

As tecnologias de servidor

São diversas as tecnologias de servidor, sejam para construir um simples site com conteúdo dinâmico ou para construir um complexo sistema B2B (Business-To-Business), é necessária a utilização de ferramentas que possibilitem consultar bancos de dados, integração com sistemas corporativos, e etc.

Existem diversas tecnologias disponíveis atualmente para o desenvolvimento dessa classe de aplicações, como PHP, ASP, ASP.NET, JSP, Servlet, CGI e etc.

A utilização de Servlets e de páginas JSP oferece diversas vantagens em relação ao uso de outras tecnologias de servidor. As principais vantagens são herdadas da própria linguagem Java, como:

Portabilidade: a aplicação desenvolvida pode ser implantada em diversas plataformas, como por exemplo Windows, Linux, Macintosh, sem que seja necessário modificar ou mesmo reconstruir a aplicação.

Orientação a Objetos: a programação é orientada a objetos, simplificando o desenvolvimento de sistemas complexos. Além disso, a linguagem oferece algumas facilidades, como por exemplo o gerenciamento automático de memória (estruturas alocadas são automaticamente liberadas, sem que o desenvolvedor precise se preocupar em gerenciar esse processo).

Flexibilidade: o Java já se encontra bastante difundido, contando com uma enorme comunidade de desenvolvedores, ampla documentação e diversas bibliotecas e códigos prontos, dos quais o desenvolvedor pode usufruir sem a necessidade de gastar algum valor a mais ou até mesmo de codificar.

Além dessas vantagens, a arquitetura de servidor escrita em Java possibilita alguns benefícios adicionais, como **escalabilidade**, onde na maior parte dos servidores de aplicações modernos, é possível distribuir a carga de processamento de aplicações desenvolvidas em diversos servidores, sendo que servidores podem ser adicionados ou removidos de maneira a acompanhar o aumento ou decréscimo dessa carga de processamento.

Protocolos, Pedidos e Respostas

Existem um conjunto de protocolos que auxiliam o mapeamento de ações dos usuários do lado cliente no uso de aplicações Web.

A Web é uma aplicação cliente/servidor em grande escala, onde o cliente (um navegador Web ou um programa FTP) se conecta ao servidor usando um protocolo. O mais comum

desses protocolos é o HTTP (Hypertext Transfer Protocol), onde em uma requisição do browser é devolvido pelo servidor textos e imagens. Esse protocolo trabalha com pedidos e respostas.

O Protocolo HTTP

O protocolo HTTP começa com uma solicitação, que por sua vez o devolve com uma resposta. A seguir você tem as solicitações desse protocolo:

- **GET** – Solicita ao servidor um recurso chamado de solicitação URI. Os parâmetros da solicitação devem ser codificados nessa solicitação, para que o mesmo os entenda. Este é o método mais usado, pois é a forma como o browser chama o servidor quando você digita um URL para que ele o recupere.
 - **POST** – Embora similar ao GET, o POST contém um corpo no qual seus parâmetros de solicitação já são codificados. O mais frequente uso desse método é na submissão de formulários (X)HTML.
 - **HEAD** – Similar ao método GET, o servidor apenas retorna a linha de resposta e os cabeçalhos de resposta.
 - **PUT** – Esse método permite o envio de arquivos para o servidor Web.
 - **DELETE** – Permite a exclusão de documentos dentro do servidor Web.
 - **OPTIONS** – É possível fazer uma consulta de quais comandos estão disponíveis para um determinado usuário.
- TRACE** – Permite depurar as requisições, devolvendo o cabeçalho de um documento.

O pedido via GET

Após enviar a solicitação, o navegador pode enviar alguns cabeçalhos que fornecem informações a respeito da solicitação, seu conteúdo, o browser que enviou sua solicitação e etc. Os cabeçalhos aparecem em linhas consecutivas. Se você faz um pedido via GET a um endereço Web, como ***www.integrator.com.br***, você recebe as seguintes informações:

GET / HTTP/1.1

Host: www.integrator.com.br

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; pt-BR; rv:1.8.0.4) Gecko/20060508 Firefox/1.5.0.6

Connection: Keep-Alive

De início você tem o método utilizado para fazer o pedido dos dados, no caso o **GET**, seguido de uma barra “/”, indicando que a informação requisitada é direta a raiz do domínio. Caso haja um chamado a um diretório específico, também chamado de Diretório Virtual, você veria **GET /ajax/ HTTP/1.1**, indicando que o diretório parte da raiz, contendo o nome indicado logo após, no caso **ajax**. Há também um outra situação, a de enviar uma string de consulta, conhecida como query string, podendo também ser visto da seguinte forma:

```
GET
/eclipse/?empresa=Integrator%20Technology%20and%20Design.
```

Nesse caso, você verá o caractere especial de interrogação “?” indicando que existem uma string de consulta, seguido do nome da variável chamado no caso de “**empresa**”, um sinal de igual indicando a atribuição de valor e o valor da variável, que no caso contém uma codificação da URL, comum nos browsers modernos, onde o espaço, por exemplo, é substituído por **%20**. Na mesma linha você tem o protocolo usado, chamado de **HTTP** versão **1.1** “**HTTP/1.1**”, que se trata de um protocolo que implementa a possibilidade de ter múltiplos hosts sobre um mesmo IP, conhecidos como Virtual Hosts.

Na segunda linha você tem o Host, que indica o objetivo do pedido, no caso o endereço digitado. Essa informação é uma exigência do protocolo HTTP 1.1, pois como já foi dito, em uma máquina pode haver múltiplos hosts, e o servidor precisa saber para qual host ele deve apontar.

A terceira linha você tem o cabeçalho **User-Agent**, que é acessível ao servidor e ao cliente, onde indica o browser que você está usando, bem como o sistema operacional e a língua usada. A quarta e última linha se trata do cabeçalho **Connection**, que tipicamente tem o valor de Keep-Alive.

O pedido via POST

Um pedido via POST é muito similar ao pedido via GET, mas se diferencia pela quantidade de informações que podem ser transmitidas por essa forma.

Tipicamente, você veria os seguintes cabeçalhos em um pedido via POST:

```
POST / HTTP/1.1
```

```
Host: www.integrator.com.br
```

```
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; pt-BR; rv:1.8.0.4) Gecko/20060508 Firefox/1.5.0.4
```

Content-Type: application/x-www-form-urlencoded

Content-Length: 57

Connection: Keep-Alive

empresa=Integrator%20Technology%20and%20Design&nome=Edson

A primeira linha indica o método utilizado, no caso o **POST**. Os cabeçalhos **HOST** e **User-Agent** são iguais ao do método **GET**.

O cabeçalho **Content-Type** indica como o corpo do pedido é codificado. Os browsers sempre codificam os dados postados como **application/x-www-form-urlencoded**, que é o MIME type usado para codificar URL simples (similar ao %20 visto no envio de informações com espaço pelo método GET).

O cabeçalho **Content-Length** indica o comprimento de bytes transmitidos, que no caso são **57**. Após o cabeçalho **Connection** você tem uma linha em branco, seguida das informações transmitidas pelo método. Nesse caso você tem a transmissão de duas variáveis (**empresa** e **nome**), seguidas de suas respectivas informações.

Note o **& “e - comercial”**, o separador padrão utilizado para envio de mais de uma variável.

A resposta HTTP

A resposta HTTP tem uma linha de status (como a de solicitação) e cabeçalhos de resposta, assim como o corpo de resposta opcional.

No caso de você entrar em um site e ele estiver disponível, você teria a seguinte resposta:

HTTP/1.1 200 OK

Date: Sat, 15 Apr 2006 18:21:25 GMT

Content-Type: text/html; charset=ISO-8859-1

Content-Length: 2541

<html>

<head>

<title> Integrator Technology and Design </title>

</head>

<body>

...

Existem respostas no qual o servidor encaminha ao browser que contém números no qual você, como desenvolvedor de aplicações Web deve decorar, que são:

- **200 (OK)**: Confirmação de que a requisição foi respondida com sucesso.
- **304 (NOT MODIFIED)**: Os recursos não foram modificados desde a última vez que você fez um pedido. Isso ocorre por causa dos mecanismos de cache do browser.
- **401 (UNAUTHORIZED)**: O cliente não tem acesso autorizado para acessar a área requisitada. Pode ocorrer de o browser pedir um nome de usuário e senha, devido ao acesso restrito a essa área no servidor.
- **403 (FORBIDDEN)**: O acesso à área requisitada falhou. Isso pode ocorrer em caso de acesso a áreas que exigem login e senha e não houve autorização para aquele usuário.
- **404 (NOT FOUND)**: Não encontrado. Ocorre quando você tenta acessar uma área inexistente no endereço passado, por exemplo, páginas removidas ou recursos excluídos.

O servidor devolverá um cabeçalho **Date** que indica a data e hora em que a resposta foi gerada no servidor. O cabeçalho de Content-Type especifica o MIME type para HTML (text/html) com uma codificação ISO-8859-1 (padrão para recursos cujas informações contenham acentos, como ocorre em nosso idioma). O cabeçalho Content-Length é similar ao visto em POST. O restante são informações que serão exibidas no browser, ou seja, o conteúdo, propriamente dito, do site.

Outros Protocolos

Não menos importante que o protocolo HTTP, você tem também os seguintes protocolos disponíveis na Internet:

- **TCP** – A sigla significa Transport Communication Protocol e representa a camada de transporte de dados e comunicação entre máquinas, cliente e servidor.
- **HTTPS** – Similar ao HTTP, mas com SSL (Secure Socket Layer) sobre esse protocolo, possui criptografia que assegura a troca de dados entre cliente e servidor, evitando assim a leitura dos dados por terceiros que não contenham a chave da criptografia.
- **FTP** – Sigla para File Transfer Protocol, onde é muito usado para transmissão de arquivos para um servidor. Fornece os principais comandos para troca de arquivos.
- **SMTP** – A sigla significa Simple Message Transfer Protocol, e fornece os comandos necessários para envio de mensagens a um servidor de e-mail.

- **POP** – Sigla para Post Office Protocol, onde permite que um cliente acesse e manipule mensagens de correio eletrônico disponíveis em um servidor.
- **IMAP** – A sigla significa Internet Message Access Protocol e permite que um cliente acesse e manipule mensagens de correio eletrônico disponíveis em um servidor, assim como ocorre no protocolo POP.