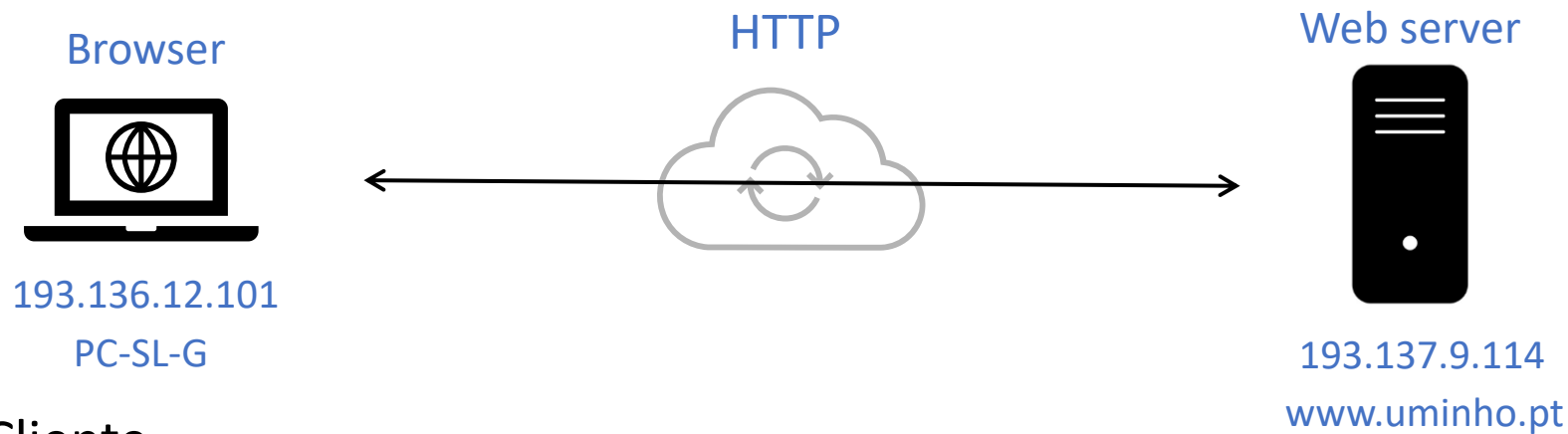


Tópicos avançados

Aplicações web baseadas em FastCGI

Multithreading

Comunicações na web



- Cliente

- Liga-se ao servidor
- Envia mensagens com pedidos

- Servidor

- Espera por ligações
- Envia resposta a cada pedido



Modelo de referência OSI

Aplicação
Apresentação
Sessão
Transporte
Rede
Ligação
Física

HTTP, DNS

TCP, UDP
IP

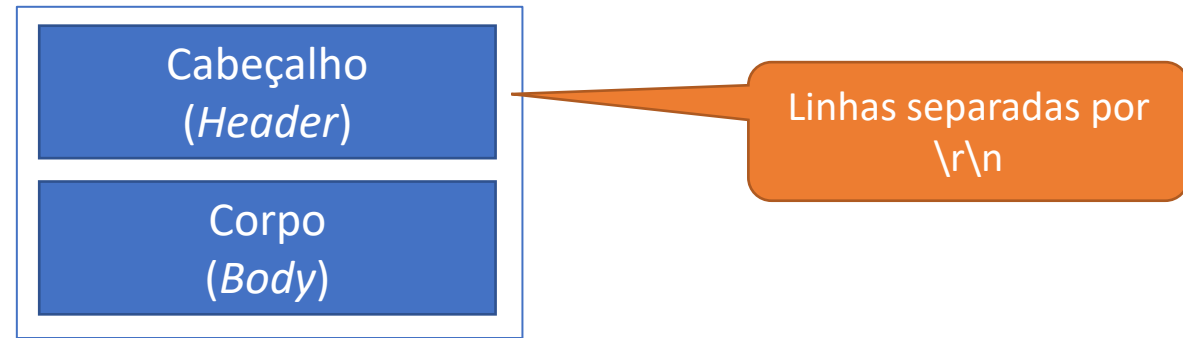
Identificação

- Endereço IP
 - *Loopback*: 127.0.0.1
- Portas de protocol TCP/UDP (HTTP 80, HTTPS 443, FTP 21)



HTTP

- Protocolo dos sítios web
 - Formato das mensagens



- Pedido

```
GET /caminho HTTP/1.1
Accept: */*
Host: www.exemplo.com
Content-Length: 12
```

Alguns dados

- Resposta

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Content-Type: text/plain
Content-Length: 17
```

Dados da resposta

Linha em
branco
(\r\n)

HTTP

- Métodos (ou tipo de pedidos)

- GET

- POST



criar

- PUT



actualizar

- DELETE

- Locais dos pedidos para enviar dados

- Corpo

- Query-string



```
GET /caminho?campo1=valor1&campo2=valor2 HTTP/1.1
Accept: */*
Host: www.exemplo.com
Content-Length: 51
```

Dados no corpo do pedido,
tipicamente com POST e PUT

HTML

- Formato de dados das páginas web
 - Os *browsers* sabem apresentar

```
<html>
  <head>
    <title>This is a title</title>
  </head>
  <body>
    <p>Hello world!</p>
  </body>
</html>
```

Separação por linhas
opcional (basta \n)

[https://www.w3schools.com/
html](https://www.w3schools.com/html)

HTML

- Entrada de dados

URL relativa ou absoluta
para script/executável

Ou GET (por defeito)

```
<p>Fill in the form</p>
<form action="/caminho/para/servico" method="post">
  <label for="name">Name:</label>
  <input type="text" id="name" name="name"><br>
  <label for="number">Grade (between 1 and 5):</label>
  <input type="number" id="number" name="grade" min="1" max="5"><br>
  <label>Gender:</label><br>
  <input type="radio" id="male" name="gender" value="male">
  <label for="male">Male</label><br>
  <input type="radio" id="female" name="gender" value="female">
  <label for="female">Female</label><br>
  <input type="reset">
  <input type="submit" value="Submit">
</form>
```

- GET envia na query-string
?name=jose&grade=5 ...
- POST envia no corpo
name=jose&grade=5 ...

FastCGI

- Permite a um servidor fornecer conteúdo de forma dinâmica
 - No Apache
 - Instalar módulo FastCGI

- Estrutura tipo

- Inicialização
- Ciclo de recepção de pedidos
 - Processamento e envio da resposta
 - Opcionalmente, terminar o pedido/resposta



```
graph LR; A[FCGX_InitRequest(...);] --> B[FCGX_Accept(...);]; B --> C[FCGX_Finish(...);];
```

FCGX_InitRequest(...);

FCGX_Accept(...);

FCGX_Finish(...);

FastCGI

- Pedido HTTP
 - Cabeçalho -> Variáveis de ambiente
 - Corpo -> entrada padrão (cin)
- Resposta HTTP
 - Saída padrão
 - Status code 200 por defeito, ou especificado
 - Directamente na saída padrão "Status: <code>" (pode-se colocar o que quiser na resposta)
 - Utilizando a API (determina o resto da resposta em caso de erro)
 - FCGX_Stream ss;
 - FCGX_SetExitStatus(404, &ss);

Não esquecer:

```
cout << "cabeçalho\r\n";  
cout << "\r\n";  
cout << "corpo";
```