;U&gt;texto...&lt;/U&gt;</code>	<u>Sublinhado.</u>
<code>&lt;FONT&gt;Texto...&lt;/FONT&gt;</code>	Formatação da fonte. Contém alguns atributos especiais: <b>color</b> ="#ff5b3f" - cor da fonte <b>size</b> =[+] [-] [n] - Tamanho da fonte. (aqui está +1)

### 6.1.4. Imagens

<IMG src="imagem.gif">

Este marcador contém alguns atributos especiais:

- **border="n"** - coloca uma borda ao redor da figura.
- **height="n"** - modifica a altura da figura.
- **width="n"** - modifica a largura da figura.
- **alt="texto qualquer"** - mostra um texto quando o mouse parar em cima da figura.
- **align="posição"** - alinha a imagem na tela.

### 6.1.5. Listas

Podemos utilizar dois tipos de listas em HTML:

Listas ordenadas

<pre>&lt;OL&gt; &lt;LI&gt;Item 1&lt;/LI&gt; &lt;LI&gt;Item 2&lt;/LI&gt; &lt;LI&gt;Item 3&lt;/LI&gt; &lt;/OL&gt;</pre>	<pre>1. Item 1 2. Item 2 3. Item 3</pre>
-----------------------------------------------------------------------------------------------------------------------	------------------------------------------

Listas sem ordem

<pre>&lt;UL&gt; &lt;LI&gt;Item 1&lt;/LI&gt; &lt;LI&gt;Item 2&lt;/LI&gt; &lt;LI&gt;Item 3&lt;/LI&gt; &lt;/UL&gt;</pre>	<pre>• Item 1 • Item 2 • Item 3</pre>
-----------------------------------------------------------------------------------------------------------------------	---------------------------------------

Nota 4: Você pode criar uma sub-lista dentro de um item de uma lista.

### 6.1.6. Links

*Links* são atalhos, ligações para um documento ou para uma parte específica de um documento. Podem também executar tarefas como pegar um arquivo (local ou via ftp), ou abrir o seu editor de *e-mails* e deixá-lo pronto para enviar uma mensagem para alguém.

<A href="[local / ação]">Texto que aparece como link.</A>

[local / ação] podem ser um dos seguintes:

- "http://..." - endereço de uma página HTML ou arquivo.  
Ex.: <A href=http://mesonpi.cat.cbpf.br> Página do Servidor Mesonpi </A>  
Visualização: Página do Servidor Mesonpi
- "ftp://..." - endereço e nome de um arquivo num site ftp.  
Ex.: <A href=ftp://ftp.cat.cbpf.br/from/browsers/netcape/nc32c403.exe>  
Communicator 4.03 </A>

## Visualização: Communicator 4.03

- "mailto:nome@resto.do.email" - abre o editor de *e-mail* e deixa-o pronto para escrever uma mensagem.  
Ex.: <A href=mailto:nome@resto.do.email> nome@resto.do.email </A>  
Visualização: nome@resto.do.email

<A href="[local / ação]#ancora">Texto que aparece como link.</A>

A propriedade **#ancora** vai para uma parte específica de um documento HTML. É preciso colocar um marcador no documento do tipo <A name="ancora"> para funcionar. Se for no mesmo documento, não é preciso colocar [local / ação].

Ex.: <A href="#início">Vai para o início da página. </A>

Visualização: Vai para o início da página.

### 6.1.7. Tabelas

As tabelas além função de listar dados, são frequentemente utilizadas em HTML para formatar documentos que possuam textos e figuras. Abaixo, veja um exemplo típico de uma tabela abaixo:

```
<TABLE> <CAPTION align="bottom"> Tabela 1 </CAPTION>
<TH>Título da tabela </TH>
<TR>
  <TD>Primeira linha e primeira coluna. </TD>
  <TD>Primeira linha e segunda coluna. </TD>
</TR>
<TR>
  <TD>Segunda linha e primeira coluna. </TD>
  <TD><IMG src="sol.jpg" align="center"> </TD>
</TR>
</TABLE>
```

Título da tabela	
Primeira linha e primeira coluna	Primeira linha e segunda coluna
Segunda linha e primeira coluna	

Tabela 1

O marcador <TABLE> possui alguns atributos especiais:

- **border = "n"** - coloca borda na tabela
- **bgcolor = "#000000"** - escolhe a cor de fundo

Os outros marcadores relativos à tabela, como você já deve estar imaginando, também possuem atributos especiais:

- **caption** - legenda

- **align** = "posição" - usado para alinhar horizontalmente o texto. Pode ser usado dentro de <TR>, <TH> e <TD>. Os valores são **left**, **center** e **right**
- **valign** = "posição" - usado para alinhar verticalmente o texto. Pode ser usado dentro de <TR>, <TH> e <TD>. Os valores são **top**, **middle**, **bottom** e **baseline**.
- **colspan** = n - especifica a quantidade de colunas da tabela que uma célula deve se estender. Com isso conseguimos colocar 2 colunas em baixo de uma coluna. Pode ser usado dentro de <TD>.
- **rowspan** = n - especifica a quantidade de linhas da tabela que uma célula deve se estender. Com isso conseguimos colocar 2 linhas ao lado de uma linha. Pode ser usado dentro de <TR>.
- **colspec** = n - especifica a largura da coluna. Pode ser usado um valor explícito ou em porcentagem.

O melhor modo para visualizar os marcadores e seus parametros é vendo o exemplo de uma tabela um pouco mais complexa. Considere agora o seguinte exemplo.

```
<TABLE border=1><CAPTION align="bottom" Tabela exemplo></CAPTION >
<TR>
  <TD rowspan=2></TD> <TH colspan=2>Média</TH>
</TR>
<TR>
  <TH>Altura</TH>
  <TH>Peso</TH>
</TR>
<TR>
  <TD>Homens</TD>
  <TD align="center">69</TD>
  <TD align="center">1,75</TD>
</TR>
<TR>
  <TD>Mulheres</TD>
  <TD align="center">64</TD>
  <TD align="center">1,68</TD>
</TR>
</TABLE>
```

	Média	
	Altura	Peso
Homens	69	1,75
Mulheres	64	1,68

Tabela Exemplo

### 6.1.8. Formulários

Veremos abaixo os marcadores dos principais itens de um formulário. Observe que os diversos itens possíveis de um formulário encontram-se entre os marcadores <FORM> e </FORM>. Observe ainda que existem dois métodos para enviar os dados do formulário para o servidor Web. O método *get* acrescenta ao final do endereço URL, os pares *NAME-VALUE* de cada item do formulário, ocupando tudo 1024 caracteres no máximo. Já o método *post* é o mais utilizado pois transmite um maior volume de dados, visto que não tem a limitação de 1024 caracteres, por não ser transmitido pelo URL. Abaixo mostramos um pequeno exemplo do método *get* seguido do código HTML de um formulário.

[http://server-name/cgi-bin/program-name?texto1="bláblábláblá"&op1=true&op2=false...](http://server-name/cgi-bin/program-name?texto1=)

```
<FORM METHOD="get / post" ACTION="url">
```

```
Área de texto: <BR><TEXTAREA NAME="textol" ROWS=5 COLS=60></TEXTAREA><BR><BR>
```

```
Checkbox: <INPUT TYPE="checkbox" NAME="op1"><BR>
```

```
Radiobox: <INPUT TYPE="radio" NAME="op2"><BR>
```

```
Caixa de password: <INPUT TYPE="password" NAME="password" SIZE=7><BR>
```

```
Caixa de texto: <INPUT TYPE="text" NAME="texto2" MAXLENGTH=10><BR>
```

```
Botão de reset: <INPUT TYPE="reset" VALUE="RESET TO DEFAULT"><BR>
```

```
Botão de enviar: <INPUT TYPE="submit" VALUE="ENVIAR DADOS"><BR>
```

```
</FORM>
```

Área de texto:

Checkbox:

Radiobox:

Caixa de Password:

Caixa de texto:

Botão de Reset:

Botão de Enviar:

## 6.2. CGI em C

A *Common Gateway Interface* (CGI) é um padrão para interfacear aplicações externas com servidores de informação, tais como HTTP e servidores Web [19]. Um documento HTML é estático, o que significa que ele existe em estado constante, ou seja, é apenas um arquivo texto que não muda. Um programa CGI, ao contrário, é executado em tempo real, de tal forma que sua saída é uma informação dinâmica.

Vamos considerar que alguém precise tornar acessível uma base de dados UNIX através do ambiente Web, para que pessoas distantes possam utiliza-la. Basicamente é necessário criar um programa CGI que o *daemon* Web execute e transmita a informação ao servidor da base de dados e receba de volta ao cliente a resposta à sua requisição (*query*). Isto é um exemplo de um *gateway*, e é também onde CGI 1.1 teve sua origem.

Quando utilizamos o termo *gateway*, estamos nos referindo a programas que guiam as requisições de informação e retornam ou geram documentos. No caso de programas CGI, o servidor pode fornecer informações que não estejam na forma visual para o cliente, tal como um banco de dados SQL, e atua como uma ponte entre o cliente e o servidor, produzindo algum elemento que pode ser utilizado pelo usuário.

O exemplo da base de dados é bem simples apesar de quase sempre ser difícil a sua implementação. Na realidade não existe limites para um programa CGI, porém deve ser sempre lembrado que o programa não deve tomar muito tempo de processamento, pois o usuário estará esperando pela resposta da sua requisição.

Os programas CGI podem ser escritos em qualquer linguagem como por exemplo: C/C++, PERL, TCL, The Bourne Shell, The C Shell, FORTRAN, VISUAL BASIC, AppleScript, etc. Para aqueles que querem utilizar a linguagem C, existem alguns programas fornecidos pelo

CERN ou pela NCSA, que permitem a comunicação com o servidor. Um destes programas se chama *post-query.c*. Este programa para ser utilizado deve ser compilado ao lado do *util.c*, também disponível no CERN ou na NCSA.

## 7. Introdução à Linguagem JavaScript

Esta parte do curso tem como objetivo realizar uma introdução à linguagem JavaScript. Optou-se por esta linguagem por ser mais fácil de se trabalhar, visto que, o código do *script* é inserido diretamente no código HTML, não precisa ser compilado e nem de editores específicos para criá-lo, ao contrário da linguagem Java. Lembre-se que JavaScript é diferente de Java. A tabela a seguir faz uma comparação entre as características principais de Java e JavaScript.

LINGUAGEM JAVASCRIPT	LINGUAGEM JAVA
Interpretada pelo cliente (no caso o navegador) na hora em que a página é carregada.	Compilada pelo programador antes do cliente carregar a página.
Baseada em objetos. A linguagem utiliza um conjunto de objetos, mas não permite a criação de classes e a herança.	Orientada à objeto. Os <i>applets</i> consistem de instâncias de classe (objetos) que possuem herança.
O <i>script</i> fica inserido no código HTML.	Os <i>applets</i> ficam separados do código HTML e são acessados por ele através de <i>tags</i> .
Não precisa declarar o tipo das variáveis utilizadas (menos digitação).	Precisa declarar o tipo de todas as variáveis utilizadas (mais digitação).
Ligação dinâmica. A referência aos objetos é checada na hora em que a página é carregada.	Ligação estática. A referência aos objetos tem que existir na hora em que carregar a página.
Não podem escrever no disco rígido automaticamente.	Podem escrever no disco rígido, desde que instruídos.

Podemos dizer que Java é uma linguagem de programação, pois você pode realmente fazer programas em Java, enquanto que JavaScript é, digamos, uma extensão do HTML.

Para começar a escrever *scripts* você tem que conhecer algumas técnicas básicas e algumas funções internas importantes. Vamos ver agora algumas características básicas para escrever *scripts*.

### 7.1. Aonde inserir o *script*?

O *script* tem que ficar inserido entre os marcadores `<SCRIPT>` e `</SCRIPT>`. Estes marcadores informam ao navegador que o seu conteúdo é um *script*. Existem outros tipos de *script* além do JavaScript, como por exemplo o Perl Script. Um item opcional é o atributo *LANGUAGE*, que especifica qual a linguagem do *script* que está inserido entre estes marcadores. O marcador ficaria assim:

Exemplo 1:

```
<SCRIPT LANGUAGE = "JavaScript">
    aqui coloca-se o código do script...
</SCRIPT>
```

Este marcador do *script* deve ficar inserido dentro dos marcadores `<HEAD>` e `</HEAD>`. Visto que o código do JavaScript é executado na hora em que a página é carregada (*run-time system*), isto garante que o código seja carregado antes que o usuário possa ter algum tipo de interação com a página.

Os navegadores mais antigos não conseguem interpretar JavaScript. Eles desconhecem as marcadores `<SCRIPT>` e `</SCRIPT>`, e com isso interpretam o código do *script* como se fosse um texto de HTML comum. Isso faz com que o código apareça todo na tela estragando a apresentação da página. Para que isto não ocorra, deve-se “esconder” o *script* entre os marcadores `<!--` e `//-->`, pois os navegadores antigos ignoram o que está entre eles.

**Dicas:** Um bom programador deve escrever seu código inserindo comentários para facilitar posterior manutenção ou implementação do código pelo próprio autor ou outro programador. Quando quiser inserir um comentário, utilize os marcadores `//`. Um outro fator importante para se escrever um código legível é a tabulação. Procure se preocupar com ela e manter um padrão, para que seu código possa ser lido e compreendido com maior clareza.

Considere o exemplo a seguir.

Exemplo 2:

```
<HTML>
<HEAD> <TITLE>Exemplo 2:</TITLE>

<SCRIPT LANGUAGE = “JavaScript”>
<!-- Esconde o script dos navegadores antigos
    É bom declarar as funções aqui! (veremos como declarar funções no item 7.4.)
    // isto é um comentário. O marcador a seguir termina código escondido. -->
</SCRIPT>
</HEAD>

<BODY>

<SCRIPT>
<!-- Esconde o script dos navegadores antigos
document.write(“Aqui é JavaScript.”);
// -->
</SCRIPT>

Aqui é texto em HTML.

</BODY>
</HTML>
```

Neste exemplo, utilizamos uma função interna do JavaScript: *document.write()*. Esta função serve para mostrar na tela o que quisermos, desde uma *string* qualquer até o conteúdo de variáveis. Esta página aparecerá na tela de um navegador que suporta JavaScript assim:

```
Aqui é JavaScript.
Aqui é texto em HTML.
```

Se não for inserido os marcadores `<!--` e `//-->`, a página apareceria assim:

```
document.write(“Aqui é JavaScript.”);
Aqui é texto em HTML.
```

Agora que já sabemos aonde colocar o código, como colocá-lo e como “escondê-lo”, podemos passar adiante.

## 7.2. Considerações iniciais

Nesta parte do capítulo veremos um resumo de todos os recursos mais importantes e úteis de JavaScript. Com eles, você poderá criar *scripts* simples para usar em páginas pessoais, por exemplo. No final do trabalho, existem referências nas quais pode-se procurar por maiores informações sobre JavaScript.

### 7.2.1 Variáveis

As variáveis abrangem dois níveis: global e local. As variáveis globais são declaradas fora de uma função e podem ser acessadas de qualquer parte do código. Já as locais, são criadas dentro das funções utilizando a instrução *var* (que veremos mais a frente no item 7.2.4.) e só podem ser utilizadas dentro da função onde foram criadas. São três os tipos de variáveis: numéricas, booleanas e *strings*. As numéricas, como o próprio nome diz, guarda números, as booleanas guardam valores lógicos (*true / false*) e as *strings* guardam uma seqüência de caracteres. Existe também um outro tipo de conteúdo em variáveis que é o *NULL*. Este valor não é igual a nada, e é utilizado para manipular variáveis não inicializadas sem que ocorra um erro no programa. No que diz respeito a nomenclatura, as variáveis devem começar com letra minúscula ou com o caracter “\_”, o restante pode conter qualquer letra (maiúscula ou minúscula) ou número.

### 7.2.2. Operadores

Os operadores são blocos de construção de expressões. São comparáveis a uma função no sentido que executam uma operação e retornam o valor dessa operação.

Operador de *string*: opera com *strings*. Ex.:

```
nome1 = "João"
nome2 = "Ferreira"
nome = nome1 + " Marques " + nome2
O resultado disso é: "João Marques Ferreira".
```

Operador Matemático: utiliza os sinais + (adição), - (subtração), \* (multiplicação), / (divisão), % (resto da divisão), ++ (incremento) e -- (decremento) para efetuar operações matemáticas. Ex.:

Valor1 = 10 Valor2 = 5 Valor3 = valor1 * valor2 O resultado é: 50.	valor1 = 10 valor2 = 5 valor3 = valor1 % valor2 O resultado é: 0.
Valor1 = 7 Valor2 = ++valor1 O resultado é: 8.	valor1 = 7 valor2 = --valor1 O resultado é: 6.

**Tabela 2: Exemplos de operadores lógicos**

Obs.: Os operadores lógicos de incremento podem ser escritos tanto na forma **++valor** ou **valor++**. O mesmo serve para os de decremento.

Operadores relacionais: comparam valores. São eles:

<	Menor que
>	Maior que
==	Igual
!=	Diferente
>=	Maior ou igual
<=	Menor ou igual

Tabela 3: Operadores relacionais

Operadores lógicos: executam as condições lógicas “E” e “OU”.

&&	E lógico
	OU lógico

Tabela 4: Operadores lógicos

Operadores de atribuição: atribuem valores as variáveis.

=	Atribuir
+=	Soma e atribuição. $x = x + 5$ é igual a $x += 5$
-=	Subtração e atribuição. $x = x - 5$ é igual a $x -= 5$
*=	Multiplicação e atribuição. $x = x * 5$ é igual a $x *= 5$
/=	Divisão e atribuição. $x = x / 5$ é igual a $x /= 5$
%=	Resto e atribuição. $x = x \% 5$ é igual a $x \% = 5$

Tabela 5: Operadores de atribuição

### 7.2.3. Estruturas de controle

Veremos agora uma parte interessante da linguagem JavaScript: as estruturas de controle. Através delas pode-se alterar a ordem de execução do código ou proporcionar um melhor nível de interatividade com a página. Vamos ver quais são estas estruturas:

Estrutura	Sintaxe	Comentários
<i>if... else</i>	<i>if</i> (condição) { declarações... } <i>else</i> { declarações }	Executa as declarações dentro do <i>if</i> se a condição for verdadeira; se for falsa, executa as do <i>else</i> .
<i>for ... in</i>	<i>for</i> (variável In objeto) { declarações... }	Se a variável dentro das propriedades de um objeto existe, executa as declarações.
<i>for</i>	<i>for</i> ( [expressão inicial;] [condição;] [expressão de incremento;] ) { declarações... }	Executa um <i>loop</i> até uma condição passar a ser falsa. É parecido com o <i>loop</i> da linguagem C.
<i>while</i>	<i>while</i> (condição) { declarações... }	Executa um <i>loop</i> até a condição ser verdadeira.

Tabela 6: Estruturas de controle

## 7.2.4. Comandos

**Var:** Este comando é mais utilizado para definir variáveis de nível local, ou seja, variáveis dentro de funções. Para declarar uma variável global, basta atribuir um valor a ela.

Variável Global	Variável Local
total = 0	var total = 0

Tabela 7: Comando *var*

**With:** Este comando permite a você trabalhar com várias propriedades de um objeto sem ter que digitar várias vezes o nome do objeto.

Sintaxe	Exemplo
with (objeto) { declarações... }	with (Math) { a = PI; b = sin(2.53) }

Tabela 8: Comando *with*

**Break:** Só pode ser utilizado dentro de *loops while* ou *for*. Este comando cancela a execução do *looping* sem verificar a condição de saída.

Ex.:

<pre>for (var x = 1; x &lt; 10; x++) {   if (x == 5) {     break   } } document.write(x)</pre>	<p>Este exemplo retorna o valor 5, pois x vai sendo incrementado de 1 unidade cada vez que o <i>loop</i> é executado e quando x é igual a 5, ele sai do <i>loop</i>.</p>
------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabela 9: Comando *break*

**Continue:** Só pode ser utilizado dentro de *loops while* ou *for*. Este comando cancela a execução do bloco de comandos, retornado ao início do *loop*.

Ex.:

<pre>for (var x = 1; x &lt; 10; x++) {   if (x == 5) {     Continue   }   document.write(x + " ") }</pre>	<p>Este exemplo imprime os números de 1 a 9, com exceção do número 5. Note que a instrução <i>document.write</i> encontra-se dentro da função. Se estivesse fora, como no exemplo anterior, apenas o último valor atribuído a x seria impresso.</p>
-----------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabela 10: Comando *continue*

O que acabamos de ver é a parte bem básica de JavaScript. As variáveis, os operadores, os comandos, tudo isso será usado para escrever blocos de comandos complexos, aos quais chamamos de funções. Os objetos e as funções são as partes mais importantes em JavaScript. Veremos agora como usar os objetos, para depois estudarmos como declarar (criar) as funções.

## 7.3. Objetos

O que o JavaScript considera objeto, em outras linguagens é chamado de estrutura de dados ou classes. Como JavaScript é baseada em objetos, precisamos saber como utilizá-los e como nos referir-mos a eles.

A linguagem JavaScript possui objetos já definidos prontos para serem utilizados. Eles estão divididos em 3 grupos:

1. os HTML
2. os internos
3. os do navegador

### 7.3.1. Objetos HTML

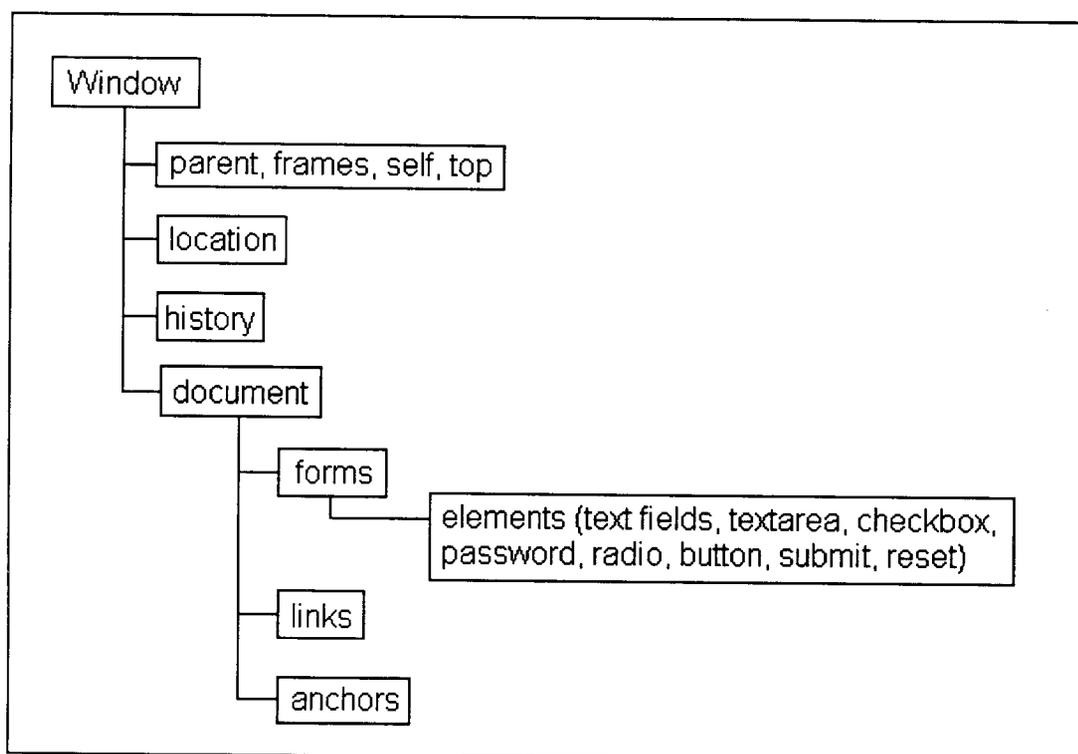


Figura 1: Hierarquia no JavaScript

Os elementos dentro de uma página HTML possuem uma hierarquia, como mostra a figura a seguir.

Os elementos são acessados como objetos HTML em JavaScript, que podem ter propriedades, métodos e responder a certos eventos (através de *event handlers*, que veremos no item 7.5.). Todos os objetos de HTML são propriedades do objeto *document*. Para o JavaScript, a janela do navegador é o objeto chamado *window*. Este objeto está no topo da hierarquia. Considere agora a figura a seguir de uma página HTML comum:

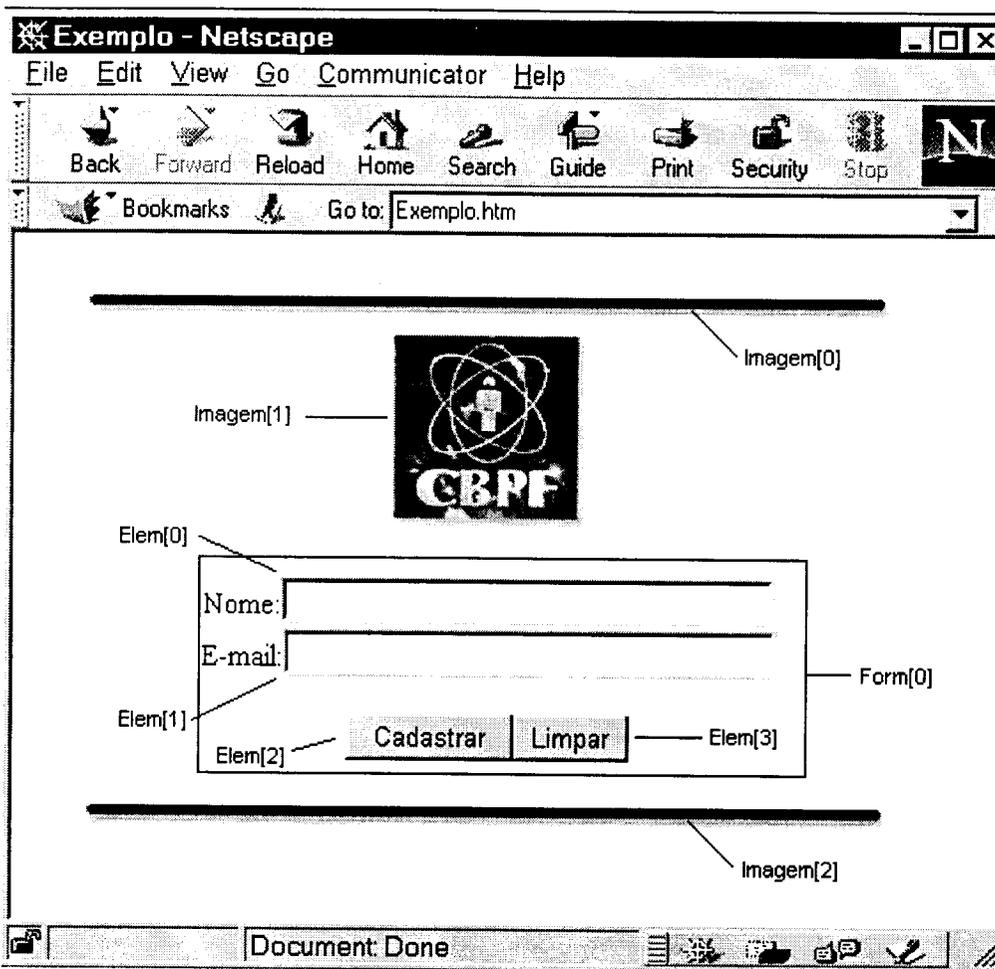


Figura 2: Página HTML comum

Esta página é um objeto *document*, ou seja, este objeto representa a página que está carregada no momento. As imagens e o formulário são propriedades deste *document*. Este objeto *document* é muito importante pois será muito usado em JavaScript. As propriedades deste objeto serão vistas mais adiante. O diagrama a seguir mostra a hierarquia do exemplo acima.

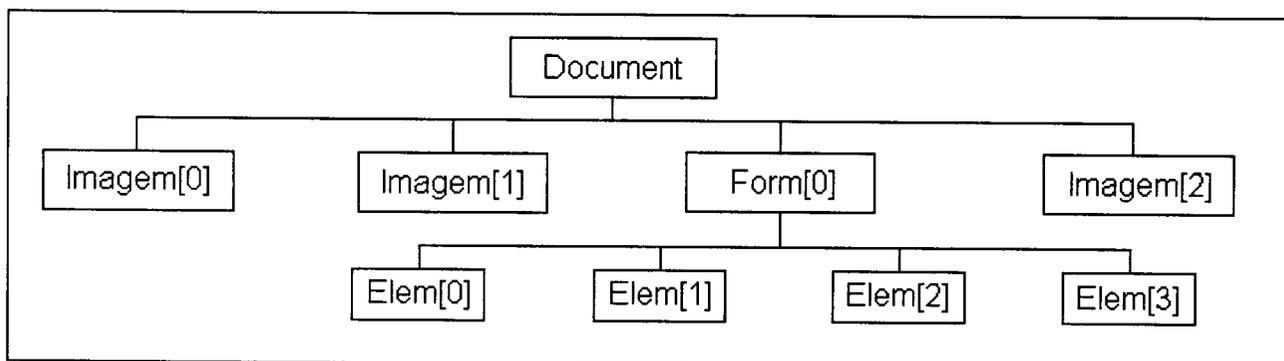


Figura 3: Diagrama de Hierarquia

Este diagrama é muito útil quando quisermos nos referir à algum objeto, pois para isso devemos especificar o nome dele e de todos os seus antecessores. Veremos agora um método rápido e simples para fazer isso. Primeiro verifica-se o nome do objeto no diagrama e depois percorre-se o diagrama de baixo para cima até chegar no objeto *document*. Deste modo ficamos sabendo quem são os antecessores do objeto em questão. Para nos referirmos ao formulário, por exemplo, devemos usar a seguinte sintaxe: `document.form[0]`. Se quisermos nos referir ao botão “Cadastrar” do formulário, devemos usar: `document.form[0].elem[2]`, e assim por diante.

Quando quisermos nos referir ao conteúdo de um objeto, como por exemplo o texto que foi digitado pelo usuário dentro da primeira caixa de diálogo *elem[0]*, devemos usar uma propriedade chamada *value*. Logo, para acessarmos o que foi digitado dentro da primeira caixa devemos usar a sintaxe *nome = document.form[0].elem[0].value;*. Desse modo, o conteúdo do *elem[0]*, que é uma *string*, será armazenado na variável *nome*. Para ver o seu valor na tela podemos usar uma sintaxe já vista antes: *document.write(nome);*.

Para evitar problemas ao trabalhar com páginas muito grandes, podemos dar nomes aos objetos. Vejamos o seguinte código:

```
<FORM name=agenda>
<INPUT type="text" name="contato">
...
</FORM>
```

Ao invés de usarmos *document.form[0].elem[0].value* podemos usar os nomes: *document.agenda.contato.value*. Isto é muito útil ao trabalhar com páginas grandes e muitos objetos.

### 7.3.2 Objetos internos.

Objeto Date: Este objeto permite a você trabalhar com datas e horas. Possui muitos métodos para manipular os dados. Vamos ver algumas maneiras para utilizar este objeto.

OBS.: Você só pode trabalhar com datas a partir de 1/1/70.

Para criar um novo objeto data:

*var Nome = new Date(parâmetros)*, onde *Nome* é o nome da variável e *parâmetros* pode ser um dos seguintes:

- Nenhum. Usa a data e a hora atual. ( *today = new Date()* )
- Uma *string* representando uma data no seguinte formato: "dia do mês, ano hora:minutos:segundos". ( *today = new Date("January 1, 98 12:00:00")* )
- Valores inteiros para ano, mês e dia. ( *today = new Date(98, 1, 1)* )
- Valores inteiros para ano, mês, dia, hora, minutos e segundos. ( *today = new Date(98, 1, 1, 12, 30, 00)* )

OBS.: A hora deve ser especificada no formato 24 Hs. Os meses vão de 0 a 11, inclusive.

Métodos do Date object			
getDate ()	getDay ()	getHours ()	getMinutes ()
getMonth ()	getSeconds ()	getTime ()	getTimeZoneOffset ()
getYear ()	setDate ()	setHours ()	setMinutes ()
setMonth ()	setSeconds ()	setTime ()	setYear ()

Tabela11: Métodos do Date object

Objeto Math: Este objeto contém propriedades e métodos para constantes e funções matemáticas, como por exemplo, funções trigonométricas, logarítmicas e exponenciais. As tabelas a seguir contém as principais propriedades e métodos deste objeto.

Propriedades	O que faz
E	constante de Napier
LN10	logaritmo natural de 10
LN2	logaritmo natural de 2
PI	valor de PI

SQRT1_2	raiz quadrada de ½
SQRT2	raiz quadrada de 2

Tabela 12: Propriedades do objeto *Math*

Método	O que faz
abs( num )	retorna o valor absoluto do argumento
acos( num )	calcula arco coseno
asin( num )	calcula arco seno
atan( num )	calcula arco tangente
ceil( num )	usa um número ponto-flutuante e retorna o menor inteiro maior ou igual ao argumento
cos( ang )	calcula coseno
exp( num )	calcula exponencial
floor( num )	usa um número ponto-flutuante e retorna o maior inteiro menor ou igual ao argumento
log( num )	calcula logaritmo
max( num1, num2 )	retorna o valor mínimo e máximo respectivamente
random()	retorna qualquer número ponto-flutuante entre 0 e 1
round( num )	usa um número ponto-flutuante e retorna o inteiro mais próximo
sin( ang )	calcula seno
sqrt( num )	calcula raiz Quadrada
tan( ang )	calcula tangente

Tabela 13: Métodos do objeto *Math*

OBS.: Todas as funções trigonométricas utilizam parametros com valores em radianos.

Nota: Ao escrever *Math* no seu *script*, não esqueça de colocar a letra “m” em maiúsculo, caso contrário, não vai funcionar.

Exemplo:

Math.PI – retorna o valor de Pi.

Math.sin(1.56) – retorna o valor do seno de 1,56 radianos.

Objeto *String*: Este objeto tem apenas uma propriedade: *length* (o tamanho), e alguns métodos. A propriedade *length* retorna o tamanho da *string*. Os métodos são para manipular o conteúdo e a aparência

Principais métodos para manipular o conteúdo das *strings*:

Método	O que faz
charAt( x )	retorna o caracter na posição “x” da esquerda para direita. (seguindo a ordem 0,1,2...)
indexOf( “chr” )	retorna a posição do caracter “chr”, da esquerda para direita.
lastIndexOf( “chr” )	retorna a posição do caracter “chr”, da direita para esquerda. (seguindo a ordem 0,1,2...)
substring( dex, atex )	retorna uma substring da posição “dex” até “atex”
toLowerCase()	converte os caracteres para “caixa baixa”.
toUpperCase()	converte os caracteres para “CAIXA ALTA”.

Tabela 14: Métodos para manipular o conteúdo das *Strings*

Exemplos:

str = "Escola de Verao 98"	
str.charAt( 5 )	retorna "a"
str.indexOf( "d" )	retorna "7"
str.lastIndexOf( "e" )	retorna "11"
str.substring( 6, 9 )	retorna "de" pois não inclui o caracter na posição 9.
str.toLowerCase()	retorna "escola de verao 98"
str.toUpperCase()	retorna "ESCOLA DE VERA0 98"

Tabela 15: Exemplos de métodos para manipular o conteúdo das Strings

Principais métodos para manipular a aparência das *strings*:

Método	O que faz
big()	escreve a <i>string</i> com fonte grande
blink()	escreve a <i>string</i> piscando
bold()	escreve a <i>string</i> em negrito
fontcolor( cor )	muda a cor da fonte da <i>string</i>
fontsize( tam )	muda o tamanho da fonte da <i>string</i>
italics()	escreve a <i>string</i> em itálico
small()	escreve a <i>string</i> com fonte pequena
strike()	escreve a <i>string</i> sublinhada
sub()	escreve a <i>string</i> no modo sobrescrito
sup()	escreve a <i>string</i> no modo superescrito

Tabela 16: Métodos para manipular a aparência das *strings*

### 7.3.3. Objetos do navegador

De acordo com a figura 1, os objetos do navegador são: *window*, *document*, *location* e *history*. Os métodos do objeto *window* nos permitem trabalhar com as janelas dos navegadores.

- **window.alert('str')** - mostra uma janela com uma *string* ('str') de alerta.
- **window.close()** - fecha uma janela.
- **window.confirm('str')** - mostra uma janela de confirmação contendo uma *String* ('str') e botões de OK e Cancel. (OK assume valor lógico *true* e cancel *false*).
- **window.open('url', 'wname')** - abre uma nova janela do navegador com um novo documento, onde '*wname*' é o nome da janela.
- **window.prompt('str')** - abre uma janela com uma *string* ('str'), uma caixa de texto e com os botões *OK* e *Cancel*. Usada para pedir algum tipo de informação ao usuário.

Veja a tabela a seguir com os principais métodos de cada objeto.

Objetos				
	window	document	location	history
Métodos	alert( str )	clear()	back()	back()
	close()	close()	forward()	forward()
	confirm( str )	open()	go( where )	go( where )
	open( url, wname )	write( str )		
	prompt( str )	writeln( str )		

Tabela 17: Objetos do navegador e seus métodos

Os métodos do objeto *document* nos permitem trabalhar com a página em si.

- **document.clear()** - limpa um documento
- **document.close()** - fecha um documento
- **document.open()** - abre um novo documento
- **document.write('str')** - escreve na tela uma *string* ('str').
- **document.writeln('str')** - escreve na tela uma *string* ('str') e força uma nova linha.

## 7.4. Funções

As funções possuem basicamente dois propósitos. Primeiro: são uma poderosa ferramenta de organização, no sentido em que o código pode ser escrito em blocos nomeados, cada um com sua tarefa específica. Segundo: ligar ações do tipo apertar botões, selecionar uma *checkbox*, cliques do mouse, entre outras com o código JavaScript, através de alguns marcadores especiais chamados *event handlers*.

Todas as funções em JavaScript precisam ser declaradas, isto é, montadas. Dentro de uma função pode existir uma chamada para outra função ou até para ela mesma. Veja a sintaxe da declaração de uma função:

```
function Nome ( parâmetro 1, parâmetro 2, ... parâmetro n ) {
  declaração
}
```

1. Escrevemos a palavra *function* (sempre com o “f” minúsculo!);
2. Nome é o nome que queremos dar a função;
  - Restrições ao nome:
    - a) não pode conter espaços em branco;
    - b) não é bom ter o mesmo nome de variáveis;
    - c) não deve conflitar com as palavras reservadas, ou seja, ter o mesmo nome de comandos, funções internas, estruturas de controle, etc. (no final da apostila há uma tabela com todas as palavras reservadas).
3. parâmetros são os valores passados para função (pode-se não colocar nada);
4. declaração é aonde utilizamos os comandos básicos para executar uma tarefa qualquer.

Veja o exemplo a seguir:

```
function print_string(texto) {
  document.write("<HR><H6>" + texto + "</H6>")
}
```

Esta função mostra na tela uma linha horizontal, usa o operador de concatenação (+), utiliza a *string* chamada “texto” como parâmetro e mostra ela na tela utilizando o estilo de título 6.

Como já foi visto antes, todas as funções devem ser declaradas dentro dos marcadores <HEAD> e </HEAD>, pois esta parte da página é carregada primeiro, reconhecendo assim todas as funções existentes. Depois da função estar declarada, você precisa fazer uma chamada a ela, para poder executá-la. Existem diferentes meios para chamar uma função. Pode ser através de eventos acionados pelo usuário ou com uma chamada simples. Para isso, coloque no seu código HTML no lugar em que você quer que o valor retornado pela função apareça, uma chamada do tipo:

```
<SCRIPT>
print_string("Centro Brasileiro de Pesquisas Físicas")
</SCRIPT>
```

Deste modo você chama a função e passa para ela a *string* "Centro Brasileiro de Pesquisas Físicas" como parâmetro. Os parâmetros não estão limitados a *strings* e números, podendo ser também objetos. Os argumentos que são passados para a função ficam armazenados em forma de *array*. Você pode acessar os argumentos de uma função do seguinte modo:

*functionName.arguments[i]*, onde *functionName* é o nome da função e *i* é o número do argumento, começando do 0 que é o número do primeiro argumento.

Para saber o número total de argumentos passados a uma função, utilize a variável *arguments.length*, precedida de *functionName*, claro.

### 7.4.1. Funções internas

O JavaScript possui também funções internas, que inicialmente não estão associadas a nenhuma variável ou objeto. Não precisam ser declaradas. Vejamos algumas delas:

- **Escape** – obtém o código ASCII de um caracter que não seja alfa numérico.  
Ex.: `document.write(escape("@"))`
- **Eval** – esta função toma uma *string* contendo uma expressão aritmética ou uma declaração como argumento e retorna o valor desta expressão  
Ex.: `valor = eval(10 * 3)`
- **parseFloat** – Converte uma *string* que representa um número, para um número com ponto flutuante. Se a *string* não puder ser validade, retorna 0.  
Ex.: `document.write(parseFloat("-32.465e12"))`
- **parseInt** – esta função converte a *string* de uma base predefinida para base 10.  
Ex.: `parseInt("3A", 10) // retorna 3`

## 7.5. Event Handlers

Os eventos são ações que ocorrem normalmente como resultado de alguma ação feita pelo usuário, como por exemplo, apertar um botão, clicar em um link, etc. Os *event handlers* são *scripts* que são executados quando ocorre um evento. Eles ficam no código HTML como atributos dos marcadores os quais você associa o código JavaScript a ser executado. Complicado? Nem tanto. Veja o exemplo a seguir:

```
<FORM>
Nome: <INPUT type="text" name="nome" size="30">
E-mail: <INPUT type="text" name="e-mail" size="30">
<INPUT TYPE="button" value="Enviar" onClick="enviar(this.form)">
<INPUT TYPE="reset" value="Reset">
</FORM>
```

Este é um formulário simples com 2 caixas de texto (uma para nome e uma para e-mail) e 2 botões (Enviar e Reset). O *event handler* é o que está em negrito, ou seja, *onClick*. Como

podemos ver, ele está como um atributo do marcador `<INPUT type="button"...>`. Este *event handler* está associado a função “enviar”, que utiliza o objeto formulário como parâmetro. Assim, a função é executada após o botão ser acionado. Podemos colocar qualquer declaração dentro das aspas que não seja uma chamada para alguma função. Podemos também utilizar mais de uma declaração, separando-as por (;). Note que o uso do *this* em *this.form* é para se referir a este formulário. Veja na tabela a seguir alguns dos *event handlers* mais importantes.

Evento	Ocorre quando...	Event Handler
<i>click</i>	usuário clica em um <i>link</i> ou elemento do formulário	onClick
<i>change</i>	usuário troca o valor de um texto ou item selecionado	onChange
<i>load</i>	usuário carrega a página no navegador	onLoad
<i>mouseover</i>	usuário passa o ponteiro do <i>mouse</i> sobre um <i>link</i> ou ancora	onMouseOver
<i>select</i>	usuário seleciona um campo de entrada de dados num formulário	onSelect
<i>submit</i>	usuário envia um formulário	onSubmit
<i>unload</i>	usuário sai da página	onUnload

Tabela 18: Event Handlers

Tabela de palavras reservadas:

abstract	boolean	break	byte	case
cath	char	class	const	continue
default	do	double	else	extends
false	final	finally	float	for
function	goto	if	implements	import
in	instanceof	int	interface	long
native	new	null	package	private
protected	public	return	short	static
super	switch	synchronized	this	throw
Throws	transient	true	try	var
Void	while	with		

Tabela 19: Palavras reservadas

Agora nós já dominamos a parte básica da linguagem JavaScript. Veremos a seguir o código-fonte de uma página HTML simples, que utiliza JavaScript para calcular as raízes quadradas de uma equação do segundo grau.

## 7.6 Exemplo de Aplicação em JavaScript

Este é um exemplo de uma aplicação com JavaScript. O código a seguir é de uma página HTML que calcula as raízes de uma equação do segundo grau. Lembramos que a forma da equação do segundo grau é:  $Ax^2 + Bx + C = 0$ . A página contém um formulário com sete caixas de texto sendo três para os valores de entrada dos parâmetros A, B e C e quatro outros para os valores de delta, das duas raízes e de comentários. A página contém também três botões sendo um para calcular o valor de delta, um outro para apagar os valores de todas as caixas de texto e um último para calcular as raízes. O programa funciona da seguinte maneira: o usuário digita os valores de A, B e C nos respectivos lugares, aperta o botão “Calcular delta” e depois aperta o botão “Calcular raízes” para obter o resultado. Eis o código:

```
<HTML>
<HEAD>
<TITLE> Exemplo de aplicação com JavaScript</TITLE>
```

```

<SCRIPT>
<!--

function compute(form) {
    form.delta.value = (form.b.value * form.b.value) - 4 * form.a.value * form.c.value // Esta função calcula o valor de delta

    if (form.delta.value < 0) // Verifica se delta < 0
        form.results.value = ("Não existem raízes reais para esta equação.")

    if (form.delta.value == 0) // Verifica se delta = 0
        form.results.value = ("Existe apenas uma raiz real para esta equação.")

    if (form.delta.value > 0) // Verifica se delta > 0
        form.results.value = ("Existem duas raízes reais para esta equação.")
    }

function raizes(form) {
    if (form.delta.value < 0) // Esta função calcula o valor das raízes
        // Verifica de delta < 0
        window.alert("Não há raízes reais para calcular!") // Se menor mostra uma janela de alerta

    form.x1.value = ((-1 * form.b.value) + Math.sqrt(form.delta.value)) / (2 * form.a.value) // Valor de X1
    form.x2.value = ((-1 * form.b.value) - Math.sqrt(form.delta.value)) / (2 * form.a.value) // Valor de X2
    }

//-->
</SCRIPT>
</HEAD>

<BODY bgcolor="#000000" text="#ffffff">

<P>Este programa irá dizer se uma equação de segundo grau tem ou não raízes e se tiver
dirá quais são elas.</P>

<P>A equação do segundo grau tem a seguinte forma:  $Ax^2+Bx+C=0$ </P>
<P>Digite os valores desejados para A, B, C. (Se não digitar nenhum valor, será assumido valor zero para a
variável.)</P>

<CENTER>
<FORM>
Valor de A: <INPUT type="text" name="a" size=3 maxlength=3>
Valor de B: <INPUT type="text" name="b" size=3 maxlength=3>
Valor de C: <INPUT type="text" name="c" size=3 maxlength=3><BR><BR>
<INPUT type="button" value="Calcular delta" onClick=compute(this.form)>
// O marcador acima chama a função compute e passa este formulário (this.form) como parâmetro.

<INPUT type="reset"><BR><BR>
</CENTER>

Delta = <INPUT type="text" name="delta" size=10>
Obs: <INPUT type="text" name="results" size=50><br><br>
<INPUT type="button" value="Calcular raízes" onClick=raizes(this.form)><BR>
<!-- O marcador acima chama a função raizes que as calcula. -->

X1 = <INPUT type="text" name="x1" size=10><BR>
X2 = <INPUT type="text" name="x2" size=10><BR>
</FORM>

</BODY>
</HTML>

```

## 8. Videoconferência

Neste capítulo abordaremos o assunto mais recente na Internet e que apesar de ainda estar em sua fase inicial, de definição de padrões e conceitos e de desenvolvimentos de aplicativos, revela ter um potencial muito grande.

Podemos definir videoconferência como sendo uma interação entre várias pessoas em localidades diferentes utilizando os recursos de multimídia para a comunicação. Com os novos conceitos trazidos pela globalização, o serviço de videoconferência passa a ser uma das principais ferramentas para a interação remota. Antes de comentar os Princípios Básicos vamos abordar algumas Aplicações e os tipos de Mídia.

Citamos abaixo algumas aplicações nas quais poderíamos empregar este recurso:

- a) Colaboração entre pesquisadores de diversos centros de pesquisa e universidades: hoje já se imagina a utilização do recurso de videoconferência sem a necessidade de se deslocar pesquisadores e professores para a troca de experiência.
- b) Ensino à distância: em um país de grandes dimensões como o Brasil, poderíamos difundir o conhecimento não de uma forma unidirecional e passiva como é a forma da televisão ou do rádio mas permitindo uma maior interação entre alunos e professores. Com este recurso, poderíamos levar os melhores professores aos pontos mais distantes do nosso país através de cursos regulares.
- c) Informações sobre demanda: através de servidores multimídia, utilizando a mesma tecnologia da videoconferência, poderíamos criar grandes bancos de informações contendo áudio e vídeo inseridos em reportagens, procedimentos, conferências, shows que poderiam ser acessados de qualquer local e quantas vezes for necessário, simultaneamente ou não.
- d) Observação remota: permite que pessoas possam observar outros ambientes no mundo inteiro.

Numa videoconferência podemos encontrar basicamente três tipos de Mídia: Documentos, Áudio e Vídeo. Cada uma com suas próprias características e prioridades na transmissão. Podemos destacar alguns pontos importantes em cada uma delas:

*Documentos* - Esta mídia pode ser dividida em dois tipos: Documentos no formato Texto e Documentos no formato Imagem. O primeiro geralmente pequeno e o segundo com tamanhos maiores. As técnicas de compressão e de codificação já são bem conhecidas e utilizadas: JPEG, GIF, RLE, etc. Uma característica a ser ressaltada é que os dados a serem digitalizados são estáticos, isto é, caso haja algum erro na transmissão, estes podem ser enviados quantas vezes for necessário. O tráfego gerado é geralmente de forma intensa e rápida.

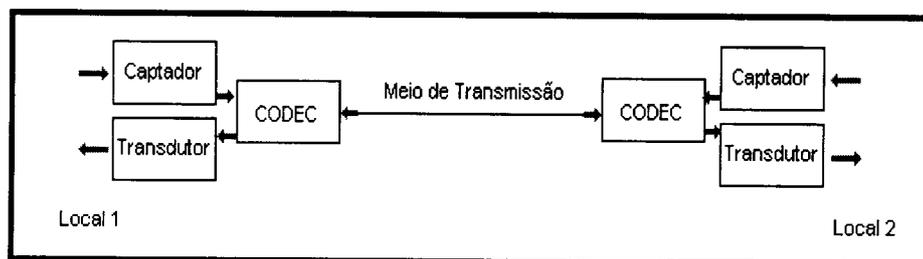
*Áudio* - a mídia de áudio se caracteriza por gerar um tráfego contínuo com taxa constante. Para se ter um entendimento correto do sinal enviado é necessário que se garanta fluxo constante de dados. Para tal, é necessário métodos de compressão e codificação tais como: PCM/ADPCM/LPC, CELP/GSM e outros.

*Vídeo* - Tal como a mídia de áudio, a mídia de vídeo se caracteriza por gerar um tráfego contínuo com taxa de constante. Uma vantagem é que perdas de informações não são tão críticas como a de áudio, porém a banda necessária para trafegar dados de vídeo é muito maior do que a de áudio. Nesta mídia a compressão é fator decisivo para tráfego da mesma. Entre as técnicas estão: MJPEG, MPEG, H.261, H.263, CellB, NV e CuSeeMe.

A complexidade dos sistemas que utilizam múltiplas mídias exige bastante do equipamento, a nível de processadores, softwares e sistemas de arquivo. A arquitetura tem que garantir uma grande largura de banda e um sistema de entrada e saída eficientes.

## 8.1. Princípios básicos

Em uma videoconferência podemos encontrar os seguintes elementos básicos: Captadores; Transdutores; CODEC e Meio de Transmissão.



Os Captadores são os elementos que conseguem obter informações de um participante, transformando-as em sinal elétrico. Entre eles: microfones, câmeras de vídeo, teclado, movimento do mouse, etc

Os Transdutores são os elementos que geram informações para um participante a partir de sinais elétricos. Entre eles: Televisão, projetores, monitor de vídeo, autofalantes, etc.

Os CODECs têm o papel de gerar um sinal digital e/ou analógico contendo a informação dos dados a serem transmitidos, de modo que estes sejam compatíveis com o meio em que circularão. Cabe a este elemento, o papel de comprimir e codificar os sinais dos Captadores e descomprimir/decodificar os sinais para os Transdutores. Estes podem ser implementados por software, hardware ou ambos.

O meio de transmissão é o caminho físico por onde os dados deverão trafegar da origem ao destino podendo utilizar diferentes tecnologias de comunicação. Entre elas, temos: ISDN, ATM, FrameRelay, Ethernet, Fast-Ethernet, etc.

Dentro deste modelo podemos ter várias classificações, das quais podemos citar:

### A) Quanto ao ambiente

- ✓ *Desktop Videoconference* - nesta utilizamos uma estação multimídia para que um indivíduo participe de uma conferência. A estação tem o papel de Transdutor, Captador e CODEC;
- ✓ *Room Videoconference* - neste caso um ambiente é preparado para que vários participantes possam participar ao mesmo tempo da conferência. Os Transdutores e Captadores são espalhados pelo ambiente. Existe ainda um equipamento com a função de CODEC que pode ser uma estação ou um hardware dedicado.

### B) Quanto a conexão

- ✓ *Ponto-a-Ponto* - algumas soluções só permitem conexões entre dois pontos, isto é, devemos definir previamente os locais que poderão se comunicar.
- ✓ *Multicast* - nesta solução é aberto um canal virtual para este tipo de comunicação. Neste canal trafegam os dados de todos os participantes. Cada usuário decide o que deseja ver.
- ✓ *Multiponto* - neste caso existe uma conexão ponto-a-ponto com um equipamento (refletor) que tem o papel de receber os dados enviados por um usuário e reenvia-los para os demais participantes, além de, transmitir para este usuário os dados dos outros participantes.

### C) Quanto ao canal de comunicação

- ✓ Canal Chaveado por Circuito: é o método de transferência de dados onde o caminho de comunicação é estabelecido no início e permanece durante a duração da sessão.
- ✓ Canal Chaveado por Pacote: é o método de transferência de dados onde a informação é dividida em pacotes, cada um dos quais possuindo uma identificação de origem e destino. Os pacotes são enviados individualmente através da rede e, dependendo das condições da rede, podem tomar rotas diferentes e chegar ao destino fora de ordem.

## 8.2. Softwares de Videoconferência

Dentre os inúmeros softwares existentes no mercado podemos destacar dois destes:

- ✓ CU-SeeMe - é um software de *desktop conferencing* para computadores Mac e PC criado pela Cornell University (<http://cu-seeme.cornell.edu>). Este software provê conexão ponto-a-ponto dando comunicação interativa de voz, áudio e chat em cima de redes baseadas em TCP/IP. Com o uso de um software chamado refletor, CU-SeeMe provê conferência Multiponto para a comunicação entre os participantes.
- ✓ Mbone - Softwares de diferentes fabricantes que utilizam os conceitos de multicast para difusão da videoconferência sendo seu código divulgado livremente na Internet. Entre eles: Difusão de Vídeo - VIC, NV Difusão de Áudio - RAT, VAT Difusão de imagens e figuras, WB Gerenciamento de sessões, SD, SDR. Existem versões para Sun, Silicon, IBM, PC, MAC, etc.

Podemos citar outras soluções proprietárias de hardware e/ou software: Invision Desktop Video - Invision Systems Corp., PictureTel Live PCS 1000 - PictureTel Corp., InPerson Desktop Conferencing - Silicon Graphics, ShowMe - Sun Microsystems Computer Co. e Vistium Personal Video 1300 - AT&T.

Para se fazer uma videoconferência usando o Mbone, um dos participantes precisa criar uma sessão que deve possuir um determinado *endereço IP Classe D*. Os demais participantes que desejarem receber ou enviar dados, devem juntar-se a esta sessão.

Existem diversos aplicativos para a realização de videoconferências, usando *multicast*, disponíveis na Internet. O Mbone utiliza as ferramentas chamadas **SD** (*Session Directory*) e **SDR** (*Session Directory*), que criam e/ou exibem as sessões existentes. O pacote de softwares **NV** (*Network Video*) e **VIC** (*Video Conferencing Tool*) são usados para o tratamento de vídeo, as ferramentas **VAT** (*Visual Audio Tool*) e **RAT** (*Robust Audio Tool*) são usados para o tratamento de áudio e a ferramenta **WB** (*Whiteboard*) é usada para textos e desenhos gráficos. A sessão é desfeita quando todos os participantes saem, liberando então o endereço IP *multicast* para reutilização.

Todos estes softwares estão gratuitamente disponíveis nas formas fonte ou binária em vários FTP's anônimos na Internet.

Dentre todos estes softwares podemos destacar quatro deles, que já foram testados e são usualmente utilizados no CBPF nas videoconferências de testes e avaliações atualmente .

### 8.2.1 Session Directory - SDR

O software **SDR** mostra os eventos Mbone disponíveis, além de possibilitar a criação de novas sessões. Substitui com vantagens o **SD**, como por exemplo na facilidade da administração dos eventos.

As Sessões podem ser de dois tipos, Públicas e Privadas. Para a criação de uma nova sessão, após a seleção da opção adequada é necessário informar quais as ferramentas que serão utilizadas, o nome e a descrição do evento, o responsável, o escopo do alcance da sessão (ttl), o dia, o horário e a duração.

Um endereço de multicast não ocupado será alocado pelo *SDR*.

### **8.2.2. Visual Audio Tool - VAT**

O software VAT foi a primeira ferramenta de audioconferência desenvolvida para a Internet. Ela roda em máquinas que suportam IP multicast e utiliza os protocolos UDP/IP e IP multicast. É um aplicativo que possibilita enviar e receber áudio através da rede.

Na janela principal são listados todos os participantes da sessão e pode-se habilitar o controle do nível de entrada e saída do som.

O gerente da sessão de videoconferência pode habilitar ou não o som dos participantes e cada um tem controle sobre o seu microfone e autofalante que podem ser configurados segundo o modo de saída do som, ou seja, como o som é “jogado” para os participantes: *Conference* - diminui a demora na saída do som, mas ocasiona maior perda de pacotes; *Lecture* - aumenta os intervalos na saída do som, porém minimiza as perdas de pacotes. A codificação de áudio também pode ser configurada.

### **8.2.3. Video Conferencing Tool - VIC**

O software VIC é uma aplicação para videoconferência, desenvolvido pelo grupo de pesquisas de rede do Lawrence Berkeley National Laboratory em colaboração com a University of California, Berkeley. Este software permite ao usuário ver e controlar sua própria transmissão, além de poder mudar a fonte de vídeo e a taxa de transmissão.

Você pode acessar o software *VIC* através do *SDR*. Na janela principal aparecerão as imagens das pessoas que estiverem transmitindo nesta sessão e os botões com as funções típicas de ajuda, configuração e saída.

O usuário pode controlar a velocidade, o número de quadros, o status da transmissão da imagem, tecnologia utilizada, ou seja, qual o tipo de codificação de vídeo usada, o tamanho da imagem, o modo de exibição e por fim, obter informações sobre a sessão corrente e os usuários que estão conectados na presente sessão.

### **8.2.4. Whiteboard - WB**

O software WB é um aplicativo que permite que um ou mais *sites* compartilhem documentos em tempo real. Este software, permite ao usuário utilizar documentos em texto ASCII, desenhos, digitações, anotações a mão livre e páginas de Postscript, para serem vistos pelos demais integrantes da sessão de videoconferência. Não é necessário nenhum hardware externo. O princípio básico é uma janela comum a todos os usuários que podem escrever/desenhar de tal forma que o que for feito por um, é visto por todos simultaneamente.

## 9. Conclusão

O crescimento exponencial da Internet se deve principalmente ao desenvolvimento do ambiente Web e em particular à sua utilização pela área comercial. A área científica já utilizava a Internet (*e-mail*, Telnet, FTP e etc.) a aproximadamente 20 anos e na realidade é neste ambiente que os avanços tecnológicos ocorrem primeiramente. É comum institutos de pesquisa e universidades serem convocados por empresas privadas para divulgação do conhecimento em áreas ou tecnologias emergentes.

Nos dias de hoje, talvez o principal problema esteja relacionado com a velocidade e confiabilidade das telecomunicações. O desenvolvimento de serviços/aplicações que envolvam áudio e vídeo (telefonia pela Internet, videoconferência, serviços multimídia, etc.) estão aguardando por tecnologias novas que elevarão a velocidade de transmissão a níveis ótimos que viabilizem a sua utilização por todos os usuários da Internet.

A principal dificuldade está no estabelecimento de conceitos, padrões, protocolos, linguagens, etc., pois a “guerra” entre as grandes empresas para impor as suas soluções proprietárias é muito grande, dificultando a compatibilização com sistemas de outras empresas. Em “guerras” deste tipo, frequentemente ganha as soluções baseadas em Sistemas Abertos.

Apesar de tudo isto, a Internet se apresenta como a mais fantástica, poderosa e irrestrita (em um futuro breve) ferramenta de divulgação de conhecimento, cultura, divertimento e comércio, ou seja, serviços de uma maneira geral. De uma forma pioneira nos meios de comunicação, a Internet tem uma característica bidirecional: o usuário é um elemento ativo no meio.

A sociedade começa absorver e utilizar as vantagens e recursos da Internet e mais do que isto, ela começa a apresentar demanda de serviços até então inimagináveis. A sociedade, cada vez mais, modifica a sua maneira de ser e agir devido a Internet. Isto é um exemplo típico de sinergia digital.

## Referências

- [01] "Hobbes'Internet Timeline v3.1", <http://www.isso.org/zakon/History/HIT.html>
- [02] "The History of the Internet", <http://www.davesite.com/Webstation/net-history.shtml>
- [03] "Notas e Informações Sobre a Internet", <http://www2.madinfo.pt/organismos/cecha/internet/net11.htm>
- [04] "Internet Brasil", <http://www.geocities.com/Athens/Parthenon/2540/olm3brap.html>
- [05] "A Internet no Brasil", <http://www.geocities.com/CollegePark/8690/brasnet.htm>
- [06] "Comitê Gestor Internet Brasil", <http://www.gt-er.cg.org.br>
- [07] "O Que É Internet", <http://www.momntreal.com.br/oquee.htm>
- [08] "Manual da Internet", <http://www.wartnet.com.br/~lopes/historia.htm>
- [09] "A História da Internet", <http://www.geocities.com/Athens/9402/intru.html>
- [10] "Using The Internet", W. A. Tolhurst, M.A. Pike, K. A. Blanton and J. R. Harris, ISBN: 1-56529-353-3, QUE Corporation, 1994.
- [11] "Using Linux", J. Tackett, D. Gunter and L. Brown, ISBN: 0-7897-0100-6, QUE Corporation, 1995.
- [12] "Guia Internet de Conectividade", Cyclades Brasil, 1996.
- [13] "Computer Networks", A. S. Tanenbaum, ISBN: 0-13-349945-6, Prentice-Hall Inc., 1996.
- [14] "Netware v4.10 – Considerações Iniciais", A. F. Novello e N. Alves, Nota Técnica – CBPF-NT-002/96, setembro de 1996.
- [15] "Solaris 2.X Network Administration – SA-380", Student Guide, Sun Microsystems, Part Number 802-6159-01, Revision C, February 1996.
- [16] "Um Tutorial da Internet", <http://home.microsoft.com/intl/br/tutorial/>.
- [17] "HTML 4.0 Specification", <http://www.w3.org/TR/REC-html40/>, 18/12/1997.
- [18] "World Wide Web", M.P. Albuquerque e N. Alves, CBPF-NT-003/95, 1995.
- [19] "Common Gateway Interface", NCSA, <http://www.w3.org/CGI/Overview.html>, 04/05/1996.
- [20] "Special Edition Using Javascript", <http://www.palenaka.com/~palenaka/que/quebooks/java/toc.htm>
- [21] "HTML – Introdução", D. C. A. Costa, F. S. Jannuzzi, M. Maestrelli, CBPF-NT-002/98, 1998.

## Leituras Alternativas

- A) "Windows NT Server Internet Guide - Microsoft Professional Editions", Microsoft Press, ISBN: 1-57231-344-7, 1996.
- B) "Connected: An Internet Encyclopedia", Brent Baccala, <http://cbpf-catnt.cat.cbpf.br/connected/>, 1996
- C) "Coping with Unix – An Iterative Survival Kit", Northwest Alliance for Computational Science and Engineering, <http://www.physics.orst.edu/tutorial/unix>.
- D) "Shlurrrpp.....Java – Tutorial on Java Programming", Vijay Mukhi's Computer Institute, <http://193.49.43.3/nfp/documents/java/tutorial/java.html>
- E) "Web Publishing Paradigms", T.Guay, <http://hoshi.cic.sfu.ca/~guay/Pradigm/>
- F) "Sistemas de Telecomunicações", H. Fuks, <http://www.inf.puc-rio.br/~refletor/st/>
- G) "Multi-cast Backbone", Microsoft WhitePapers, <http://www.microsoft.com/mind/0297/mbone/mbone.htm>
- H) "Face-to-Face Collaboration", Revista BYTE, Nov 1994
- I) "Stardust IP Multicast Initiative", Stardust Technologies, Inc, <http://www.ipmulticast.com/>
- J) "Mbone (or IP Multicast) Information Web", ICAST Corporation, <http://www.mbone.com/>