

Protocolo HTTP

RODRIGO PREIS BEOCK, Leandro¹; DE FREITAS CONSONE, Cibele²; RODRIGUES
LIMA, Leandro³; PETRICA, Eder⁴
UNIVERSIDADE DO ESTADO DE MATO GROSSO/UNEMAT - COLÍDER⁵

RESUMO:

O presente resumo tem por objeto tratar do conceito do protocolo HTTP usado em paginas web. O mesmo trata de conceitos da World Wide Web, conceitos sobre a troca de mensagens cliente-servidor através do protocolo HTTP, ampliando desta forma o conhecimento e o entendimento do assunto.

Palavras-chave: Navegador, aplicativos, web, cliente, servidor.

INTRODUÇÃO

Atualmente o nosso país conta com vários cursos voltados a área da computação, para poder suprir a demanda do mercado de trabalho que também passa por um processo de crescimento, aumenta o anseio por melhores tecnologias. Uma dessas tecnologias é a arquitetura de redes web, que tem evoluído muito nos últimos anos. Portanto, é necessário empenho para que consigamos acompanhar a evolução desta tecnologia, que vem se tornando uma área muito interessante para ser pesquisada e entendida, e futuramente fazer uso deste conhecimento para o trabalho ou para qualquer outro tipo de atividade que requisite um conhecimento da área.

¹ Acadêmico de Licenciatura em Computação (UNEMAT/COLÍDER)
UNIVERSIDADE DO ESTADO DO MATO GROSSO/UNEMAT-COLÍDER
Email: piqueno_89@hotmail.com

² Acadêmica de Licenciatura em Computação (UNEMAT/COLÍDER)
UNIVERSIDADE DO ESTADO DO MATO GROSSO/UNEMAT-COLÍDER
Email: cibelly_14@hotmail.com

³ Acadêmico de Licenciatura em Computação (UNEMAT/COLÍDER)
UNIVERSIDADE DO ESTADO DO MATO GROSSO/UNEMAT-COLÍDER
Email: alves_net@hotmail.com

⁴ Acadêmico de Licenciatura em Computação (UNEMAT/COLÍDER)
UNIVERSIDADE DO ESTADO DO MATO GROSSO/UNEMAT-COLÍDER
Email: eeddeerr@gmail.com

⁵ UNIVERSIDADE DO ESTADO DO MATO GROSSO/UNEMAT-COLÍDER
Luiz Aldori Neves Fernandes, Colíder-MT – 78.500-000 – Fone: (66) 3541-2529
Email: computacao@colider.unemat.br

IDENTIFICAÇÃO DA PESQUISA

O intuito desse trabalho é apresentar, em síntese, o que é o protocolo HTTP e como ele funciona.

METODOLOGIA

Recorreu-se à pesquisa bibliográfica para o desenvolvimento deste trabalho por que segundo (GIL, 1991) a pesquisa bibliográfica permite ao investigador a cobertura de uma gama de fenômenos muito mais ampla do que aquela que poderia pesquisar diretamente. Ainda segundo o autor a pesquisa bibliográfica é indispensável no início das pesquisas.

RESULTADOS PRELIMINARES

1. A World Wide Web

A World Wide Web (Rede de Alcance Mundial) é uma rede que interliga milhares de milhões de computadores em todo o mundo, pode ser considerado como um “repositório global”, TANEMBAUM (2003) define a World Wide Web da seguinte forma: “*A World Wide Web é uma estrutura arquitetônica que permite o acesso a documentos vinculados espalhados por milhões de máquinas na Internet*”.

Por tanto a internet é uma poderosa ferramenta de comunicação e oferece uma imensa e variada quantidade de informações dos mais diversos assuntos. Também é utilizada para a transferência de dados podendo trabalhar com tipos de arquivos como: documentos, imagens, áudio, vídeo entre outros. Para (TANEMBAUM, 2003), a web é vista pelo usuário como uma coleção global de documentos, esses documentos geralmente são chamados de páginas Web ou apenas páginas. Cada página pode conter vínculos, mais conhecidos como links para outras páginas, quando o usuário clica em um desses links é levado para a próxima página, esta ideia de fazer uma página apontar para outra é conhecida como hipertexto, segundo (TANEMBAUM, 2003), esse conceito foi criado por um professor do MIT (*Massachusetts Institute of Technology*), Vannevar Bush, em 1945, bem antes da criação da internet.

Com o surgimento da internet apareceram vários protocolos que regulamentavam as transferências de dados na rede, para regularizar a situação é padronizar o MIT e a CERN (*Organisation Européenne pour la Recherche Nucléaire*) assinaram um acordo chamado de W3C (*World Wide Web Consortium*), que rege normas para a padronização da internet, e

ainda segundo (TANEMBAUM, 2003) uma destas normas é a utilização do protocolo HTTP (*Hypertext Transfer Protocol*) como padrão para navegação por páginas de hipertexto.

2. O protocolo HTTP

O HTTP – Protocolo de transferência de Hipertexto (*Hypertext Transfer Protocol*) é um protocolo da camada de aplicação da internet, segundo (KUROSE, 2006) o HTTP está dividido em duas partes, o programa cliente e o programa servidor, os dois programas são executados em máquinas diferentes, conversam um com o outro por meio de mensagens HTTP. O HTTP define a estrutura destas mensagens e o modo como são trocadas entre o cliente e o servidor.

As páginas webs são constituídas por documentos ou arquivos de vários formatos como comentado acima, essas páginas são construídas com HTML (*HyperText Markup Language*, que significa *Linguagem de Marcação de Hipertexto*) é a linguagem padrão para a programação de páginas web, a página construída em linguagem HTML é interpretada por um navegador ou navegador, que em HTTP é considerado como cliente. As páginas ou textos HTML também são considerados como documentos pelo protocolo HTTP, essas definições são explicadas melhor por (KUROSE, 2006) quando o autor fala sobre páginas web:

*Uma página web (também denominada documento) é construída de objetos. Um objeto é simplesmente um arquivo – tal como um arquivo HTML, uma imagem JPEG, uma imagem GIF, um Applet Java, um clipe de áudio e assim por diante – que se pode acessar com um único URL. A maioria das páginas web é construída de um **arquivo-base HTML** e diversos objetos referenciados. Por exemplo, se uma página web contiver um texto HTML e cinco imagens JPEG, então ela terá seis objetos: arquivo-base HTML e mais cinco imagens. (...)*

Páginas web se encontram em um servidor web, onde podem estar várias outras páginas. O cliente ou usuário acessa essa página através de um navegador que interpreta o código HTML e constrói a página a ser visualizada por ele. Uma explicação mais simplificada do funcionamento seria a seguinte: o cliente faz uma requisição de uma página ao servidor através do protocolo HTTP e o servidor responde ao cliente com a página solicitada, como pode ser vista na figura 01.



Figura 01 – Comunicação Cliente servidor. Fonte: KUROSE, 2006.

2.1 Cliente

Um navegador é essencialmente um programa que pode exibir uma página uma página web, mas primeiro é necessário encontrar a página, e como o navegador faz isso? Através da URL (Uniform Resource Locators, que quer dizer: *Localizador Padrão de Recursos*), as URLs podem ser comparadas com ponteiros que apontam para um determinado local, neste caso para uma página web. Segundo (TENEMBAUM, 2003) as URL são os identificadores universais da página, ou seja, não existem URLs repetidas, seria como o CPF de uma pessoa no Brasil.

As URLs são formadas por três partes: a primeira é o tipo de protocolo que está sendo usado, a segunda o nome DNS da máquina ou servidor em que está a página e por último o nome específico da página, normalmente o nome do arquivo que a página representa. Um exemplo, a URL para a página principal da Universidade do Estado de Mato Grosso – campus de Colíder-MT é a seguinte:

URL: <http://www.unematcolider.org/site/>

A primeira parte da URL acima destacada pela cor preta mostra a primeira parte que diz respeito ao protocolo que está sendo usado, neste caso o *http://*, a segunda parte

destacada pela cor verde mostra o DNS da máquina que ostenta a página *www.unematcolider.org*, também conhecido como nome do Host, e a terceira e última parte destacada na cor vermelha é o nome do arquivo */site/*, segundo (TANEMBAUM, 2003) o nome do arquivo é um caminho relativo ao diretório da Web padrão em *www.unematcolider.org*.

Pode-se destacar que, sob o ponto de vista do usuário, uma solicitação de uma página HTML se resume, às vezes, em apenas um clique do mouse, na verdade ela frequentemente da origem a várias solicitações HTTP que são enviadas do navegador para o servidor web, que são seguidas por suas respostas enviadas de volta pelo servidor.

Segundo (TANEMBAUM, 2003) quando um usuário clica em um Hyperlink, o navegador executa uma série de etapas realizadas em sequência, para buscar a página indicada pelo Hyperlink. E sequência de passos que o navegador realiza é demonstrada abaixo descrita por (TANEMBAUM, 2003):

1. O navegador determina o URL (verificando o que foi selecionado).
2. O navegador pergunta ao DNS qual é o endereço IP de *www.unematcolider.org*.
3. O DNS responde com 200.140.76.75.
4. O navegador estabelece uma conexão TCP com a porta 80 em 20.140.76.75.
5. Em seguida, o navegador envia um comando solicitando o arquivo */site/*.
6. O servidor *www.unematcolider.org* envia o arquivo */site/*.
7. A conexão TCP é encerrada
8. O navegador exibe todo o texto de */site/*.
9. O navegador busca e exibe todas as imagens que o arquivo contém.

Este processo pode ser visualizado observando a figura 01, é descrita uma relação entre o usuário e o servidor.

Nem todas as páginas web contêm documentos no formato HTML, as páginas podem conter documentos com vários tipos de extensões que podem ser vinculados a um documento HTML, em vez de se construir navegadores que interpretem todos os tipos de formatos, o que não seria viável porque os navegadores ficariam enormes, para poderem interpretar todos os formatos existentes, e que vem crescendo a cada dia. Uma solução encontrada para esse problema é o MIME (Multipurpose Internet Mail Extensions, que em português quer dizer: Extensões Multi função para Mensagens de Internet). Quando um servidor retorna uma página

ele envia algumas informações adicionais como MIME. Em (TANEMBAUM, 2003) o autor explica claramente esse processo:

“Páginas do tipo text/html são exibidas diretamente, como também as páginas criadas em alguns outros tipos internos. Se o tipo MIME não for um dos tipos internos, o navegador consulta sua tabela de tipos MIME para saber como exibir a página. Essa tabela associa um tipo MIME a um visualizador”.

Ainda segundo o autor existem duas formas de abrir um arquivo que não é interno do navegador, a instalação de um plug-in ou uma aplicação auxiliar, como ilustrado na figura 02 (a). O plug-in é uma extensão para o próprio navegador, e tem acesso a página a página atual, quando o usuário termina seu trabalho e o plug-in não é mais utilizado ele é retirado da memória.

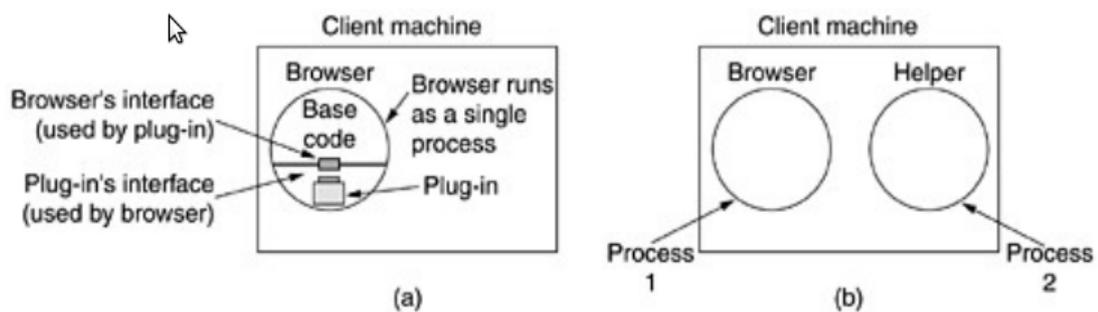


Figura 02: Plug-in e aplicativo auxiliar. Fonte: (TANEMBAUM, 2003).

O que o autor quer dizer quando fala de uma aplicação auxiliar é que quando o navegador não consegue interpretar um documento por exemplo um arquivo PDF, ele procura no registro do sistema operacional um programa para visualizar o documento, no caso do PDF, no sistema Windows (por exemplo) ele chama o programa *Acrobat Reader* ou qualquer outro que esteja instalado no sistema que seja capaz de abrir o arquivo, a figura 02 (b) ilustra o funcionamento de um aplicativo auxiliar ao navegador.

2.2 Servidor

Segundo (KUROSE, 2006) um servidor é uma aplicação, que espera passivamente de forma assíncrona por um conexão e não o computador em que a aplicação é executado, sendo assim, segundo (KUROSE, 2006) é incorreto afirmar que um computador com um processador rápido e de alta potencialidade de Hardware seria um servidor.

De certa forma um servidor web se divide em três partes o servidor propriamente dito a rede e o browser (cliente). Teixeira define servidor web como:

“Um servidor web esta sempre em um laço infinito, permanentemente aguardando por requisições dos clientes. Nesta espera, existem alguns atrasos que são inevitáveis, como a espera pela transmissão dos dados na rede, o acesso ao disco do servidor, o escalonamento dos processos pelo sistema operacional, entre outros. O servidor, portanto, deve ser projetado de modo a atender o maior numero de requisições que lhe seja possível”.

Para o cliente solicitar algo ao servidor ele utiliza métodos para fazer requisições ao servidor, assim como o servidor também utiliza-se de métodos para enviar a respostas as solicitações dos clientes, os métodos HTTP serão tratados com mais detalhes na próxima sessão.

2.2.1 Métodos HTTP

O protocolo HTTP define um conjunto de métodos que o cliente pode invocar, que funcionam como comandos enviados ao servidor web. O protocolo HTTP 1.0, descrito na RFC 1945 (Berners-Lee *et al.*, 1996), são eles GET HEAD e POST.

As mensagens de requisição e resposta do protocolo HTTP seguem um padrão, uma requisição é constituída de uma linha informando o método a ser executado em seguida de um cabeçalho que pode ter mais de uma linha e por ultimo o corpo da mensagem caso seja necessário.

Segundo (TANEMBAUM, 2003) o método GET é uma solicitação ao servidor de envio de pagina ou objeto exemplo:

```
GET /index.html HTTP/1.1
Host: www.unemat.br
Connection: close
User-agent: Mozilla/4.0
Accept-language: fr
```

Neste modelo de requisição podemos notar que na primeira linha esta o tipo do método (GET) o arquivo solicitado e por ultimo o modelo do HTTP usado na segunda linha esta o endereço do host, de onde esta hospedado o objeto, Connection: close mostra que não é uma conexão persistente, User-agente: informa o browser utilizado e Accept-language: refere-se a linguagem que o é solicitada, caso ela não exista no domínio será usado a linguagem default.

A resposta do servidor deve ser:

```
HTTP/1.1 200 OK
Connection: close
Date: Mon, 12 Dec 2005 04:15:03 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Sun. 5 May 2005 09:25:23 GMT
Content-Length: 6821
Content-Type: text/html
```

Analisando a mensagem de resposta identificamos na primeira linha o estado da solicitação que no modelo a cima esta OK, o servidor esta enviando o objeto solicitado, Connection: close, esta informando que é uma conexão não persistente, Date, é a data de acesso ao objeto, Serve, o tipo de servidor web , Last-Modified, data da última alteração ou criação do arquivo Content-Length, tamanho do arquivo e por último Content-Type, reponsavel por informar o tipo do arquivo.

A baixo segue relação dos métodos usado:

GET: Método que solicita algum recurso ou objeto ao servidor

HEAD: Solicita informações de um determinado objeto sem que esse seja enviado ao cliente apenas para testa a validade do ultimo acesso.

POST: Método usado para envio de arquivo dados ou formulário HTML ao servidor.

OPTIONS: Por meio desse método o cliente obtém as propriedades do servidor.

DELETE: Informa por meio do URL o objeto a ser deletado.

TRACE: Para enviar mensagem do tipo loopback para teste.

PUT: Aceita criar ou modificar algum objeto do servidor.

CONNECT: Comunicar com servidores Proxy.

A próxima sessão trata dos cookies, que ajudam os servidores manterem informações de visita de seus usuários.

2.3 Cookies

O termo cookie é derivado do inglês que significa bolacha. Recebeu esse nome de uma antiga gíria usada pelos programadores que consistia em um programa chamava um procedimento e recebia de volta algo que seria necessário apresentar novamente mais tarde para realizar algum trabalho. Foi criado pela Netscape para solucionar o problema do envio e solicitação de arquivos, que era esquecido pelo servidor e que poderia ser usado por outros computadores com o mesmo IP conforme (TANEMBAUM, 2003), o que causava problemas, pois não se sabia na realidade se era ou não aquele usuário mesmo.

Os cookies são arquivos ou strings e não são programas executáveis. Eles são tratados como dados pelo navegador, não existe nenhuma maneira dele ser usado como vírus, apensar de que podem ser explorados bugs no servidor e causar a ativação de um cookie como vírus, por um hacker.

Basicamente ele é um grupo de dados trocados entre o servidor de páginas e o navegador colocado em um ficheiro criado no computador do usuário. Serve para manter a persistência das sessões HTTP.

Ele funciona da seguinte forma: Um usuário solicita uma página da Web, nisso o servidor pode fornecer informações adicionais acompanhando a página solicitada. Essas informações podem incluir um cookie, um pequeno arquivo ou string (com 4 KB no máximo). Este cookie pode ter até 5 campos (figura 03): Domain, Path, Content, Expires, Secure. Domain informa de onde veio o cookie. O navegador confirma que os servidores estão enviando dados fiéis a respeito de seu domínio. Cada domínio pode armazenar no máximo 20 cookies por cliente. O campo Path é um caminho na estrutura de diretórios do servidor que identifica as partes da árvore de arquivos do servidor que podem usar o cookie. Frequentemente, ele obtém o símbolo / (barra), que representa a árvore inteira. O campo Content utiliza a forma nome = valor, podendo o servidor definir da maneira que quiser tanto o valor quanto o nome, e é nele que fica armazenado o conteúdo do cookie. Expires é o campo que faz o cookie persistir, nele contem a data e o horário, e se ele estiver ausente o navegador descartará automaticamente após o término da sessão. O último campo define se ele é seguro ou não.

Domain	Path	Content	Expires	Secure
---------------	-------------	----------------	----------------	---------------

toms-casino.com	/	CustomerID=497793521	15-10-02 17:00	Yes
joes-store.com	/	Cart=1-00501;1-07031;2-13721	11-10-02 14:22	No
aportal.com	/	Prefs=Stk:SUNW+ORCL;Spt:Jet s	31-12-10 23:59	No
sneaky.com	/	UserID=3627239101	31-12-12 23:59	No

Figura x: Alguns exemplos de cookie. Fonte: (TANEMBAUM, 2003).

O cookie é usado para identificar um usuário que configurou uma página web, para que na próxima vez que ele entrar ela esteja configurada do modo em que ele deixou. Pode ser usado também quando se faz a solicitação de armazenamento de senha, na vez posterior em que entrar no site, a sua senha será lembrada. É usado também em sites de compra, como e-commerce, armazenando os produtos que o cliente colocou no carrinho para que no final da compra não necessite fazer todo o processo novamente.

CONSIDERAÇÕES FINAIS

Os resultados obtidos após a conclusão deste trabalho foram muito satisfatórios, apesar das muitas dificuldades encontradas no decorrer do projeto, o mesmo foi concluído com êxito e os conhecimentos obtidos durante a construção deste serão de grande valia para futuras implementações.

REFERÊNCIAS

GIL, A. C., **Como Elaborar Projetos de Pesquisa**, 3ª Edição, São Paulo-SP, Atlas, 1991.

TEIXEIRA, Mário A. M., **Suporte a serviços diferenciados em servidores web: modelos e algoritmos**, Acessado em: Novembro de 2010. Disponível em: <http://www.deinf.ufma.br/~mario/producao/tese_swds.pdf>.

KUROSE, James. F. & ROSS, Keith W. **Redes de Computadores e a internet: Uma abordagem top-down**, 3ª Edição, Editora Pearson, São Paulo– SP, 2006.

TANEMBAUM, Andrew S. **Redes de Computadores**, 7ª Edição, Editora Campus, Rio de Janeiro – RJ, 2003.