

---

# PROMPT 2011

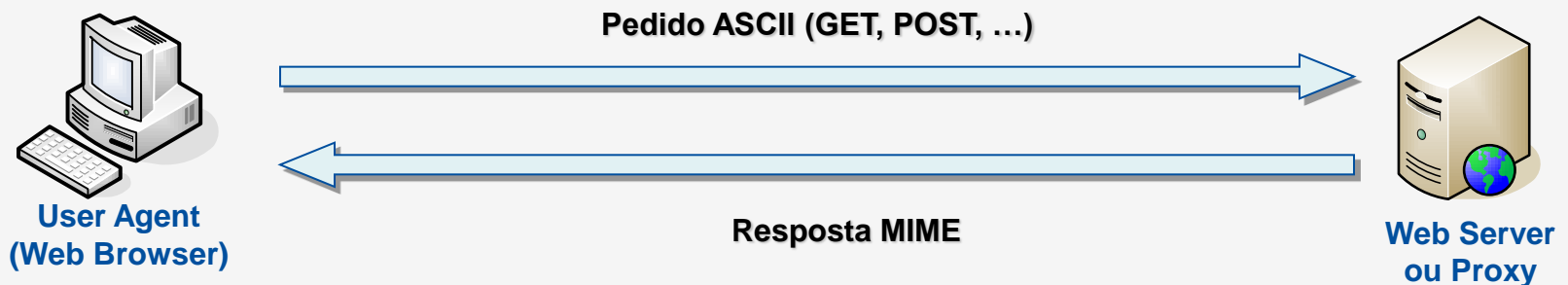
## Módulo 3 – Aplicações Web

### Hypertext Transport Protocol (HTTP)

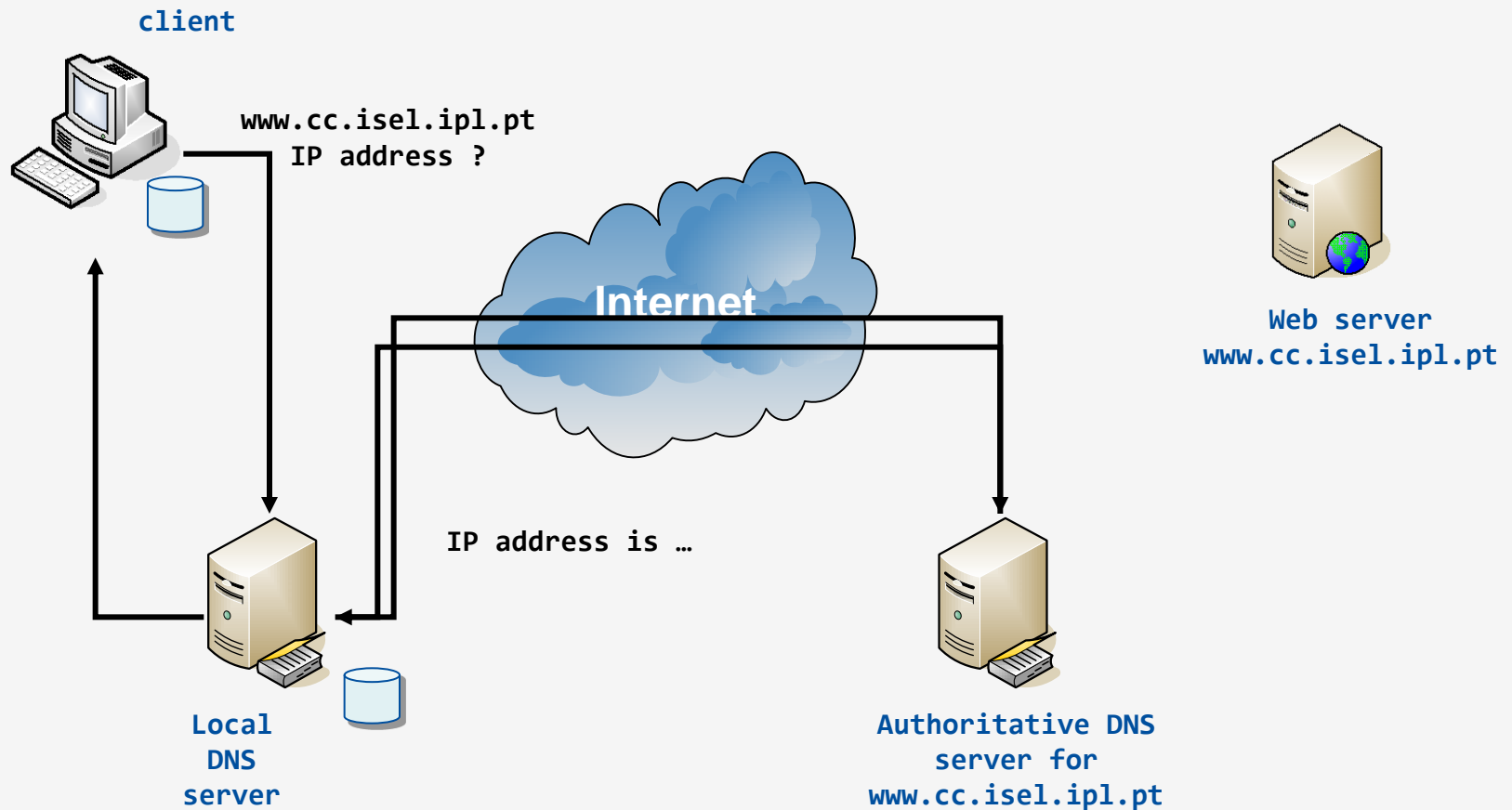
---

# Hypertext Transfer Protocol (HTTP)

- Protocolo usado para **transportar informação na *World Wide Web***
  - Objectivo original: Transferir documentos em hiper-texto (HTML)
- Evolução do protocolo coordenada pelo W3C
  - RFC 1945 - Versão 1.0
  - RFC 2616 - Versão 1.1 (usada actualmente)
- Modelo de interacção: Pedido/Resposta
  - Principal característica: **stateless**
- Servidor recebe pedidos em TCP no porto 80 (normalmente)



# Pedido no *browser*



# Pedido HTTP

## Sintaxe

```
<COMMAND> <Request-URI> HTTP/1.X <crLf>
{<Header>: <Value> <crLf>}+
<crLf>
[<data>]
```

<Request-URI> - Path absoluto relativamente à raiz do servidor Web, ou URI (URL) absoluto.

## Exemplo

```
GET /docu2.html HTTP/1.1
Accept: */* (outros exemplos: text/plain; text/html; image/gif )
User-Agent: Mozilla4.0 (compatible; MSIE 5.0; Windows NT 5.0)
Host: www.isel.ipl.pt
From: montulli@www.cc.ukans.edu
    * a blank line *
```

# Resposta HTTP

## Sintaxe

```
HTTP/1.X <result-code> [<message>] <crLf>
{ <Header>: <Value> <crLf>}+
<crLf>
[<data>]
```

## Exemplo

```
HTTP/1.1 200 OK
Date: Wednesday, 12-Jul-06 15:04:12 GMT
Server: Apache/1.3.6 (Unix) (Red Hat Linux)
MIME-version: 1.0
Last-modified: Monday, 11-Jul-06 23:33:16 GMT
Content-type: text/html
Content-length: 2345
    * a blank line *
<HTML><HEAD><TITLE> . . . </TITLE> . . .etc.
```

# Pedidos HTTP

Tipo de pedidos ( <i>Request Methods</i> )	Descrição
<b>GET</b>	Pedido a representação de um recurso (método mais comum)
<b>HEAD</b>	Pedido idêntico ao GET mas o corpo da resposta não é enviado (apenas os <i>headers</i> )
<b>POST</b>	Envio de dados para um recurso. Os dados vão no corpo do pedido
<b>PUT</b>	Actualiza a representação de um recurso
<b>DELETE</b>	Remove um recurso
<b>TRACE</b>	Retorna o eco do pedido
<b>OPTIONS</b>	Retorna os métodos suportados pelo servidor

# Códigos de resposta do servidor

Classe das  
respostas

1xx Informação  
2xx Sucesso  
3xx Redireccionamento  
4xx Erro Cliente  
5xx Erro Servidor

## 3xx Redirection

*The client must take additional action to complete the request.*

300: Multiple Choices

**301: Moved Permanently**

302: Moved Temporarily (HTTP/1.0)

**302: Found (HTTP/1.1)**

## 1xx Informational

*Request received, continuing process.*

100: Continue

101: Switching Protocols

## 2xx Success

*The action was successfully received, understood, and accepted.*

**200: OK**

201: Created

202: Accepted

203: Non-Authoritative Information

204: No Content

205: Reset Content

206: Partial Content

## 4xx Client Error

*The request contains bad syntax or cannot be fulfilled.*

**400: Bad Request**

**401: Unauthorized**

402: Payment Required

403: Forbidden

**404: Not Found**

405: Method Not Allowed

406: Not Acceptable

407: Proxy Authentication Required

408: Request Timeout

## 5xx Server Error

*The server failed to fulfill an apparently valid request.*

**500: Internal Server Error**

501: Not Implemented

502: Bad Gateway

503: Service Unavailable

504: Gateway Timeout

505: HTTP Version Not Supported

509: Bandwidth Limit Exceeded

# Cabeçalhos MIME (1)

---

- **Geral:** informação não relacionada com a entidade a transferir
  - **Date:** Tue, 15 Nov 1994 08:12:31 GMT
  - **MIME-Version:** 1.0
- **Opções cliente:** permite ao cliente passar informações acerca do pedido
  - **Accept:** \*/\*
  - **User-Agent:** Lynx/2.2 libwww/2.14
  - **Range:** bytes=0-499
  - **If-Modified-Since:** Thu, 25 Sep 1997 15:01:26 GMT
  - **Cookie:** EGSOFT\_ID=193.137.220.2-640807376.29152144; expires=Fri, 31-Dec-1997 0:00:00 GMT;path=/



## Cabeçalhos MIME (2)

---

- **Resposta:** informação sobre o servidor
  - **Server:** Apache/1.2b7
  - **Content-Range:** bytes=0-499/2000
  - **Set-Cookie:** EGSOFT\_ID=193.137.220.2-640807376.29152144; expires=Fri, 31-Dec-1997 0:00:00 GMT;path=/
- **Entidade:** informação sobre os dados entre cliente e servidor
  - **Last-Modified:** Thu, 25 Sep 1997 15:01:26 GMT
  - **Content-Length:** 6575
  - **Content-Type:** text/html
  - **Expires:** Thu, 25 Sep 1997 15:01:26 GMT

# GET

- Pede um recurso (ficheiro, CGI, etc.)
- **Corpo sempre vazio**

## Pedido

```
GET /index.html HTTP/1.1
User-Agent: Mozilla/2.02Gold (WinNT; I...)
Host: www.ora.com
Accept: image/gif, image/x-xbitmap, image/jpeg
* a blank line *
```

## Resposta (exemplo)

```
HTTP/1.1 200 Document follows
Date: Fri, 20 Sep 1996 08:17:58 GMT
Server: NCSA/1.5.2
Last-Modified: Mon, 17 Jun 1996 21:53:08 GMT
Content-Type: text/html
Content-Length: 2482
* a blank line *
... corpo do documento ...
```

---

# demo

**Realizar pedido  
HTTP GET**

- 
- Utilizar a aplicação `telnet` para realizar pedido a determinado servidor Web

# HEAD

- Pede de informações sobre o documento e não o documento em si
  - Data da última modificação, dimensão, tipo, etc.
- **Corpo sempre vazio** (pedido e resposta)

## Pedido

```
HEAD /default.htm HTTP/1.1
Accept: */*
Host: www.deetc.isel.ipl.pt
* a blank line *
```

## Resposta

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
MicrosoftOfficeWebServer: 5.0_Pub
X-Powered-By: ASP.NET
Date: Tue, 07 Oct 2008 21:46:22 GMT
Content-Type: text/html
Accept-Ranges: bytes
Last-Modified: Thu, 18 Sep 2008 19:59:03 GMT
ETag: "1000c919c91:df5"
Content-Length: 9870
```

# POST (1)

- **O cliente envia dados ao servidor no pedido (no corpo)**
- O conteúdo do pedido é passado à aplicação que processa os dados

## Pedido

```
POST /cgi-bin/post-query HTTP/1.1
Accept: */*
User-Agent: Lynx/2.2 libwww/2.14
From: grobe@www.cc.ukans.edu
Content-type: application/x-www-form-urlencoded
Content-length: 150
Host: www.isel.pt

org=Academic%20Computing%20Services
&users=10000
&browsers=lynx
&browsers=cello
&browsers=mosaic
&others=MacMosaic%2C%20WinMosaic
&contact=Michael%20Grobe%20grobe@kuhub.cc.ukans.edu
```

## Resposta (incompleta)

Content-Type: text/html

```
<h1>Query Results</h1>
```

You submitted the following name/value pairs:

```
<ul>
```

```
<li>org = Academic Computing Services </li>
```

```
<li>users = 10000 </li>
```

```
<li>browsers = cello </li>
```

```
<li>browsers = lynx </li>
```

```
<li>browsers = xmosaic </li>
```

```
<li>others = Mac Mosaic, Win Mosaic </li>
```

```
<li>contact = Michael Grobe grobe@kuhub.cc.ukans.edu </li>
```

```
</ul>
```

# HTTP Headers (1)

---

- **General headers**

- Cache-Control, Connection, Date, Pragma, Transfer-Encoding, Upgrade, e Via.

- **Entity headers**

- Allow, Content-Base, Content-Encoding, Content-Language, Content-Length, Content-Location, Content-MD5, Content-Range, Content-Type, ETag, Expires, e Last-Modified.

- **Request headers**

- Accept, Accept-Charset, Accept-Encoding, Accept-Language, Authorization, From, Host, If-Modified-Since, If-Match, If-None-Match, If-Range, If-Unmodified-Since, Max-Forwards, Proxy-Authorization, Range, Referer, e User-Agent.

- **Response headers**

- Accept-Ranges, Age, Location, Proxy-Authenticate, Public, Retry-After, Server, Vary, Warning, e WWW-Authenticate.

## HTTP Headers (2)

- Request headers

Header	Descrição
Accept	Formatos aceites pelo cliente (text/html, etc...)
Accept-Encoding	Indica se o cliente suporta compressão dos dados
If-Modified-Since	Pedido condicional (recebe resposta 304)
User-Agent	Assinatura do <i>browser</i> que realiza o pedido
Host	Diferencia vários <i>sites</i> na mesma máquina (obrigatório)

- Response headers

Header	Descrição
Accept-Ranges	Indica se o servidor aceita pedidos em blocos (# bytes ou none)
Server	Assinatura do servidor



## HTTP Headers (3)

---

- Entity headers (sobre o conteúdo)

Header	Descrição
ETag	Identificador único (gerado pelo servidor) do recurso
Last-Modified	Altura na qual o recurso foi alterado
Content-Length	Número de bytes ocupados pelo corpo ( <i>body</i> ) da resposta
Content-Type	Tipo de conteúdo (text/html, text/plain, etc...)

---

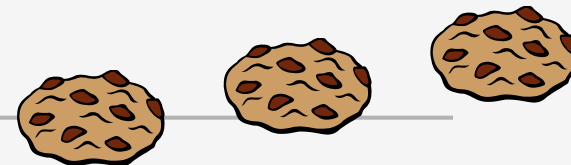
# MANUTENÇÃO DE ESTADO EM APLICAÇÕES WEB

# Manutenção de estado em aplicações Web

- **O protocolo HTTP é *stateless***
  - Os pedidos são independentes e sem relação (mesmo pedidos consecutivos do mesmo cliente na mesma ligação TCP)
- Como manter uma **conversação** entre o cliente e o servidor em HTTP?
  - Utilizam-se HTTP Cookies



RFC 2965: *HTTP State Management Mechanism*  
RFC 2964: *Use of HTTP State Management*



- O que são Cookies?
  - Mecanismo que fornece a aplicações HTTP servidoras suporte para guardar e obter informações sobre o cliente
  - Manutenção de informação de estado sobre o cliente
- Que informação contém um Cookie?
  - Informação sobre o estado do cliente na forma de par `nome=valor`
  - ‘Range’ de URLs para o qual o estado é válido
  - Data de validade (para o caso de Cookies persistentes)
- Como são suportados em HTTP ?
  - **Utilizando headers HTTP**

# Headers HTTP referentes aos Cookies

---

Sintaxe do header 'Cookie' no pedido HTTP

**Cookie:** {<NAME>=<VALUE>;}+

Sintaxe do header 'Set-Cookie' na resposta HTTP

**Set-Cookie:** {<NAME>=<VALUE>;}+ [expires=<DATE>;]  
[path=<PATH>;] [domain=<DOMAIN\_NAME>;] [secure]

**Nota:** <DATE> -> Wdy, DD-Mon-YYYY HH:MM:SS GMT

*Exemplo:* Thursday 02-Feb-2006 00:00:00 GMT

# Algumas notas sobre Cookies (1)

---

- Uma resposta HTTP pode conter múltiplos cabeçalhos Set-Cookie
- Instâncias com a mesma path e nome, são rescritas tendo precedência a última delas
  - Instâncias com a mesma path mas com nomes diferentes, são adicionadas ao conjunto
- Definir um Cookie com uma path mais genérica, não remove Cookies com paths mais específicas
  - São enviados todos os mapeamentos para o Cookie, mesmo com paths diferentes
  - Quando enviados para o servidor, os Cookies com path mais específica deverão ser enviados primeiro
- O campo expires indica ao cliente quando o Cookie "deve" ser removido
  - No entanto o cliente não é obrigado a removê-lo!
  - O cliente pode remover o Cookie antes deste expirar se o número de Cookies exceder os limites internos do cliente

## Algumas notas sobre Cookies (2)

---

- Limites máximos para o número de Cookies que um cliente pode guardar de cada vez:
  - 4 Kb por Cookie
  - 300 Cookies no total
  - 20 Cookies por domínio
- Para uma aplicação servidora **apagar um Cookie** no cliente, deverá enviar na resposta um Cookie com o mesmo nome e uma data de expiração passada
  - Os proxys HTTP não deverão fazer cache dos cabeçalhos Set-Cookie
  - Se um proxy receber um cabeçalho Set-Cookie deverá propagá-lo para o cliente independentemente do código de resposta ser 304 (Não modificado) ou 200 (Ok)
  - Da mesma forma, se um pedido HTTP tiver cabeçalhos Cookie, estes deverão ser propagados pelo *proxy*, independentemente de ser um pedido condicional (If-modified-since) ou não.

# Utilizações normais de Cookies

---

- **Podem ser** utilizados para:
  - Criar sessões (conversação)
  - Deixar registo de navegação
  - Deixar registo de preferências do cliente
  - Evitar *login* (*login* automático)
- **Não podem** (não é possível) ser utilizados para:
  - Aceder ao disco rígido
  - Enviar vírus para o cliente
  - Enviar informação para o servidor sem conhecimento do cliente



# Exemplos de transacções HTTP com Cookies

---

## (1) Cliente envia um pedido e recebe a resposta:

- Set-Cookie: CUSTOMER=WILE\_E\_COYOTE; path=/; expires=Wednesday, 09-May-1999 23:00:00 GMT

## (2) Cliente envia um pedido na path '/':

- Cookie: CUSTOMER=WILE\_E\_COYOTE

## (3) Servidor envia resposta

- Set-Cookie: PART\_NUMBER=ROCKET\_LAUNCHER\_0001; path=/'

## (4) Cliente envia um pedido na path '/':

- Cookie: CUSTOMER=WILE\_E\_COYOTE; PART\_NUMBER=ROCKET\_LAUNCHER\_0001

## (5) Servidor envia resposta:

- Set-Cookie: SHIPPING=EDEX; path=/foo

## (6) Cliente envia um pedido na path '/':

- Cookie: CUSTOMER=WILE\_E\_COYOTE; PART\_NUMBER=ROCKET\_LAUNCHER\_0001

## (7) Cliente envia um pedido na path '/foo':

- Cookie: SHIPPING=EDEX; CUSTOMER=WILE\_E\_COYOTE; PART\_NUMBER=ROCKET\_LAUNCHER\_0001

# Bibliografia

---

- HTTP @ W3C
  - <http://www.w3.org/Protocols>
- Hypertext Transfer Protocol -- HTTP/1.1 ([RFC 2616](#))
  - <http://www.w3.org/Protocols/rfc2616/rfc2616.html>
- HTTP State Management Mechanism ([RFC 2965](#))
  - <ftp://ftp.rfc-editor.org/in-notes/rfc2965.txt>
- Use of HTTP State Management ([RFC 2964](#))
  - <ftp://ftp.rfc-editor.org/in-notes/rfc2964.txt>
- Aplicações de análise de tráfego
  - Fiddler: <http://www.fiddlertool.com/fiddler/> (funciona como proxy)
  - HttpWatch: <http://www.httpwatch.com/> (para o IE e Firefox)
- XMLHttpRequest
  - <http://www.w3.org/TR/XMLHttpRequest/>