

[< VOLTAR](#)

# A estrutura e funcionamento das redes TCP/IP

Conceituar estrutura e funcionamento das redes baseadas no Modelo de referência TCP/IP, utilizando a referência básica do protocolo HTTP. Como complemento desta aula, serão abordados os métodos GET e POST, utilizados para o envio de dados.

## NESTE TÓPICO

- › Introdução
- › O protocolo HTTP
- › Método de envio de dados: GET e POST
- › Envio de dados via formulário
- › Envio de dados via link
- › Referências



## Introdução

O desenvolvimento da arquitetura TCP/IP foi patrocinado pelo DARPA (*Defense Advanced Research Projects Agency*) e tem como base principal um serviço de transporte orientado à conexão, fornecido pelo protocolo TCP, em um serviço de rede não orientado à conexão (datagrama não confiável), fornecido pelo protocolo IP. Lembre-se que o protocolo TCP está localizado na **camada de transporte**, tanto no modelo de referência OSI, quanto no TCP/IP, assim como o protocolo UDP (*User Datagram Protocol*).

Esse modelo surgiu no final da década de 1960, em meados da então chamada "Guerra Fria". Nessa época, predominava o medo de ataques em grandes proporções aos EUA (tanto às forças armadas quanto às universidades). Lembre-se que a Guerra Fria foi ideológica, com disputas pelas principais zonas de influências mundiais, causando a divisão do mundo em 2 grandes zonas: o bloco capitalista, liderado pelos Estados Unidos, e o bloco socialista, liderado pela União Soviética.

O propósito desse modelo era manter os sistemas de comunicação de dados interconectados mesmo em caso de avarias totais aos servidores e linhas de comunicação, mantendo os pacotes em trânsito, não importando por qual meio de transmissão. Por este motivo, a camada intra-rede (enlace/física no modelo OSI) não possui foco para protocolos, e sim a partir da camada de acesso à rede (camada de rede no modelo *OSI*).

## O protocolo HTTP

O HTTP (*HyperText Transfer Protocol*) é o protocolo que possibilita que os dados sejam transportados pela rede mundial de computadores. A *World Wide Web* (WWW).

O HTTP é um protocolo que consegue trafegar de forma leve e rápida, suprimindo a necessidade básica para um sistema de informação distribuído colaborativo hipermídia e o protocolo HTTP trabalha sobre uma conexão TCP/IP. (W3C, 2012a).

Para entendermos um pouco melhor sobre o protocolo HTTP, nada melhor que verificarmos sua definição original de 1991, pela W3C (2012b), que explica que a conexão a um servidor web é realizada através de um nome de domínio ou de um número de IP válido. Se um número de porta não for especificado, o default é 80 para o protocolo HTTP.

Feita a conexão com o servidor, o cliente envia uma solicitação (**Request**) para o servidor.

O servidor, por sua vez, analisa a solicitação e devolve uma resposta (**Response**) de volta para o cliente no formato HTML.

O fluxo Request/Response ocorre com a troca de bytes de caracteres ASCII.

O browser instalado no equipamento cliente é o responsável por esta solicitação do servidor obter e processar o texto HTML recebido.



## Método de envio de dados: GET e POST

Como já mencionamos anteriormente, para estabelecermos a comunicação, utilizamos um navegador como cliente "e o servidor é um computador que possui um software servidor web responsável por aceitar requisições feitas nos navegadores (usando o protocolo HTTP) e devolver respostas na forma de elementos em HTML". (TERUEL, 2009, p.135).

No entanto, para que informações sejam enviadas para o servidor, utilizamos os métodos **Get** ou **Post**, embora existam outros métodos que podem ser enviados ao servidor como HEAD, PUT, DELETE, OPTIONS, TRACE e CONNECT (W3C, 2012c; GONÇALVES, 2007, p.2), no entanto, nosso foco são os dois primeiros.

Segundo Teruel (2007), há duas formas de envio de dados do cliente para o servidor através do navegador. Uma é através de formulários e a outra, através de links.

## Envio de dados via formulário

Tanto o **Get** quanto o **Post** enviam dados ao servidor através de um formulário. São enviados os dados informados pelo usuário e as informações do equipamento cliente, como um cabeçalho, que identifica no lado servidor o método, a versão do protocolo HTTP, o endereço do host, informações do browser, etc.

A principal diferença entre os dois métodos é que o Get envia os dados exibindo-os no campo de endereço do navegador enquanto o Post envia os dados no corpo da requisição e não exibe no campo de endereço do navegador. (TERUEL, 2009, p.137; GONÇALVES, 2007).

Vejamos um exemplo de página escrita em HTML para envio de dados ao servidor.

A imagem mostra um formulário web simples. No topo, há o rótulo "Nome..:" seguido de um campo de entrada de texto. Abaixo dele, há o rótulo "E-Mail:" seguido de um campo de entrada de texto mais longo. Na base do formulário, há um botão retangular com o texto "Enviar" no centro.

Figura 1: Formulário básico com 2 mensagens, 2 text e 1 submit.

Vejamos agora o HTML, com o método Get sendo utilizado para enviar os dados:

```
1. <form action="mailto:email@dominio.com.br" method="get">
2.     <pre>
3.         Nome..: <input type="text" name="nome" size="40" />
4.         E-Mail: <input type="text" name="mail" size="50" />
5.             <input type="submit" name="enviar" value="Enviar" />
6.     </pre>
7. </form>
```



No entanto, com o método Get, vejamos como fica o campo de endereço do browser após enviar os dados (lembrando que o método Get será o **default**, se não informado):

```
1. http://localhost:8080/webTeste/pesquisa.jsp?nome=Antonio+Andrade
2. &mail=aas%40hotmail.com.br&enviar=Enviar
```

Se fosse utilizado o método Post ao invés de Get, os dados seriam enviados ao servidor da mesma forma e não seria exibido no campo de endereço do **browser**.

## Envio de dados via link

Conforme Teruel (2007), o envio de dados através de links é realizado através do método Get.

Vejamos o seguinte exemplo em HTML:

```
1. <a href="pesquisa.jsp?nome=Antonio&estado=SP">Pesquisa</a>
2.
```

Que, ao clicar no link:

```
1. http://localhost:8080/webTeste/pesquisa.jsp?nome=Antonio&estado=SP
```

Enfim, conseguimos ver algumas poucas linhas de código! Como já ficou claro anteriormente, a base teórica é importante e necessária para que na hora da prática as coisas fiquem mais fáceis.

## Referências

GONÇALVES, Edson. *Desenvolvendo aplicações web com JSP, servlets, Java Server Faces, Hibernate, EJB 3 Persistence e Ajax*. São Paulo: Ciência Moderna, 2007.

TERUEL, Evandro Carlos. *Web total: desenvolva sites com tecnologias de uso livre: prático e avançado*. São Paulo: Érica, 2009.

W3C. Basic HTTP as defined in 1992. Disponível em: <[www.w3.org/Protocols/HTTP/HTTP2.html](http://www.w3.org/Protocols/HTTP/HTTP2.html)> (http://www.w3.org/Protocols/HTTP/HTTP2.html). Acesso em 08 set. 2012.

\_\_\_\_\_. The Original HTTP as defined in 1991. Disponível em: <[www.w3.org/Protocols/HTTP/AsImplemented.html](http://www.w3.org/Protocols/HTTP/AsImplemented.html)> (http://www.w3.org/Protocols/HTTP/AsImplemented.html). Acesso em 08 set. 2012.

\_\_\_\_\_. Method Definitions. Disponível em: <[www.w3.org/Protocols/rfc2616/rfc2616-sec9.html](http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html)> (http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html). Acesso em 07 set. 2012.



Avalie este tópico



ANTERIOR

Servidores de conteúdo e navegadores

Biblioteca

(<https://www.uninove.br/conheca-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/>)

a-

uninove/biblioteca/sobre-

a-

biblioteca/apresentacao/)

Portal Uninove

(<http://www.uninove.br>)

Mapa do Site



Índice

Servlets, JSP e Servidores de conteúdo

© Todos os direitos reservados

Ajuda?

(<https://ava.uninove.br/curso/>)

