

HTTP

Carlos Gustavo A. da Rocha

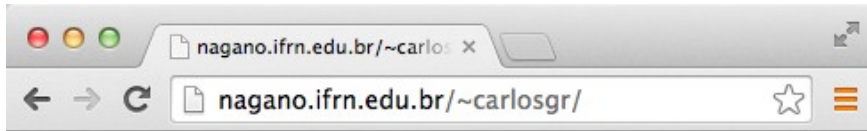
HTTP

- **HyperText Transfer Protocol**
 - Protocolo da camada de aplicação da web
 - Modelo cliente servidor
 - **Clientes:** Firefox, Chrome, Internet Explorer, Safari, ...
 - **Servidores:** Apache, Nginx, IIS, ...
 - Clientes e servidores “conversam” por meio da troca de mensagens HTTP
 - O protocolo HTTP define a **estrutura** destas mensagens, e o **modo** como cliente e servidor as trocam

HTTP

- **HTTP**
 - Boa parte do protocolo controla a transferência de páginas web entre clientes e servidores
 - Constituída de objetos (arquivos html, jpg, gif, wmv, ...)
 - Cada objeto possui uma URL **única em toda a internet**
 - URLs possuem duas partes
 - Nome do servidor
 - Caminho do objeto

HTTP



Olá mundo



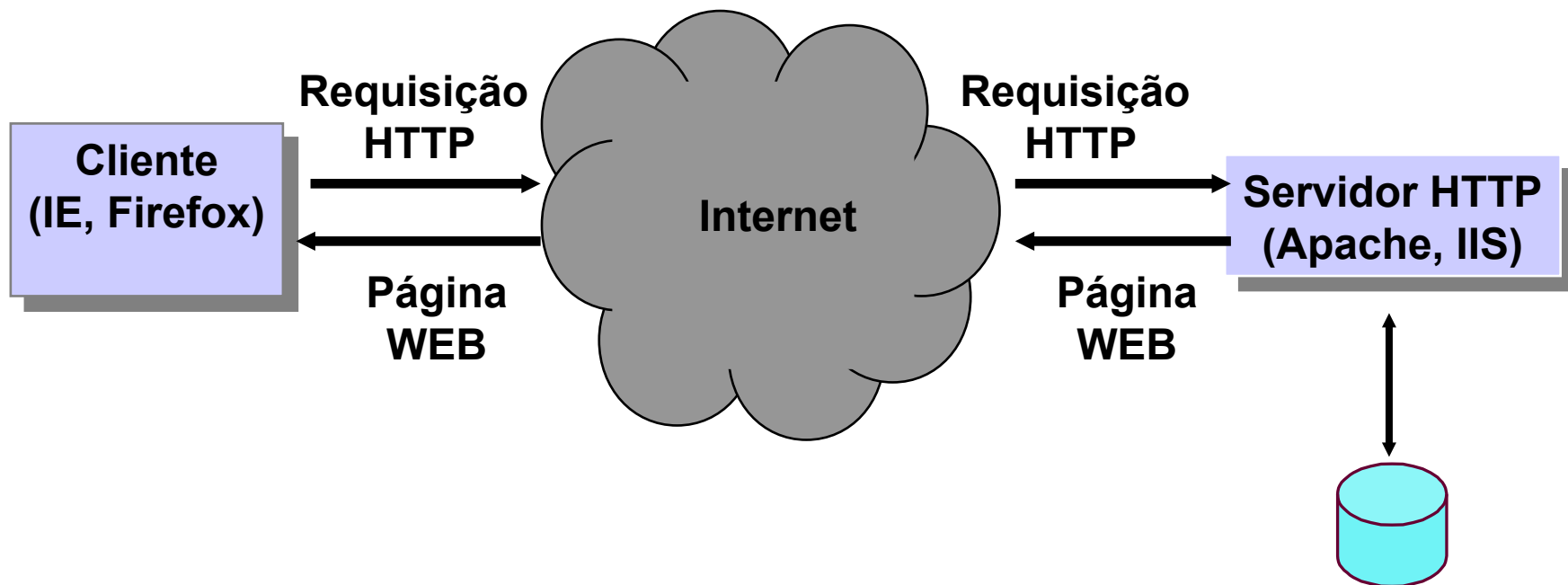
Google™

Exemplo de página web simples com 3 objetos
http://nome_servidor/caminho_objeto

<http://nagano.ifrn.edu.br/~carlosgr/index.html>
<http://nagano.ifrn.edu.br/~carlosgr/terra.jpg>
http://nagano.ifrn.edu.br/~carlosgr/Logo_25wht.gif

HTTP

- Componentes



HTTP

- Algumas características do protocolo
 - Duas versões foram padronizadas
 - HTTP 1.0 – 1996
 - HTTP 1.1 – 1999
 - São “compatíveis”. Cliente 1.0 acessa servidor 1.1, cliente 1.1 acessa servidor 1.0
 - Utiliza o protocolo de transporte TCP
 - Porta TCP 80

HTTP

- Transferência de uma página web (**não persistente**)

`http://nagano.ifrn.edu.br/~carlosgr`

1. Cliente inicia uma conexão TCP com o servidor `nagano.ifrn.edu.br` na porta 80
2. Cliente envia uma **mensagem de requisição HTTP** ao servidor solicitando o objeto inicial da página web (arquivo HTML)
3. Servidor recebe a requisição e responde com o objeto solicitado através de uma **mensagem de resposta HTTP**
4. Servidor encerra a conexão TCP, após a confirmação de que o cliente recebeu corretamente sua resposta
5. Cliente recebe a mensagem de resposta HTTP. Esta mensagem indica que o objeto “encapsulado” é um arquivo HTML. O cliente extrai o arquivo HTML da resposta, analisa seu conteúdo, e encontra referências para 2 outros objetos
6. Os passos 1 até 4 são repetidos para cada um dos objetos referenciados



tempo

HTTP

- Transferência de uma página web
 - A medida que o cliente recebe os objetos que formam uma página web, os apresenta na tela
 - Dois clientes (navegadores) distintos podem exibir uma página web de forma diferente
 - O HTTP nada tem haver com o modo como navegadores interpretam e exibem as páginas web
 - Define apenas a comunicação entre clientes e servidores

HTTP

- Conexões **não persistentes** X **persistentes**
 - Conexões não persistentes (padrão do HTTP 1.0)
 - Para cada objeto de uma página: **abertura de conexão; solicitação e transferência do objeto; encerramento da conexão**
 - Boa parte das páginas atuais possui centenas de objetos
 - Abrir centenas de conexões sequencialmente é muito lento; em paralelo consumiria muitos recursos do SO de clientes e servidores

HTTP

- Conexões **não persistentes** X **persistentes**
 - Conexões persistentes (padrão do HTTP 1.1)
 - Permite que várias solicitações e transferências de objetos sejam feitas **utilizando uma mesma conexão**
 - Além disso as requisições são feitas em paralelo para diversos objetos distintos.
 - Desta forma **se atinge alta velocidade, sem um consumo exagerado de recursos** em servidores e clientes

HTTP

- Transferência de uma página web (**persistente**)

`http://nagano.ifrn.edu.br/~carlosgr`

1. Cliente inicia uma conexão TCP com o servidor `nagano.ifrn.edu.br` na porta 80
2. Cliente envia uma **mensagem de requisição HTTP** ao servidor solicitando o objeto inicial da página web (arquivo HTML)
3. Servidor recebe a requisição e responde com o objeto solicitado através de uma **mensagem de resposta HTTP**
5. Cliente recebe a mensagem de resposta HTTP. Esta mensagem indica que o objeto “encapsulado” é um arquivo HTML. O cliente extrai o arquivo HTML da resposta; analisa seu conteúdo; encontra referências para 2 outros objetos e os requisita imediatamente, em paralelo, pela mesma conexão TCP.



tempo

HTTP

- Formato das mensagens – **requisição HTTP**
 - Escritas em texto ASCII (legíveis)
 - Possui uma ou mais linhas de texto
 - Primeira: **linha de requisição**
 - Possui três campos
 - Seguintes: **linhas de cabeçalho (opções)**
 - Possui dois ou mais campos

HTTP

- Examinando uma mensagem de **requisição**

```
GET /~carlosgr HTTP/1.1  
Host: nagano.ifrn.edu.br  
User-agent: Mozilla/4.0  
Accept-language: pt-br
```

← Linha de requisição
← Linhas de cabeçalho

HTTP

- Examinando uma mensagem de **requisição**

GET /~carlosgr HTTP/1.1

← Linha de requisição



Caminho do objeto solicitado

Versão do protocolo

Método (tipo da requisição). GET é o mais comum. Usado quando o Navegador solicita um objeto do servidor

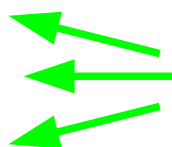
HTTP

- Examinando uma mensagem de **requisição**

Host: **nagano.ifrn.edu.br**

User-agent: **Mozilla/4.0**

Accept-language: **pt-br**



Linhas de cabeçalho

- ▶ Indica uma linguagem preferencial do objeto requisitado
- ▶ Indica o tipo de cliente, ou seja o navegador
- ▶ Indica o nome do servidor onde se deve buscar o objeto

HTTP

- Examinando uma mensagem de **requisição**
 - Linha de requisição
 - **Sempre** única
 - **Obrigatória**
 - Linhas de cabeçalho
 - **Não são obrigatórias**
 - **Indicam opções** relacionadas a requisição
 - Existem cerca de 50 opções disponíveis no HTTP1.1

HTTP

- Formato de mensagens – **respostas HTTP**
 - Possui até três seções
 - **Linha de estado**
 - Status da resposta (sucesso, erro, ...)
 - **Linhas de cabeçalho**
 - Informações sobre a resposta do servidor
 - **Dados** (objeto solicitado)
 - Arquivo HTML, imagem JPG, ...

HTTP

- Examinando uma mensagem de **resposta**

HTTP/1.1 200 OK ← Linha de status

Date: Fri, 27 Jun 2014 13:40:15 GMT

Server: Apache/2.2.8 (Ubuntu)

Last-Modified: Fri, 27 Jun 2014 13:24:29 GMT

ETag: "7dac1-c2-4fcd13aa37d40"

Accept-Ranges: bytes

Content-Length: 194

Content-Type: text/html

<html>

<body>

...

← Arquivo
(HTML)

Linhas de
cabeçalho

HTTP

- Examinando uma mensagem de **resposta**

HTTP/1.1 200 OK ← Linha de status

↓

Tudo OK, arquivo será enviado ao cliente

↓

Mostra que o servidor está utilizando HTTP/1.1

HTTP

- Examinando uma mensagem de **resposta**

Date: Fri, 27 Jun 2014 13:40:15 GMT

Server: Apache/2.2.8 (Ubuntu)

Last-Modified: Fri, 27 Jun 2014 13:24:29 GMT

...

...

Content-Length: 194

Content-Type: text/html

Tipo do arquivo na resposta

Tamanho do arquivo na resposta

Data e hora de modificação do arquivo existente no servidor

Tipo do servidor

Data e hora atuais

Linhas de
cabeçalho

HTTP

- Examinando uma mensagem de **resposta**
 - Linha de **Status**
 - **Também única e obrigatória**
 - Existem diversos status possíveis
 - 200 OK
 - 301 Moved Permanently
 - 403 Forbidden
 - 404 Not Found
 - 500 Internal Server Error

HTTP

- Examinando uma mensagem de **resposta**
 - Linhas de **cabeçalho**
 - Não obrigatórias, mas quase sempre estão presentes
 - Contém **informações diversas** sobre
 - O servidor
 - O conteúdo dos dados existentes na resposta
 - ...

HTTP

- Simulando um cliente HTTP

1 – Abra um terminal Linux (ou prompt do DOS) e digite:

```
telnet nagano.ifrn.edu.br 80
```

2 – Digite uma requisição HTTP:

```
GET /~carlosgr/index.html HTTP/1.1
```

```
Host: nagano.ifrn.edu.br
```

3 – tecele <enter> duas vezes, e analise a resposta do servidor

HTTP

- Cookies
 - Tecnologia que permite que servidores HTTP identifiquem (e monitorem) seus usuários
 - Componentes
 - **Linhas de cabeçalho**, utilizadas nas mensagens de requisição e resposta HTTP
 - **Arquivos** mantidos na máquina do usuário e gerenciados por seu navegador
 - Um **banco de dados**, mantido no servidor HTTP



HTTP

- Cookies - funcionamento

- 1 – Usuário acessa um site que use cookies pela primeira vez
- 2 – Servidor responde incluindo em sua primeira resposta o cabeçalho
Set-cookie: seguido de alguma identificação única. Por exemplo:

Set-cookie: 1678453

- 3 – O navegador armazena esta informação em um arquivo texto
- 4 – Todas as vezes que o usuário **voltar a acessar** este site, o Navegador irá incluir em suas requisições a linha:

Cookie: 1678453

HTTP

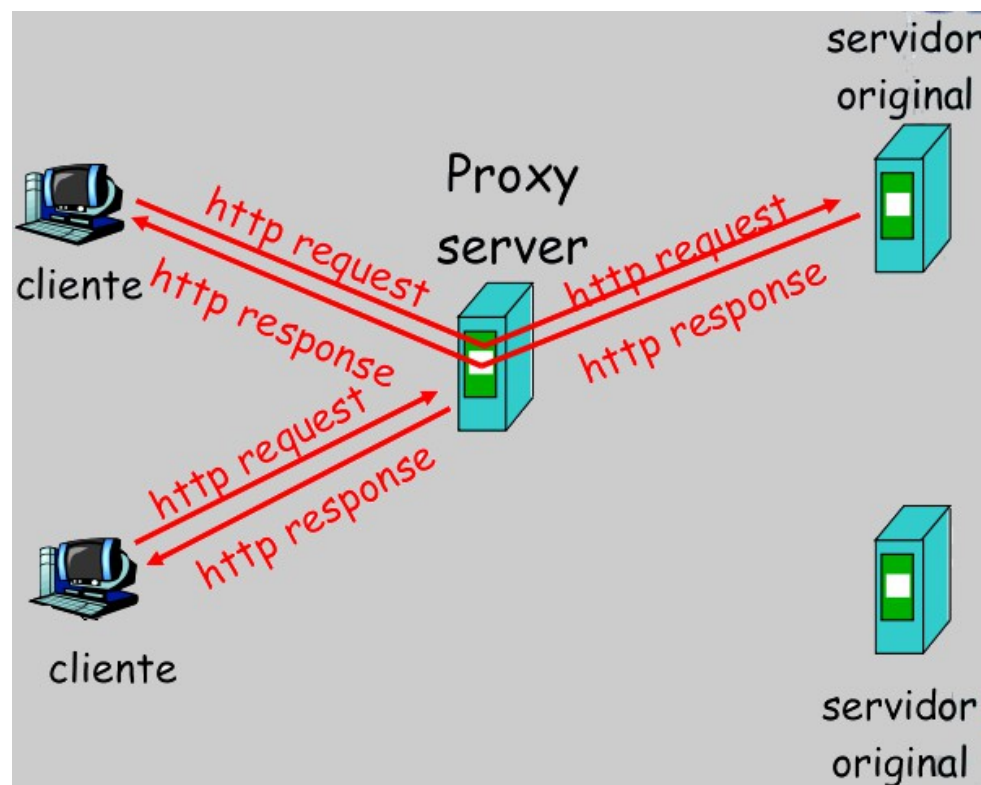
- Cookies – Informações
 - Como um cookie pode identificar cada usuário de um site, com eles podemos
 - Saber que páginas do site um usuário visita
 - Em que horários
 - Apresentar páginas ou propaganda personalizada
 - ...

HTTP

- Caches WEB (Proxy) - Funcionamento

- 1 - **Usuário configura o navegador** para o acesso ser feito usando um Cache WEB
- 2 - Navegador envia todos os pedidos HTTP para o Cache WEB

- Se o objeto existe no Cache WEB ele retorna o objeto
- Senão, o Cache WEB solicita objeto do servidor original e então o envia ao cliente armazenando uma cópia



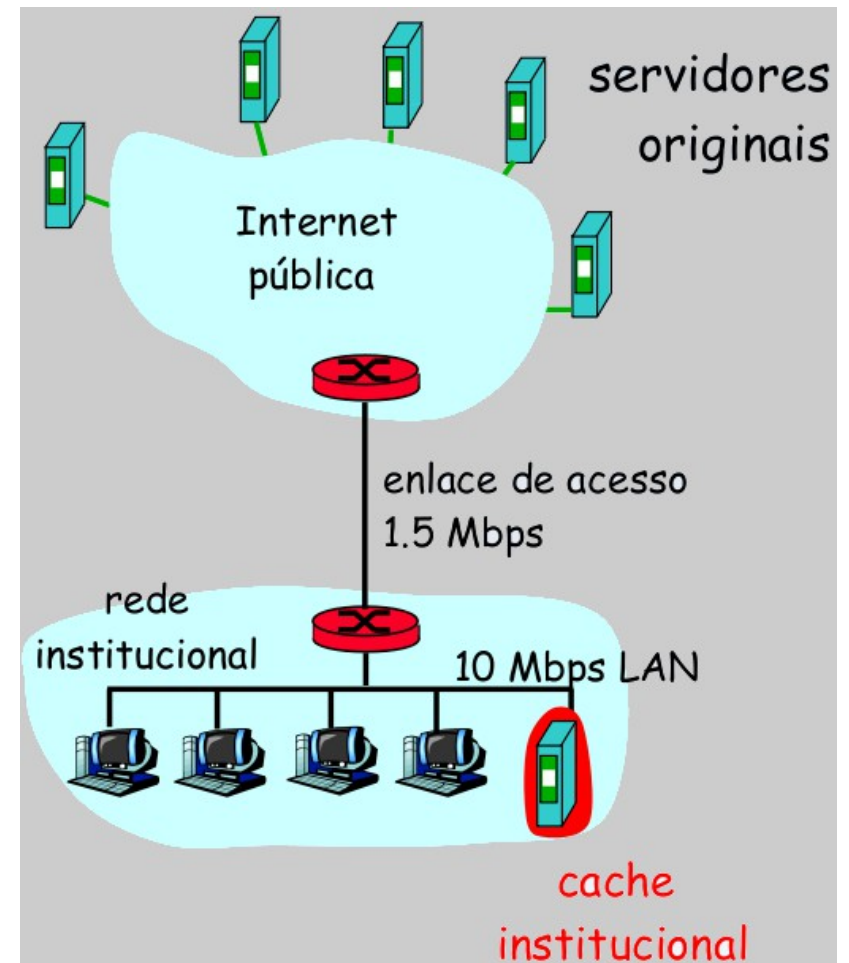
HTTP

- Caches WEB – Uso
 - Funcionam como **um ponto intermediário** entre o cliente e o servidor HTTP
 - São instalados por provedores, instituições, etc
 - **Reduz o tempo de resposta** para a requisição do cliente
 - **Reduz o tráfego** no link de internet da instituição
 - Possibilita uma série de controles de acesso e **filtragens**
 - Possibilita a geração de uma série de **estatísticas** de uso da Internet pelos usuários



HTTP

- Caches WEB – Uso
 - Um cache bem configurado terá boa parte das páginas armazenadas localmente
 - Possibilita um uso ótimo do link de acesso externo
 - Normalmente possui um custo bastante elevado



HTTP

• GET Condicional

Razão: não enviar objetos se a versão que o cliente já possui está atualizada

Cliente: especifica data da versão armazenada na requisição HTTP

If-modified-since:

Servidor: resposta não contém objeto se a cópia esta atualizada:

HTTP/1.0 304 Not Modified

Cliente

Servidor

