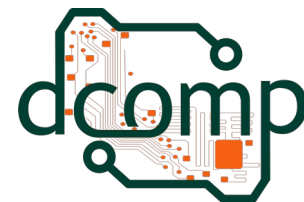




Universidade Federal do Espírito Santo  
Centro de Ciências Exatas, Naturais e da Saúde  
Departamento de Computação



# Fundamentos WEB

## **Fundamentos de Programação WEB**

Site: <http://www.jeiks.net/fundpweb>

E-mail: [jacson.silva@ufes.br](mailto:jacson.silva@ufes.br)

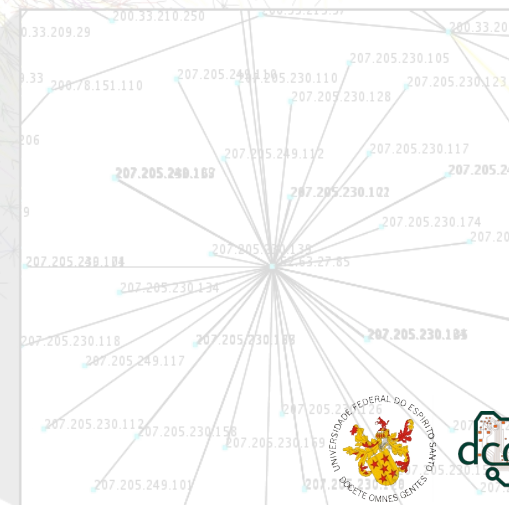
- Conteúdo:
  - Introdução
  - Histórico
  - Web
  - HTTP.





# Web

- Até a década de 1990, a Internet era utilizada
  - por pesquisadores, acadêmicos e universitários,
  - para troca de arquivos e
  - para correio eletrônico.
- Então, no início dessa década, iniciou-se a
  - *World Wide Web*:
    - Rede mundial de computadores
    - *ou teia de alcance mundial*
- Que atraiu a atenção de todos.



A Wikipédia tem um bom material relacionado, acesse:  
[https://pt.wikipedia.org/wiki/História\\_da\\_Internet](https://pt.wikipedia.org/wiki/História_da_Internet)



# Protocolo HTTP

- *Hypertext Transfer Protocol* (HTTP)

## Protocolo de Transferência de Hipertexto

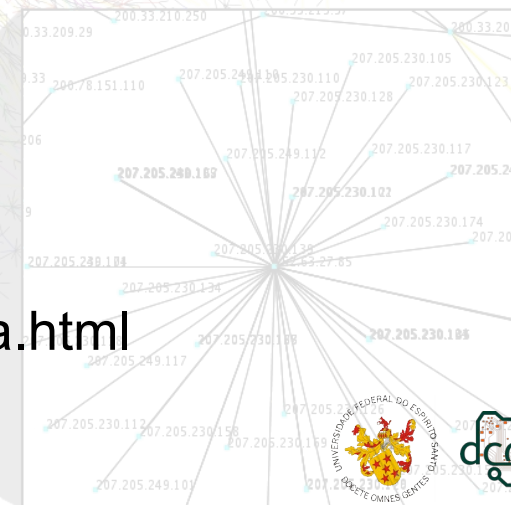
- Protocolo de camada de aplicação da Web;
- Definido pela RFC 1945 e RFC 2616;
- É implementado por um servidor e um cliente.
- Os dois sistemas comunicam-se com mensagens HTTP;
- O protocolo HTTP define a estrutura dessas mensagens e o modo de trocá-las.





# A página Web e o HTTP

- Uma página Web é constituída de objetos.
- O Objeto é um arquivo, como: html, jpg, applet, etc.
- Uma página Web geralmente é um
  - *arquivo-base HTML*, que
  - referencia diversos outros objetos.
- O pedido da página Web é realizado por uma URL:
  - URL: Uniform Resource Locator;
  - Ex.: `www.ufes.br/pagina.html`
  - Onde:
    - Hospedeiro: `www.ufes.br`
    - Caminho da página Web requisitada: `pagina.html`





# Página Web e HTTP

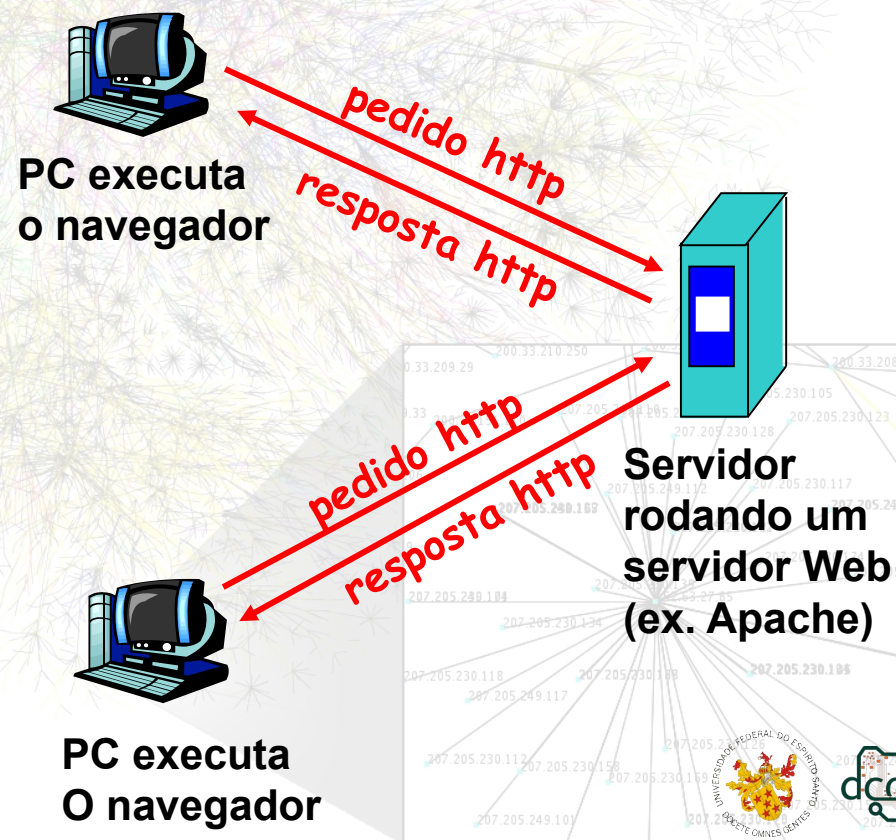
- Cada objeto é endereçável por uma URL:

www.dominioweb.com/pasta/figura.gif

nome do hospedeiro servidor

nome do caminho

- O HTTP utiliza o TCP como protocolo de transporte





# Protocolo HTTP

- HTTP é sem estado
  - O servidor não mantém informação sobre pedidos anteriores do cliente, exceto nos logs.
- Uma conexão HTTP é:
  - Não persistente:
    - No máximo um objeto é enviado numa conexão TCP;
    - HTTP/1.0 usa o HTTP não persistente.
  - Persistente:
    - Múltiplos objetos podem ser enviados sobre uma única conexão TCP entre cliente e servidor;
    - HTTP/1.1 usa conexões persistentes no seu modo padrão.





# Formato da mensagem HTTP

- Mensagem de pedido

GET /somedir/page.html HTTP/1.0

Host: www.someschool.edu

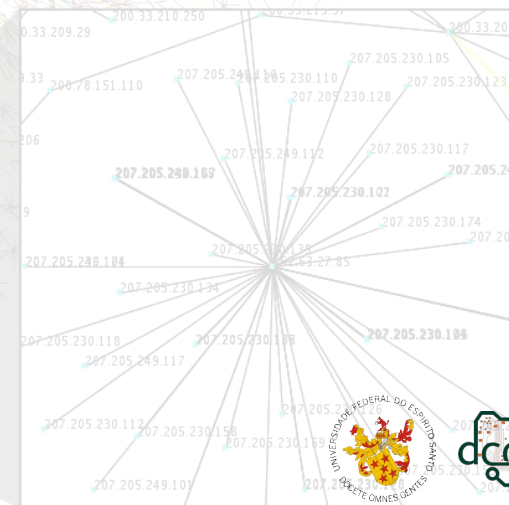
User-agent: Mozilla/4.0

Connection: close

Accept-language:fr

<enter>

<enter>





# Exemplos



- **Comando:** `telnet www.ufes.br 80`

GET /index.html HTTP/1.1

Host: www.ufes.br

*Pressione ENTER duas vezes*

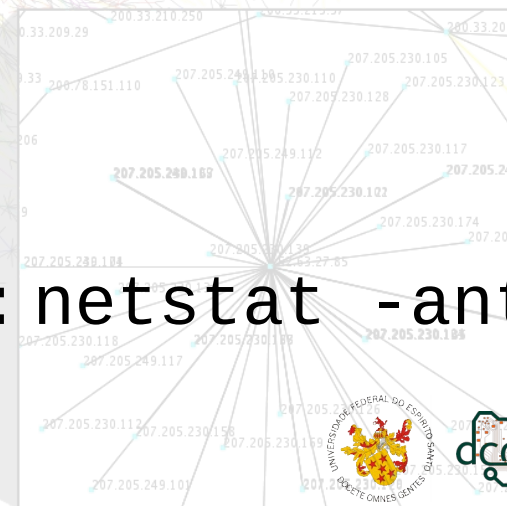
- **Comando:** `telnet www.ufes.br 80`

GET /index.html HTTP/1.0

Host: www.ufes.br

*Pressione ENTER duas vezes*

- **No Linux, para ver as conexões, use:** `netstat -antp`





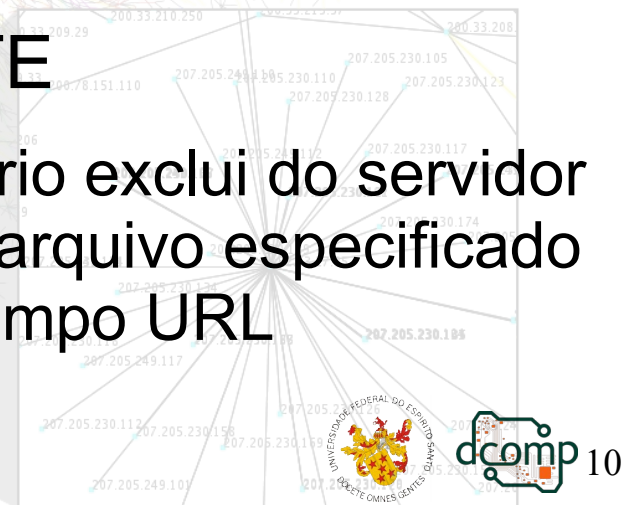
# Tipos de métodos

# HTTP/1.0

- GET
  - Usuário requisita um objeto
- POST
  - Usuário preenche formulário (colocado no corpo da mensagem)
- HEAD
  - Pede para o servidor não enviar o objeto requerido junto com a resposta (usado p/ depuração)

# HTTP/1.1

- GET, POST, HEAD
- PUT
  - Upload de arquivo contido no corpo da mensagem para o caminho especificado no campo URL
- DELETE
  - Usuário exclui do servidor Web arquivo especificado no campo URL









# Formato da mensagem HTTP

- Mensagem de resposta:

# HTTP/1.1 200 OK

# Connection close

Date: Thu, 06 Aug 1998 12:00:15 GMT

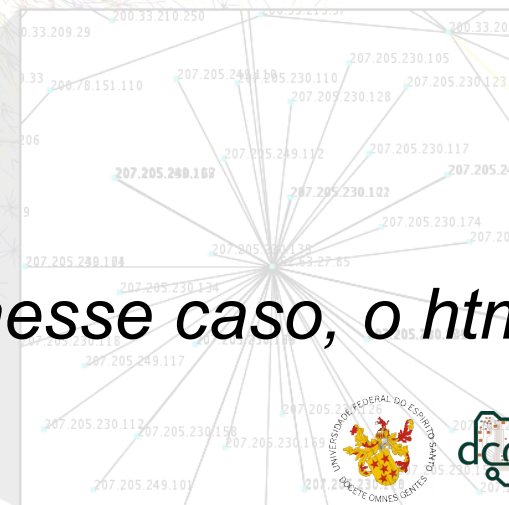
Server: Apache/1.3.0 (Unix)

Last-Modified: Mon, 22 Jun 1998 .....

Content-Length: 6821

Content-Type: text/html

*todo o conteúdo do objeto pedido, nesse caso, o html*



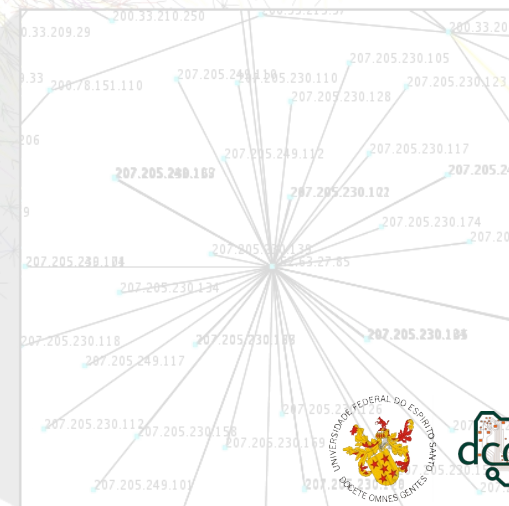






# Cookies

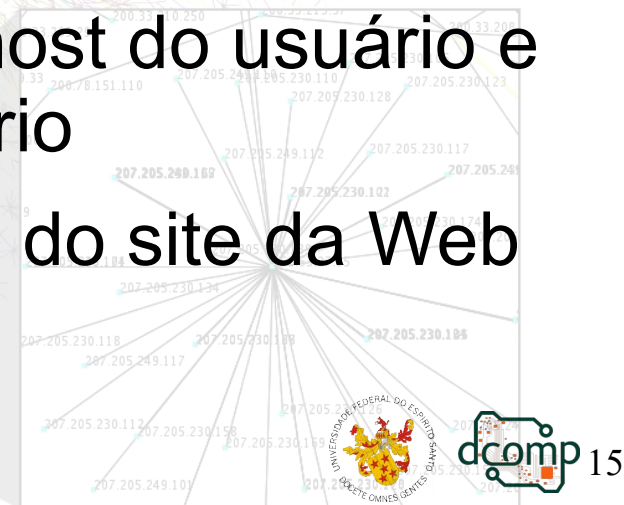
- Utilizados para manter o “estado” da conexão
- São armazenados no disco rígido do cliente com dados do usuário.
- Permitem que os sites identifiquem e monitorem os seus usuários.
- Utilizados por muitos sites para manter a “sessão”.





# Cookies

- Quatro componentes:
  1. linha de cabeçalho do cookie na mensagem de resposta HTTP;
    - Set-cookie: 1678
  2. linha de cabeçalho do cookie na mensagem de pedido HTTP
    - Cookie: 1678
  3. arquivo do cookie mantido no host do usuário e gerenciado pelo browser do usuário
  4. Banco de Dados (BD) de apoio do site da Web





# Cookies

- Tarefas realizadas com cookies:
  - Autorização após armazenamento do registro da pessoa;
  - Registro da lista de compras no Ecommerce;
  - Sugestões - recomendar produtos;
  - Estado da sessão do usuário (Web email) – identificação do usuário;
  - Eles armazenam coisas que você acessou, sites que você viu.





# Cookies

- Segurança:
  - permitem que os sites tenham informações do usuário;
  - pode fornecer nome e e-mail do usuário para os sites;
  - mecanismos de busca usam redirecionamento e cookies para aprender ainda mais sobre o usuário
  - agências de propaganda obtêm perfil a partir dos sites visitados e oferecem produtos perturbando os usuários





# Vamos ver as mensagens enviadas ao servidor



- Edite um arquivo e adicione as linhas:

```
#!/bin/bash
```

```
while read linha;do
```

```
    echo $linha >&2
```

```
done
```

- Execute:

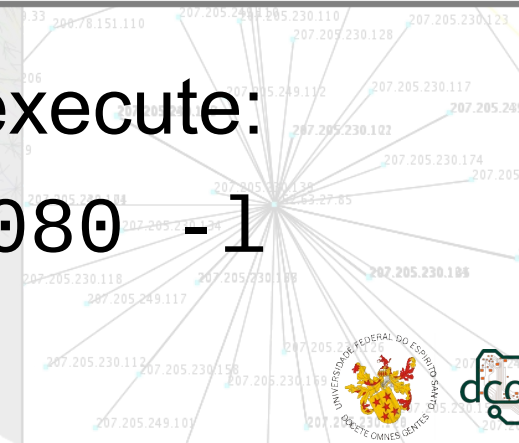
```
chmod +x server.sh
```

- No Linux, com *netcat tradicional*, execute:

```
nc -e $PWD/server.sh -p 8080 -l
```

**Após isso, entre no Navegador de Internet e acesse: <http://localhost:8080>**

**Veja no terminal as mensagens enviadas pelo Navegador ao Servidor.**





E agora, um servidor um pouco mais elaborado...  
Arquivo: [servidor\\_web\\_bash.sh](#)



Servidor Web em bash:

1. Execute-o no terminal com o *nc*:

```
nc -e $PWD/server.sh -p 8080 -l
```

2. Acesse essa porta 8080 usando o navegador:  
<http://localhost:8080>

3. Quais foram as mensagens trocadas entre o servidor e o navegador?

