

**INSTITUTO  
FEDERAL**

Paraíba

Campus  
Cajazeiras

# PROGRAMAÇÃO P/ WEB 1

## Introdução a Aplicações Web

PROF. DIEGO PESSOA

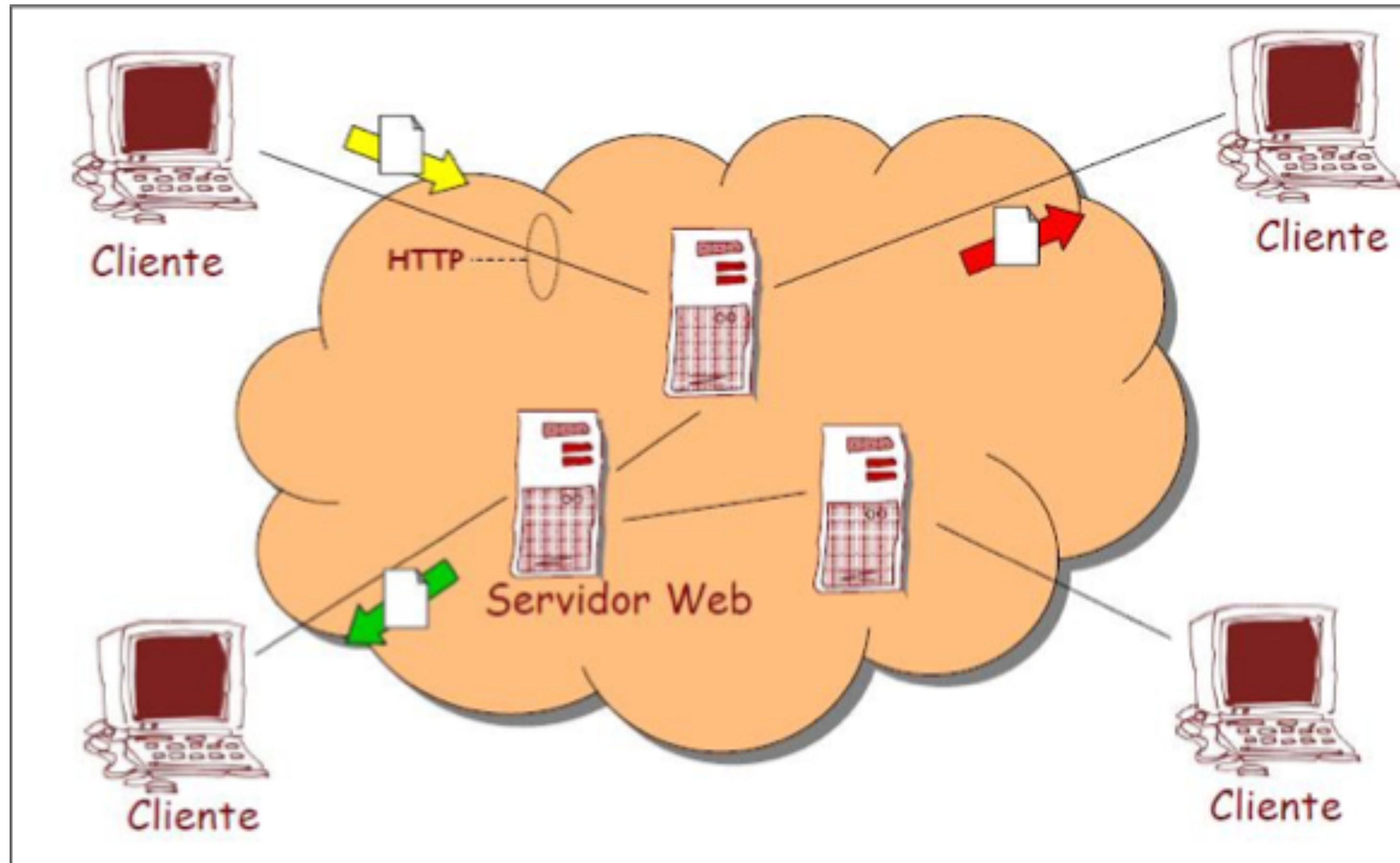
✉ DIEGO.PESSOA@IFPB.EDU.BR

🐙 @DIEGOEP



CST em Análise e  
Desenvolvimento  
de Sistemas

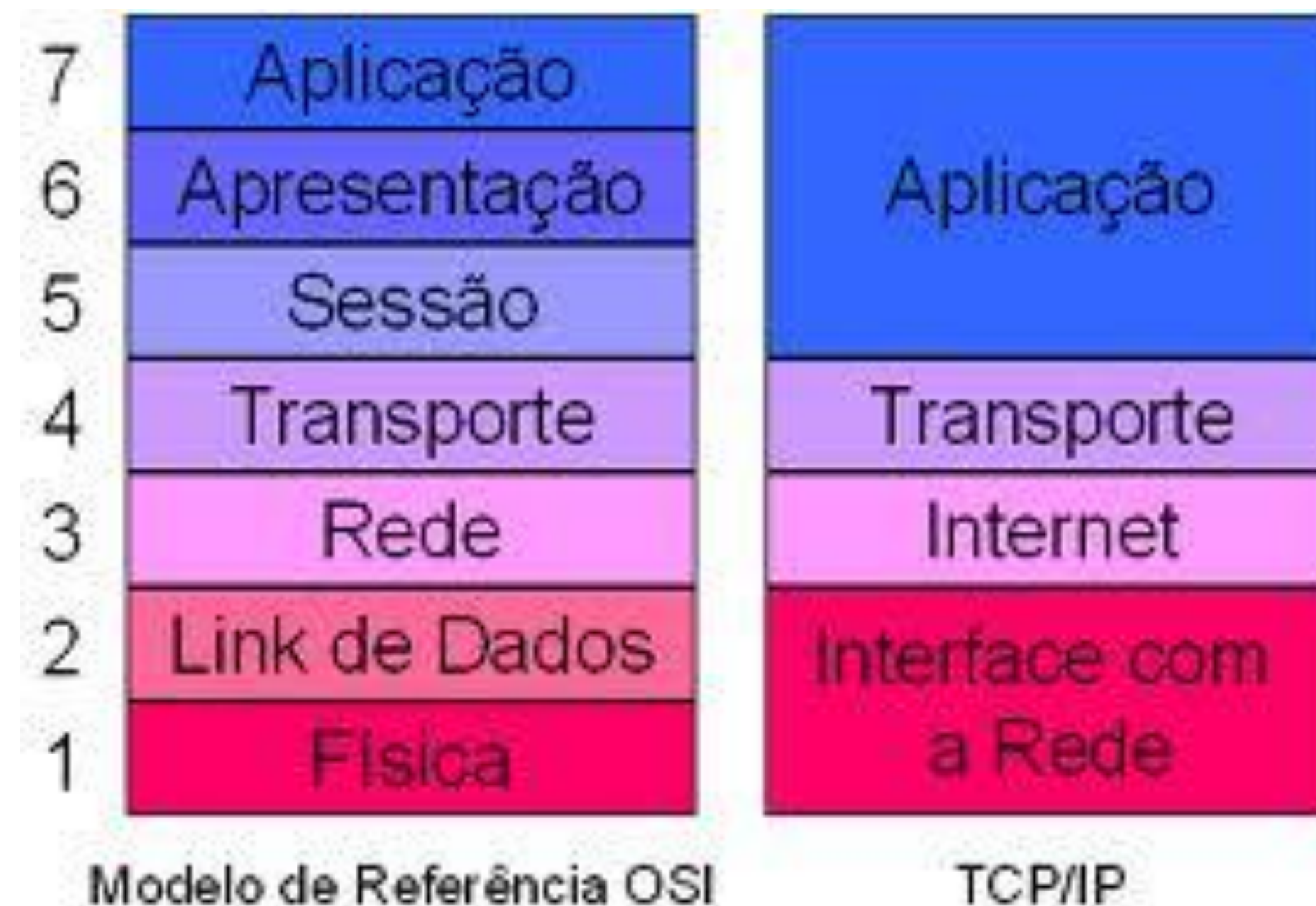
# A Web





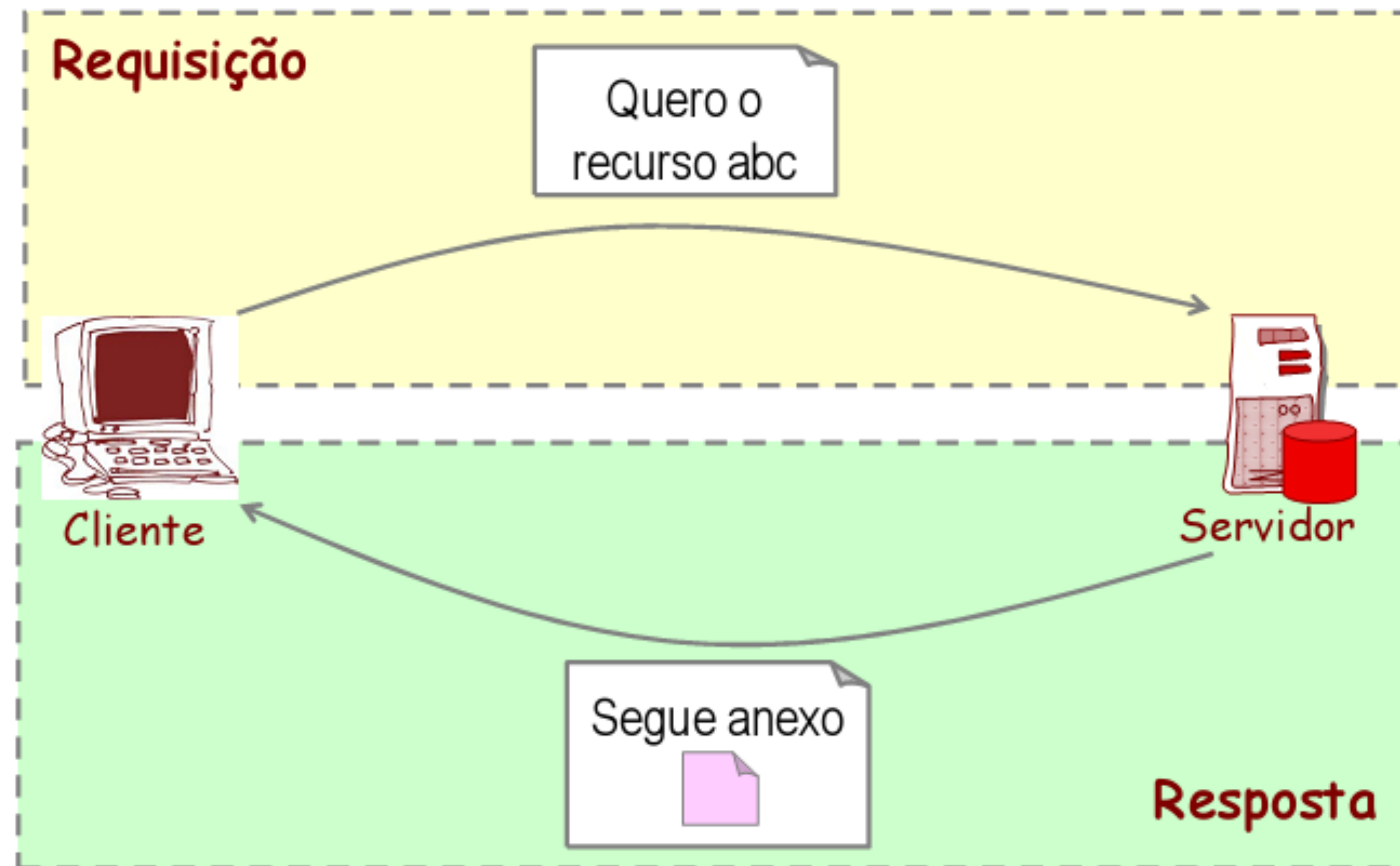
# Protocolo HTTP – Definição

- ▶ Atua na camada de aplicação para a distribuição e recuperação de informação.
- ▶ Quanto à forma de transporte, o uso mais comum é em conjunto com o TCP, já que este garante o envio íntegro de dados pela rede.



# Protocolo HTTP – Atuação

- ▶ Atua na forma de requisição/resposta, na qual é estabelecida uma conexão que depois é fechada.





# Protocolo HTTP – Atuação

## ► Requisição HTTP GET

método	path do recurso	dados	versão do protocolo
GET	/loja/consulta.jsp?id=123&tipo=10		HTTP/1.1
Host: www.lojatal.com.br			
User-Agent: IE/7.0			
Accept: text/html,application/xml,text/html,text/plain,image/gif,image/jpeg			
Connection: keep-alive			
Cookie: jsessionid=12abd4f67cc34			
parâmetros do request			

# Protocolo HTTP – Atuação

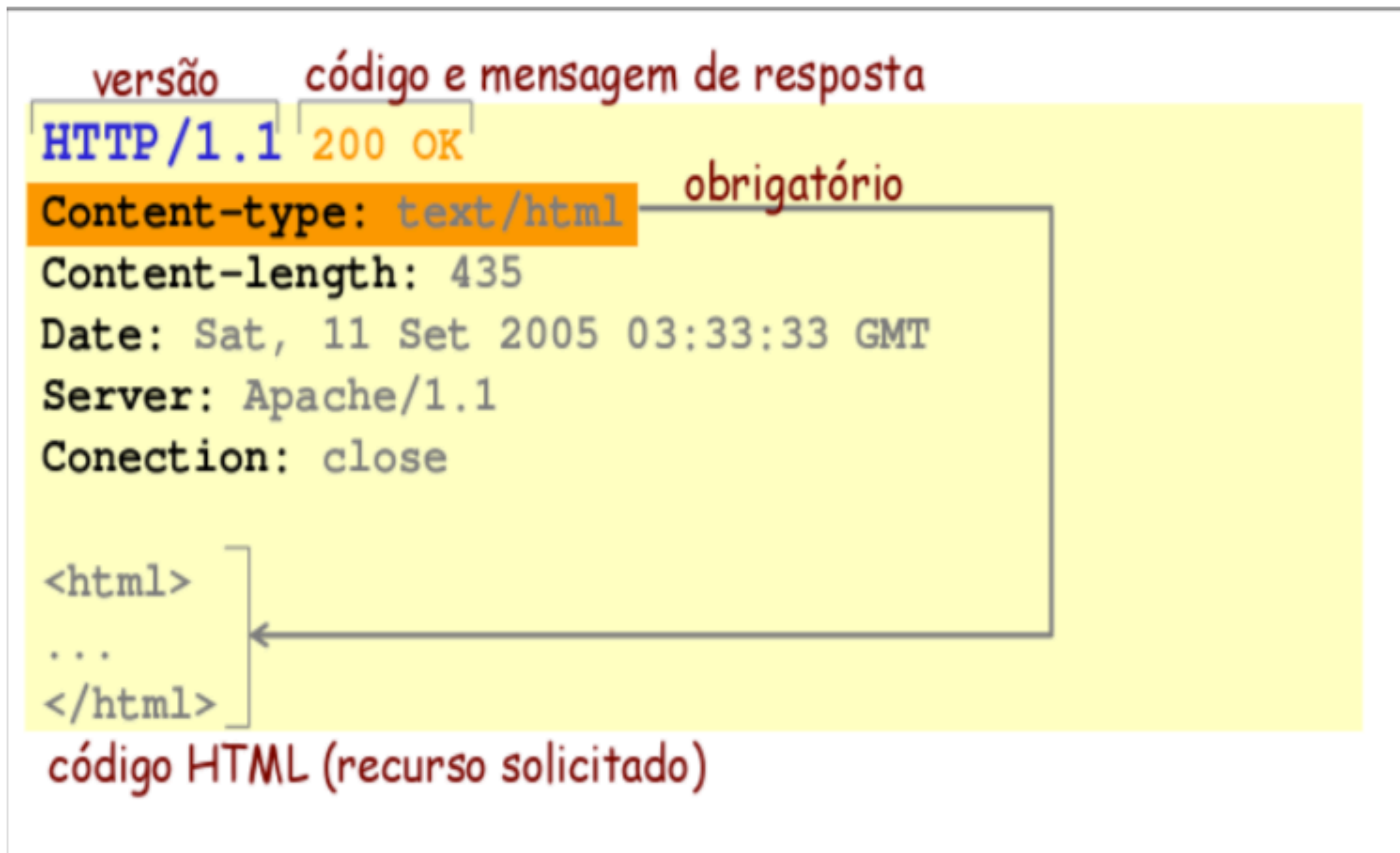
## ► Requisição HTTP POST





# Protocolo HTTP – Atuação

## ► Resposta



# Protocolo HTTP – Métodos

- ▶ O protocolo HTTP define oito métodos que indicam a ação a ser realizada no recurso especificado.
- ▶ Métodos também conhecidos como verbos são informados no momento que é feito uma requisição ao servidor de aplicação.

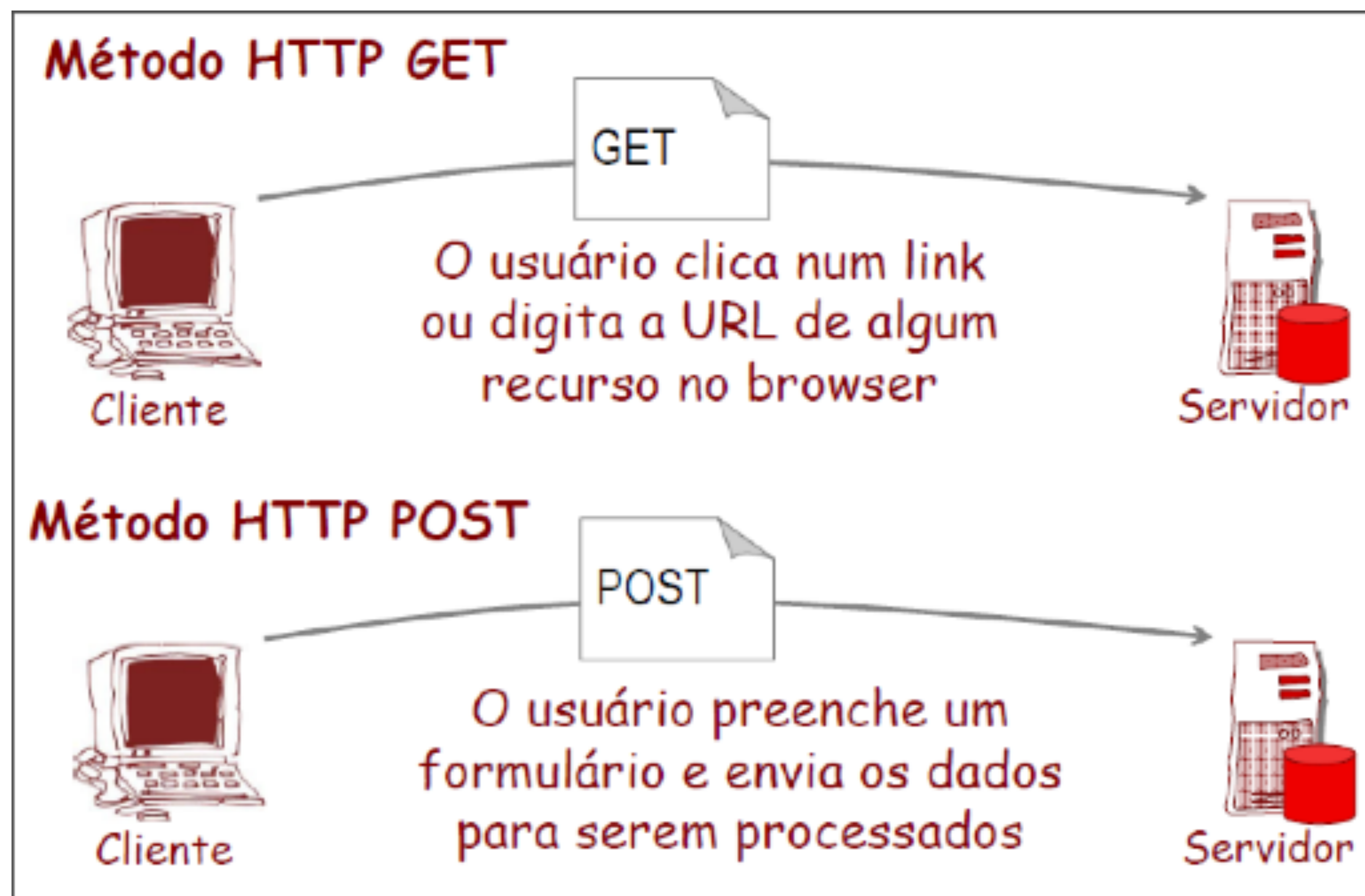


# Protocolo HTTP – Métodos

- ▶ **GET**: tem como característica enviar seus parâmetros direto na URL.
  - ▶ Os parâmetros são codificados;  
É o método mais utilizado. O browser chama o servidor para que ele retorne algum recurso. Ex.: Uma página html, foto, video, etc.
- ▶ **POST**: é um método que envia os dados no corpo da mensagem (payload) e possui características mais seguras para se enviar requisições ao servidor.
- ▶ **HEAD**: semelhante ao GET mas não obtém o recurso, apenas metadados que são a linha de resposta e o cabeçalho de resposta;
- ▶ **PUT**: envia um recurso para o servidor, por exemplo algum arquivo. É utilizado também para representar atualização. de um recurso.
- ▶ **DELETE**: exclui um recurso do servidor, por exemplo algum documento.
- ▶ **TRACE**: através da resposta obtida por uma requisição trace é possível saber o que foi alterado na requisição por um servidor intermediário.
- ▶ **OPTIONS**: consulta os métodos HTTP que o servidor aceita para um determinado usuário.
- ▶ **CONNECT**: usado em soluções de proxy.

# Protocolo HTTP – Métodos

- ▶ Os métodos mais utilizados são:



# Protocolo HTTP – Métodos

- ▶ Diferenças entre os métodos GET e POST:
  - ▶ A quantidade de dados que pode se passar no GET é limitada pelo servidor (256 bytes, em geral)
  - ▶ Os dados enviados via GET são exibidos na barra de endereços do browser (expõe dados sensíveis)
  - ▶ Não é possível marcar (favoritos) o resultado de um request que usa POST



# Protocolo HTTPS – Definição

- ▶ É uma aplicação que tem por finalidade garantir segurança nas transmissões de dados através da internet em aplicações que necessitam de sigilo de suas informações e que seus dados permaneçam privados.
- ▶ É a utilização do protocolo HTTP em conjunto com o protocolo SSL – Secure Socket Layer;
- ▶ O protocolo SSL possui uma criptografia que assegura a troca de dados entre cliente e servidor, evitando assim a leitura dos dados por terceiros que não contenham a chave criptografia;

# Protocolo HTTPS – Atuação

- ▶ O protocolo SSL atua entre as camadas de transporte(TCP) e os protocolos de aplicação

HTTP	TELNET	FTP	SMTP	SHTTP	OUTROS
SSL					
TCP					
IP					

# Protocolo HTTPS – Atuação

- ▶ As requisições ocorrem da seguinte forma:





# Protocolo HTTPS – Funcionamento

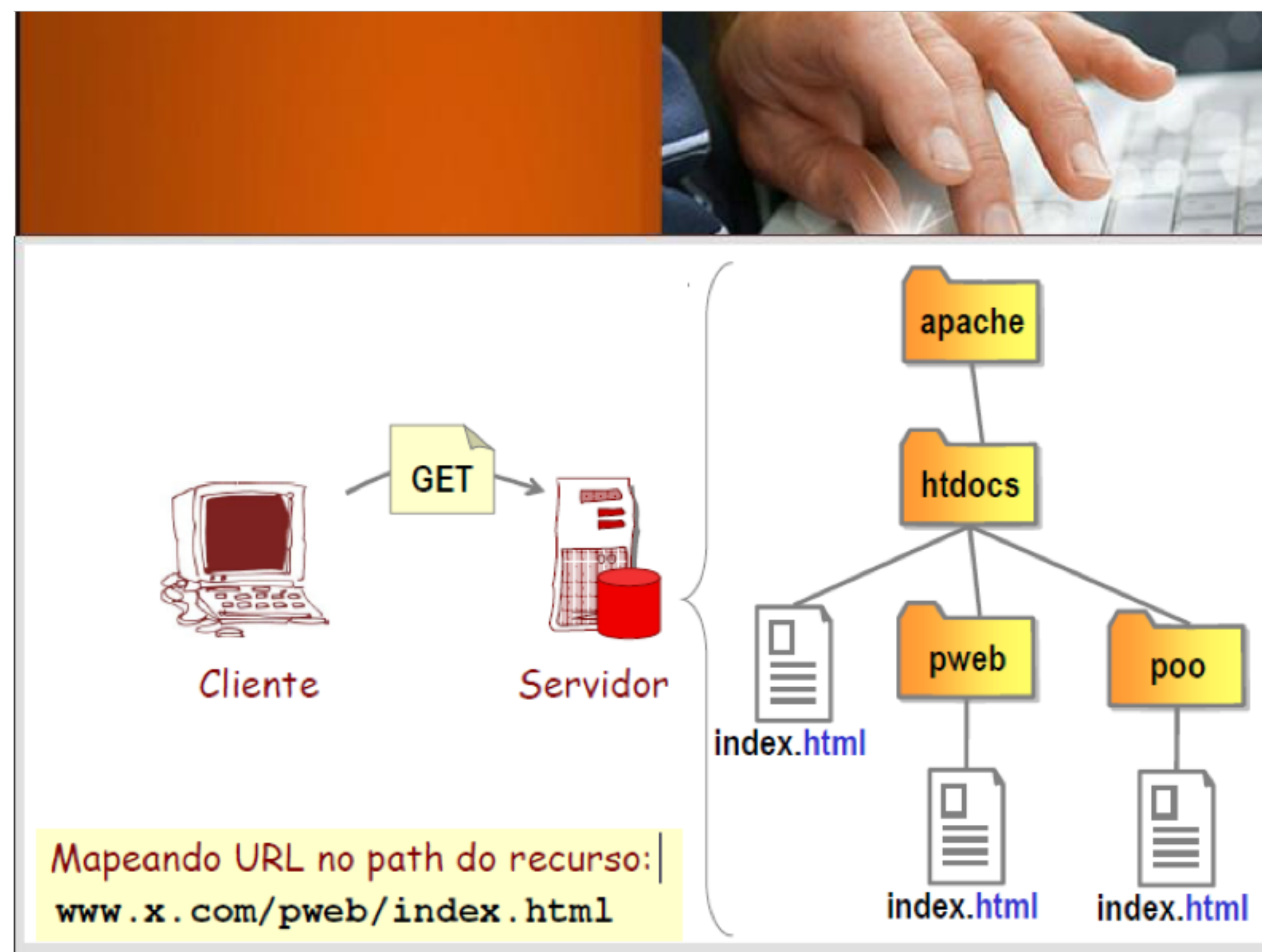
- ▶ Um usuário, através de um browser, acessa o site do servidor que disponibiliza uma página segura (HTTPS).
- ▶ Esta conexão somente será possível se o usuário estiver com a opção de SSL ativada, ou através de um proxy com SSL ativado. Caso contrário, a página estará indisponível para este usuário;
- ▶ O servidor responde enviando seu certificado digital, autenticando-se. Caso o servidor não possua um certificado digital, uma mensagem de advertência é exibida ao usuário.

# Protocolo HTTPS – Funcionamento

- ▶ O browser do usuário gera uma chave de sessão para criptografar todas as comunicações com o servidor.
- ▶ O browser do usuário criptografa a chave de sessão com a chave pública do servidor. O servidor então será o único a poder decriptografar os dados, e obter a chave de sessão.
- ▶ Uma sessão segura é estabelecida. Tudo acontece em segundos e não requer nenhuma ação do usuário. Um ícone no formato de uma chave ou de um cadeado é exibido, possibilitando ao usuário cliente ser notificado que a conexão segura foi estabelecida.

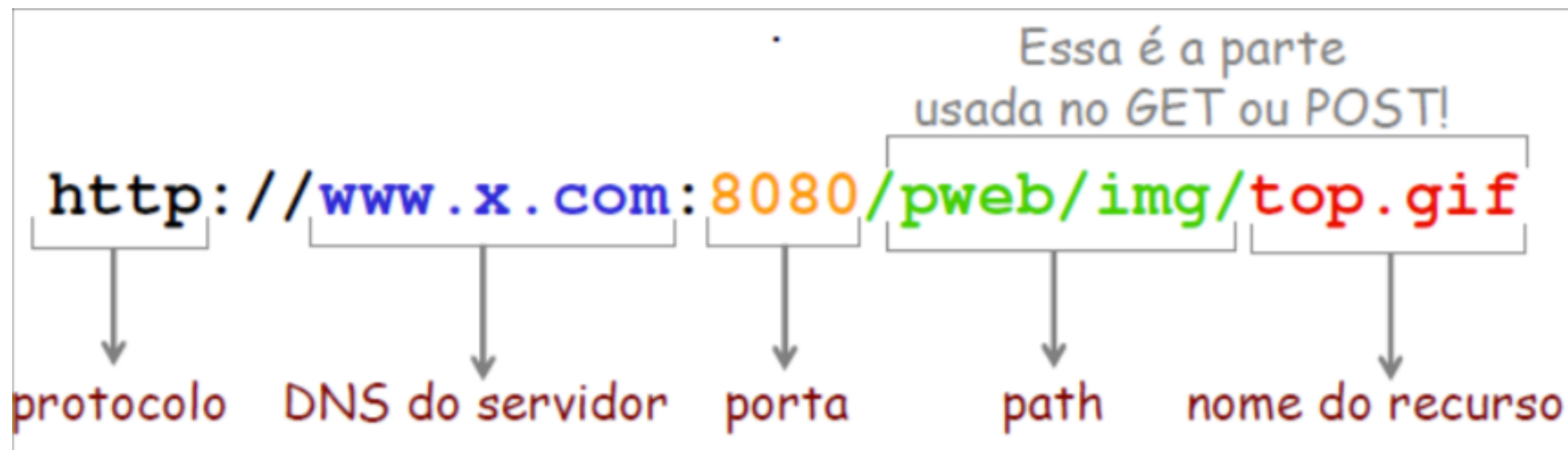
# Protocolo HTTP/HTTPS – Uso

- ▶ Com a utilização dos protocolos HTTP ou HTTPS podemos recuperar informações armazenadas no servidor de aplicação através dos principais métodos, como GET e POST.



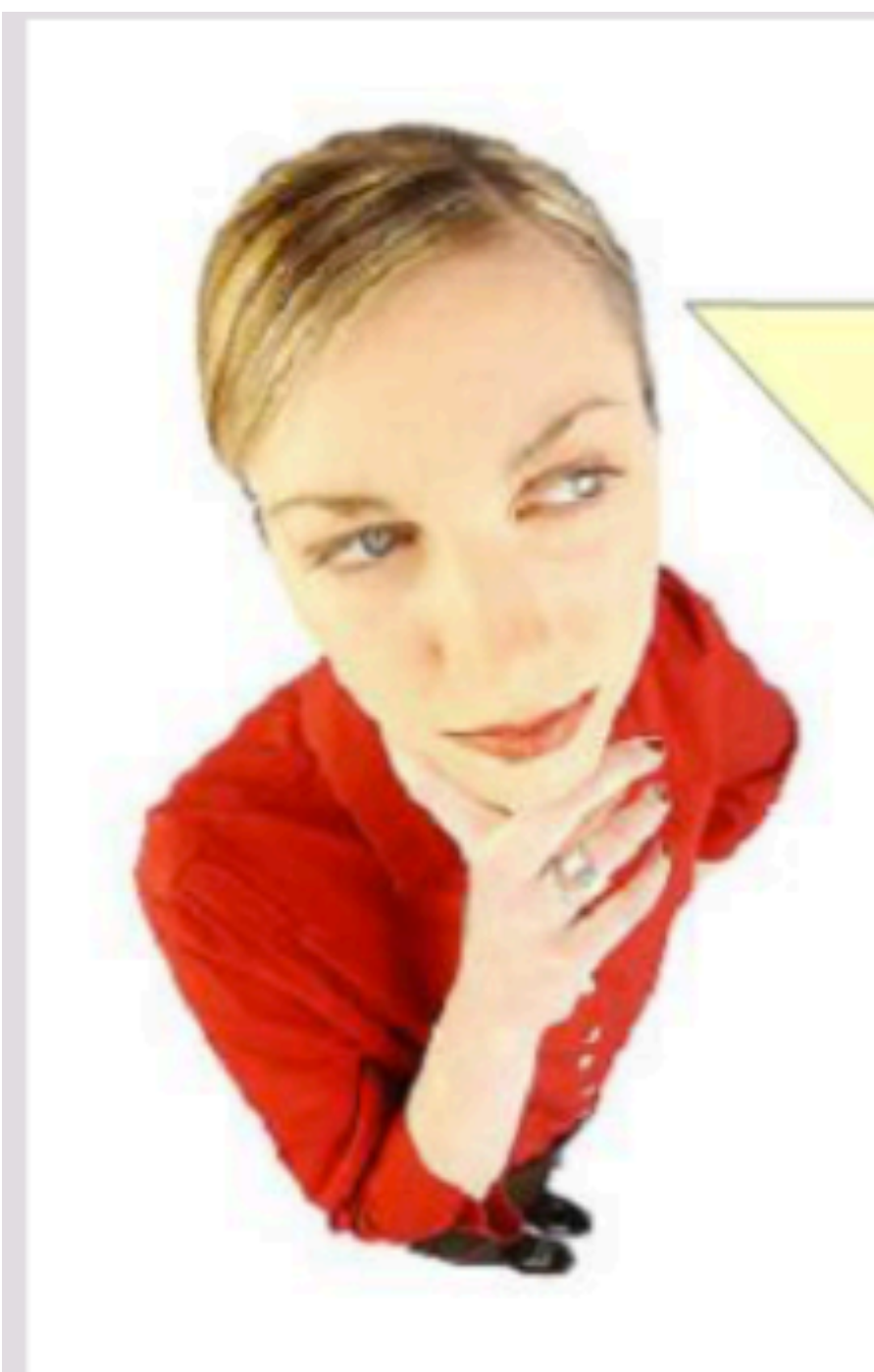


# Protocolo HTTP/HTTPS – URL Típica



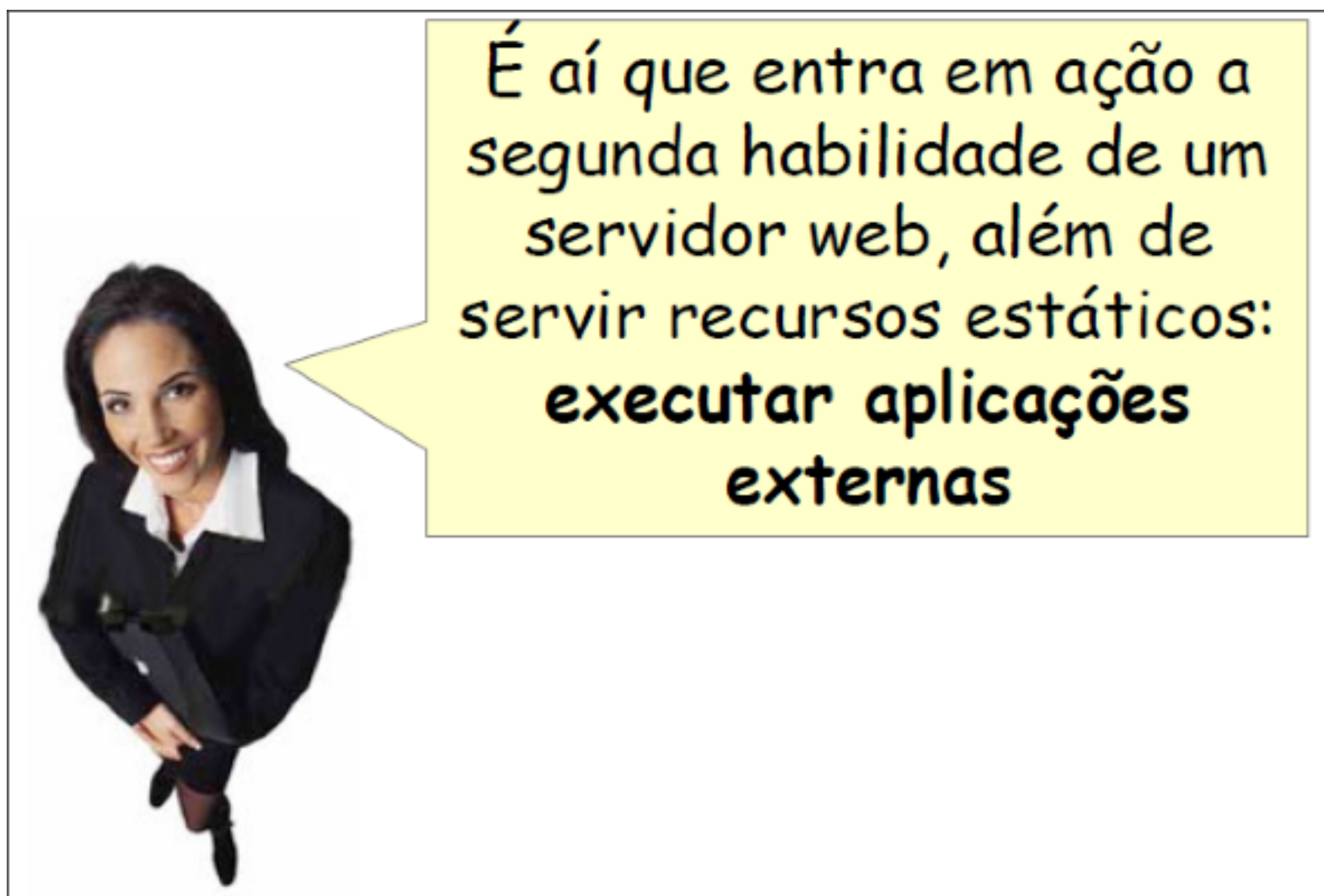
- ▶ O Path é contado a partir de um diretório base que é específico de cada servidor web (o do apache é htdocs)
- ▶ Se o recurso não estiver no path indicado ou o nome dele estiver errado, o servidor devolve uma resposta de erro.

# Protocolo HTTP/HTTPS – Conteúdo Estático



E se eu quiser obter uma informação gerada dinamicamente ou processar dados enviados por um formulário HTML?

# Protocolo HTTP/HTTPS – Conteúdo Estático





# Servidor Web – Funcionamento

- ▶ 1- O servidor web recebe um pedido de execução de uma aplicação (via POST ou GET)
- ▶ 2- O servidor web cria um processo para a aplicação e passa alguns parâmetros em variáveis especiais
- ▶ 3- A aplicação executa e devolve o resultado (código HTML) para o servidor web pela saída padrão
- ▶ 4- O servidor monta um frame de resposta HTTP, põe o HTML dentro dele e devolve para o cliente

# Implementação em Java – Servlets

- ▶ Características:

- ▶ São classes Java especiais que são acessadas a partir de um servidor Web
- ▶ Possui uma instância única, mas dispara múltiplas threads para cada requisição
- ▶ Executam sob um servidor Web que roda e têm toda a ajuda e proteção dele

- ▶ Exemplo!