

Instituto Superior de Engenharia de Lisboa

Departamento de Engenharia da Electrónica das
Telecomunicações e dos Computadores

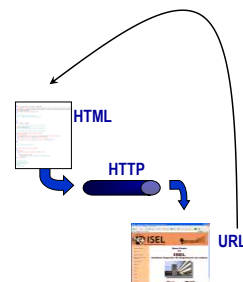
Infraestruturas Computacionais Distribuídas WEB

Versão 1.2



Índice

- Introdução – Internet/Intranet/Extranet
- Modelo Cliente – Servidor
- Serviços
- Serviço WEB
 - Esquema de nomes URI – URL
 - Protocolo HTTP
 - Linguagem HTML



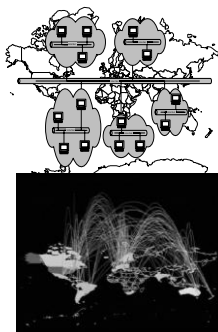
ISEL-DEI/TC

2



Internet - Introdução

- A Internet é uma rede global de comunicações que interliga milhares de Intranets. (token-ring, FDDI, Ethernet, etc.)
- Teve origem no exército dos E.U.A. com a criação da ARPANET em 1969
 - Motivação:
 - Comunicações táticas na eventualidade dos telefones ficarem inutilizados



ISEL-DEI/TC

3



Intranet/Extranet

- Intranet – Rede interna utilizada pelas organizações para partilharem ficheiros e documentos web (websites).
 - Tecnologias de rede utilizadas:
 - FDDI
 - Token-Ring
 - ATM
 - Ethernet
 - Etc...
 - Para uma maior segurança, normalmente não são acessíveis através da Internet.
 - Ex: A rede do ISEL é interna e não pode ser acedida através da Internet (NAT fornecido pelo proxy). Apenas é permitido que os servidores sejam visíveis exteriormente.
- Extranet – É uma intranet externa. Ou seja, é uma ligação entre intranets através da internet que através de ligações fiáveis e comunicações protegidas permite redes de negócio para negócio unindo empresas parceiras.
 - Por usar os padrões abertos da Internet, os parceiros não precisam usar o mesmo sistema operativo, hardware ou browser.

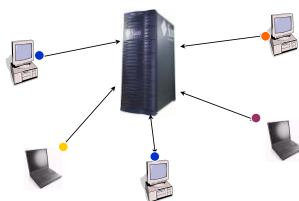
ISEL-DEI/TC

4



Arquitectura

Modelo Cliente – Servidor



- Exemplo de Serviços:
 - WEB
 - FTP – File Transfer Protocol
 - eMAIL – Electronic Mail

ISEL-DEI/TC

5



Serviços mais conhecidos

| Nome do Serviço | Porto/Protocolo | Descrição |
|-----------------|-----------------|--|
| FTP | 20,21 / TCP | Transferência de Ficheiros |
| TELNET | 23 / TCP | Shell Remota |
| SSH | 22 | Secure Shell, Transferência de ficheiros, Port forward |
| SMTP | 25 / TCP | Envio de Email |
| POP3 | 110 / TCP | Consulta de Email |
| RPC | 111 / TCP UDP | Remote Procedure Call |
| HTTP | 80 / TCP | Servidor WEB |
| HTTPS | 443 / TCP | Servidor WEB seguro |
| NFS | 2049 / TCP UDP | Partilha de ficheiros em rede |
| IRC | 6667 / TCP | Internet Relay Chat |
| ... | ... | ... |

ISEL-DEI/TC

6



Serviço WEB – Introdução

- A web é uma aplicação distribuída de grande escala
- Teve origem no CERN no final da década de 80, devido à necessidade dos investigadores partilharem os resultados das suas experiências
- A World Wide Web nasceu quando Tim Berners-Lee, um investigador do CERN, desenvolveu a linguagem HTML (*HyperText Markup Language*).
- Permite um acesso simples, através de um browser (cliente), a documentos que se encontram em computadores remotos (servidores) ligados à Internet
- Facilita a visualização transparente dos documentos **sem a preocupação da heterogeneidade** das máquinas que os armazenam

ISEL/DEI/TC

7



Serviço WEB – Introdução (cont)

- A *World Wide Web (Web)* é uma rede de recursos de informação. A Web apoia-se em três mecanismos para tornar estes recursos sempre disponíveis ao maior número possível de pessoas:
 - Um esquema de nomes uniforme para localizar os recursos na Web (ex. URIs, URLs)
 - O protocolo para aceder aos recursos através da Web (ex. HTTP - *HyperText Transfer Protocol*)
 - Hipertexto, para uma fácil navegação entre os recursos (ex. HTML - *HyperText Markup Language*)

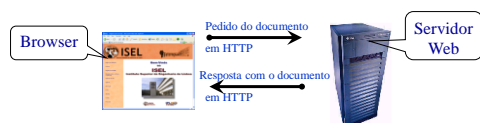
ISEL/DEI/TC

8



Arquitectura WEB

- Baseado no modelo Cliente – Servidor
- A comunicação ao nível de transporte é feita por TCP
- A comunicação obedece ao protocolo HTTP (*HyperText Transfer Protocol*)
 - Pedido/envio de informação ao/para servidor



ISEL/DEI/TC

9



Interacção Cliente – Servidor

- **Interacção Cliente – Servidor:**
 - O browser interpreta o URL e obtém do DNS o respectivo endereço IP
 - O browser estabelece uma ligação TCP com o servidor Web
 - O browser através do protocolo HTTP envia um pedido do documento em causa
 - O servidor responde enviando o documento pedido e termina a ligação
 - O browser apresenta o documento pedido
- **Interacção Cliente – Servidor:**
 - O servidor Web, por omissão, espera ligações TCP no porto 80
 - Estabelecimento da ligação
 - Servidor recebe o pedido do Cliente
 - Após encontrar o recurso pedido envia-o para o cliente
 - Termina a ligação

ISEL/DEI/TC

10



Características dos Servidores Web

- Por omissão espera pedidos TCP no porto 80
 - Tem de ser especificado no URL caso contrário
 - Ex: `http://localhost:8080`
- O servidor Web não guarda informação de estado (mensagens idempotentes)
 - Simplifica a sua implementação e reduz os recursos necessários para correr
 - Se houver um problema no servidor, este pode simplesmente reiniciar-se sem precisar de mecanismos elaborados de recuperação
 - Se a comunicação com os clientes deixar de ser possível, o servidor não é afectado
 - Têm de ser utilizados outros mecanismos para manter o estado, dificultando o desenvolvimento das aplicações (ex: comércio electrónico)

ISEL/DEI/TC

11



Identificadores de Recursos

URI (Uniform Resource Identifiers)

ISEL/DEI/TC

12



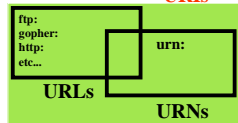
URI

URI - Uniform Resource Identifiers (RFC 2396)

- Definem a sintaxe para identificadores/localizadores de recursos.

Formato genérico de um URI

- <esquema>:<parte-especifica-esquema>



Tipos de URIs:

- URL - Uniform Resource Locators (RFCs 2368, 1738, 1808)

- Identificam por localização

- URN - Uniform Resource Names (RFC 2141)

- Identificadores persistentes, independentes da localização.

ISL-DEETC

13



URL – Uniform Resource Locator

URL



- Definido pelo RFC1738 disponível em: <http://www.ietf.org/rfc/rfc1738.txt>

- Caracteres utilizados nos URLs:

- Caracteres US-ASCII

- Caracteres que podem ser reservados:

- ["; , / ? " ' : @ = " & " #]

- Se aparecerem fora do contexto para o qual foram reservados, devem ser codificados.

- Codificação de caracteres: %código em hexadecimal>

- Se o esquema tiver links relativos e uma estrutura hierárquica, os componentes da hierarquia devem ser separados por "/"

ISL-DEETC

14



URL – Uniform Resource Locator (cont)

- No caso do esquema **HTTP** o URL tem o seguinte formato:

- http_URL = "http://" <host>[":" <port>] [<path>] [arguments]

- host – Nome ou endereço IP de um computador
- port – número inteiro identificador do porto (80/TCP se omitido)
- path – caminho identificador do documento
- arguments = "?" <arg>["&" <arg>]
- arg = <argName> "=" <argValue>

Ex:

http://www.deetc.isl.pt/

http://phoenix/~dremedios/cgi-bin /hello.cgi?Nome=Diogo+Rem%E9dios

ISL-DEETC

15



URL – Uniform Resource Locator (cont)

- Para o esquema do **FTP** temos o seguinte formato:

- ftp_URL = "ftp://" <user>[":" <password>"]@["<host>["<port>"]/<url-path>]

- user – UserName do um utilizador
- password – password do utilizador referido
- host – Nome ou endereço IP de um computador
- port – porto a efectuar a ligação (21/TCP se omitido)
- url-path – [<cwd1>/<cwd2>.../<cwdN>][<name>][<type>=<typecode>]
- cwdx – argumento a fornecer ao comando *cwd* (change working directory)
- name – nome do ficheiro
- type – código com, por exemplo, o tipo de transferência a efectuar texto/binário

Ex: ftp://dremedios@phoenix/public_html

- No caso do **MAILTO** o URL tem o seguinte formato:

- mailtoURL = "mailto:" [<rfc-822-addr-spec>] [arguments]

- arguments = "?" <arg>["&" <arg>]

- arg = <argName> "=" <argValue>

Ex: <mailto:dremedios@deetc.isl.pt>

Ex: <mailto:dremedios@deetc.isl.pt?subject=teste&body=corpo%20da%20mensagem>

- Para o esquema do **emule**:

Ex: <ed2k://file|ydsrink314setup.zip|8147716CFCA7DC2E0D1706FE34B22CF44C4DB1>

ISL-DEETC

16



Protocolo HTTP

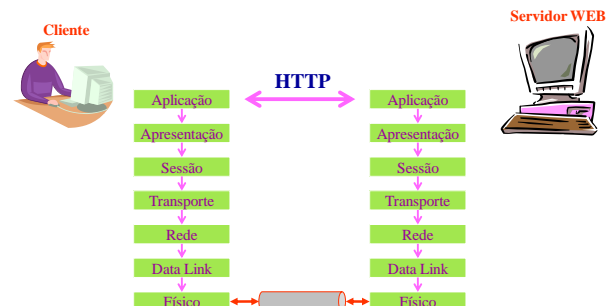
(HyperText Transfer Protocol)

ISL-DEETC

17



Protocolo – HTTP (HyperText Transfer Protocol)



ISL-DEETC

18



Protocolo – HTTP (HyperText Transfer Protocol)

- Protocolo ao nível aplicacional para arquitecturas cliente/servidor
- Protocolo *stateless*
- É o protocolo usado na Web
- Pensado para permitir o acesso a documentação em formato electrónico.

ISEL/DEI/TC

19



Funcionamento do Protocolo HTTP

- HTTP/1.1: RFC 2616 (<http://www.w3.org/Protocols/>)
- Princípio de funcionamento pedido/resposta:
 - O cliente estabelece uma ligação com o servidor;
 - O cliente envia ao servidor uma mensagem com o seu pedido;
 - O servidor responde com uma mensagem com o resultado do serviço.

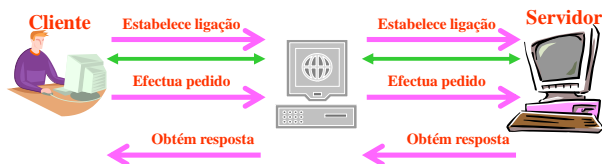


ISEL/DEI/TC

20



Funcionamento do Protocolo HTTP



ISEL/DEI/TC

21



Funcionamento do Protocolo HTTP

- Tipos de intermediários:

- Proxy*;
- Gateway*;
- Tunnel*.



ISEL/DEI/TC

22



Funcionamento do Protocolo HTTP

- Proxy*:

Pode ser visto como um agente despachante, que recebe os pedidos para um dado URI e reescreve parte ou totalidade da mensagem e reencaminha o pedido para o servidor identificado no URI.



ISEL/DEI/TC

23



Funcionamento do Protocolo HTTP

- Gateway*:

É um agente intermediário que permite transferir informação entre servidores que “residem” em camadas diferentes. Se necessário convertem os pedidos para o protocolo do servidor destino.



ISEL/DEI/TC

24



Funcionamento do Protocolo HTTP

Tunnel:

Funciona como um ponto de ligação entre dois interlocutores sem modificar o conteúdo das mensagens. É muito utilizado quando o cliente e o servidor estão separados por um *firewall*.



ISEL/DEI/TC

25



Funcionamento do Protocolo HTTP

O pedido do cliente para o servidor e a respectiva resposta é feita em ASCII

As mensagens podem obedecer ao formato MIME (Multipurpose Internet Mail Extensions: **RFC 2045**), bastando indicar qual a sua versão

Têm três componentes:

Comando (ex: **GET**, **HEAD**, **POST**, PUT, DELETE, TRACE, CONNECT)

Cabeçalhos:

Cliente → Servidor (indicar a versão do browser, os formatos de documentos que reconhece, etc...)

Servidor → Cliente (indicar a versão do servidor, dimensão dos documentos, data da última modificação, etc...)

Corpo da mensagem



ISEL/DEI/TC

26



Conteúdo das mensagens HTTP

Pedido enviado do cliente para o servidor:

A primeira linha é constituída por:

- Identificação do pedido, denominado comando ou método
- A identificação do documento em causa (URI)
- A versão do protocolo HTTP

As linhas seguintes dependem do método indicado na primeira linha e contêm os cabeçalhos de acordo com o RFC 822

A última linha está em branco (CRLF) indicando o fim do pedido

Resposta enviada do servidor para o cliente:

A primeira linha indica o estado do pedido efectuado

Se foi servido com sucesso ou, caso contrário, o código de erro indicando o problema

As seguintes linhas contêm cabeçalhos

De seguida virá a informação correspondente ao pedido efectuado (ex: documento)

ISEL/DEI/TC

27



HTTP – augmented BNF

generic-message = <start-line> *(<message-header> CRLF) CRLF [< message-body>]

start-line = <Request-Line> | <Status-Line>

Request-Line = <Method> SP <Request-URI> SP <HTTP-Version> CRLF

Method = "OPTIONS" | "GET" | "HEAD" | "POST" | "PUT" | "DELETE" | "TRACE" | "CONNECT" | <extension-method>

extension-method = token

Request-URI = "*" | <absoluteURI> | <abs_path>

No caso da versão do HTTP ser o 1.1 tem de se fornecer a informação do host. Ou pelo URI absoluto ou pelo cabeçalho "host: <hostname>"

Message-header = header-name ":" header-value CRLF

Response = <Status-Line> *(general-header | response-header | entity-header) CRLF [message-body]

Status-Line = <HTTP-Version> SP <Status-Code> SP <Reason-Phrase> CRLF

ISEL/DEI/TC

28



Códigos de resposta do servidor

1XX - Informativo
2XX - códigos de sucesso
3XX - recolocação/redireccionamento
4XX - Erro no cliente
5XX - Erro no servidor

Códigos de resposta

Exemplos

200 OK
201 Criado
204 Sem conteúdo
304 Não modificado
400 Pedido mal formado
401 Não tem autorização
404 Não encontrado
501 Comando não implementado
503 Serviço não disponível

ISEL/DEI/TC

29



Cabeçalhos HTTP (RFC 822)

Permite a troca de informação entre o cliente e o servidor

Quatro categorias:

Geral – Informação que não está directamente relacionada com o cliente, o servidor e o HTTP

Pedido – Permite indicar preferências nos formatos recebidos e parâmetros do servidor

Resposta – Informação do servidor que responde

Dados – Informações referentes aos dados transferidos

Cabeçalhos gerais:

Cache-control – Indica como efectuar a cache das páginas no cliente (sim, não e tempo de validade)

Upgrade – Indica a versão do protocolo HTTP a usar

MIME-version – Indica a versão do MIME a utilizar na comunicação

Date – Formato preferencial para a data

ISEL/DEI/TC

30



Cabeçalhos HTTP (cont)

- Cabeçalhos de pedido (cliente → servidor):**
 - Accept** – Indica os tipos de dados que sabe interpretar (ex: */*, image/jpeg, text/html, application/msword, etc...)
 - Cookie** – Permite associar um estado ao cliente. Par “nome” = “valor”
 - If-modified-since** – indicando uma data, permite informar que só queremos receber o documento caso tenha sido alterado desde então
- Cabeçalhos de resposta (servidor → cliente):**
 - Set-cookie** – Fornece um par “nome”=“valor” que fica associado ao cliente permitindo a manutenção de estado
 - Server** – Informa qual o servidor que está a servir o pedido e qual a sua versão
- Cabeçalhos de dados:**
 - Content-length** – Dimensão em bytes do corpo de dados
 - Content-type** – Tipo de dados contidos no corpo de dados (ex: image/jpeg, text/html, etc...)
 - Expires** – Indica o tempo até quando o documento provavelmente não será alterado e poderá ficar em cache
 - Last-modified** – Data e hora da última modificação ao ficheiro
 - Location** – Indica que o documento está noutra URL

ISEL/DEETC

31



Métodos GET e HEAD

- Método GET**
 - Pede um documento ao servidor dado um URL (ex: ficheiro, CGI, etc...)
 - A resposta do servidor contém:
 - Na primeira linha, o estado
 - Nas seguintes os cabeçalhos
 - E em seguida o documento pedido
- Método HEAD**
 - Semelhante ao GET, mas a resposta não inclui o documento
 - Utilizado para saber informações sobre o recurso pedido
 - A resposta do servidor contém:
 - Na primeira linha, o estado
 - Nas seguintes os cabeçalhos

•Util para saber se a versão que o browser tem em cache já foi alterada
•Calcular o tempo de download de determinado documento

ISEL/DEETC

32



Exemplo – GET e HEAD (telnet)

Pedido de GET:

```
Stelnet phoenix 80
C: Trying 10.64.11.121...
C: Connected to phoenix.
C: Escape character is '^J'.
C:
S: GET / HTTP/1.1
C: Host: phoenix
C:
S: HTTP/1.1 200 OK
S: Date: Mon, 26 Apr 2004 08:20:24 GMT
S: Server: Apache/2.0.40 (Red Hat Linux)
S: Last-Modified: Fri, 23 Apr 2004 16:47:08 GMT
S: ETag: "15b738-1740-c73a3b00"
S: Accept-Ranges: bytes
S: Content-Length: 5952
S: Content-Type: text/html; charset=ISO-8859-1
S:
S: <html>
S: <head>
S: <meta http-equiv="Content-Language" content="pt">
S: <meta name="GENERATOR" content="Microsoft
FrontPage 5.0">
S: <meta name="ProgId"
content="FrontPage.Editor.Document">
S:
S: <title>Phoenix [SES Server]</title>
S: </head>
S:
S: <body link="#0000FF" vlink="#800080">
...
S: </body>
S: </html>
S:
C: Connection closed by foreign host.
```

ISEL/DEETC

33

Pedido de HEAD:

```
Stelnet phoenix 80
C: Trying 10.64.11.121...
C: Connected to phoenix.
C: Escape character is '^J'.
C:
S: HEAD / HTTP/1.1
C: Host: phoenix
C:
S: HTTP/1.1 200 OK
S: Date: Mon, 26 Apr 2004 08:21:07 GMT
S: Server: Apache/2.0.40 (Red Hat Linux)
S: Last-Modified: Fri, 23 Apr 2004 16:47:08 GMT
S: ETag: "15b738-1740-c73a3b00"
S: Accept-Ranges: bytes
S: Content-Length: 5952
S: Content-Type: text/html; charset=ISO-8859-1
C: Connection closed by foreign host.
S:
```



Exemplo GET num browser

Aceder à página:

<http://phoenix.deetc.isel.ipl.pt/~dremedios/index.html>

[http://host\[:porto\]/caminho](http://host[:porto]/caminho)

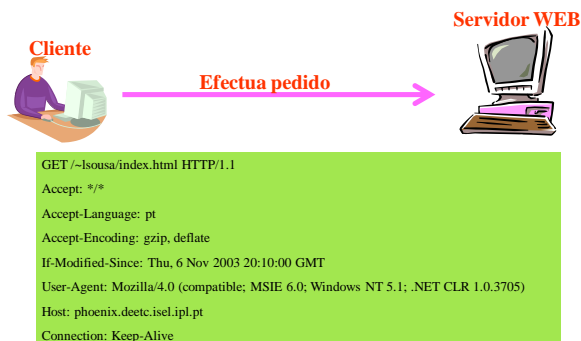


ISEL/DEETC

34



Exemplo

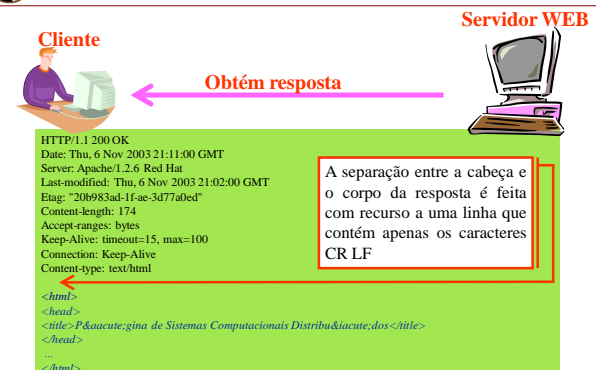


ISEL/DEETC

35



Exemplo



ISEL/DEETC

36



Método POST

Método POST

- Envia dados do cliente para o servidor (no corpo da mensagem)
- Os dados são enviados no servidor a um programa auxiliar, para serem processados (ex: CGI)
- O programa a processar os dados é especificado no URL

```
C: POST /cgi-bin/post-query HTTP/1.1
C: Host: phoenix
C: Accept: */*
C: User-Agent: Lynx/2.2 libwww/2.14
C: From: grobe@www.cc.ukans.edu
C: Content-type: application/x-www-form-urlencoded
C: Content-length: 150
C:
C: org=Academic%20Computing%20Services
&users=10000
C: &browsers=lynx
C: &browsers=cello
C: &browsers=mosaic
C: &others=MacMosaic%2C%20WinMosaic
```

```
Resposta: (incompleta)
S:
S: Content-Type: text/html
S:
S: <h1>Query Results</h1>
S: You submitted the following name/value pairs:
S: <ul>
S: <li>org = Academic Computing Services </li>
S: <li>users = 10000 </li>
S: <li>browsers = cello </li>
S: <li>browsers = lynx </li>
S: <li>browsers = xmosaic </li>
S: <li>others = Mac Mosaic, Win Mosaic </li>
S: </ul>
```

Resposta gerada dinamicamente

ISEL/DEI/TC

37



Métodos PUT, DELETE, TRACE e CONNECT

Método PUT

- Permite colocar no servidor Web, um determinado documento (simétrico ao GET)
- O URL indica o local onde colocar o documento

Método DELETE

- Apaga o documento referenciado no URL

Método TRACE

- Permite descobrir, caso seja pedido um dado documento, se no seu trajeto passa por proxy's

Método CONNECT

- Permite fazer *tunneling* nos proxy's que o suportem

Os métodos DELETE, TRACE e CONNECT nem sempre são suportados

ISEL/DEI/TC

38



Métodos HTTP

| Método | HTTP 1.0 | HTTP 1.1 |
|--------|----------|----------|
| GET | ✓ | ✓ |
| HEAD | ✓ | ✓ |
| POST | ✓ | ✓ |
| PUT | X | ✓ |
| DELETE | X | ✓ |
| TRACE | X | ✓ |

ISEL/DEI/TC

39



Fundamentos do HTML (HyperText Markup Language)

ISEL/DEI/TC

40



HTML (HyperText Markup Language)

- É uma *Markup Language* e teve origem na linguagem *SGML* (*Standard Generalized Markup Language*)
- É uma linguagem que permite descrever a estrutura e a forma como os dados devem ser formatados de forma independente da plataforma
- É baseada em etiquetas (tags) HTML que os browsers interpretam para representar os documentos
- Actualmente o standard é o XHTML 1.0 que é uma evolução da versão HTML 4.01 publicado em Dezembro de 1999.
- Links sobre HTML :
 - <http://www.w3.org/MarkUp/>
 - A especificação está disponível em:
 - <http://www.w3.org/TR/REC-html40/>
 - <http://www.w3schools.com/html/>
 - Lista de tags HTML: http://www.w3schools.com/html/html_reference.asp
 - <http://www.htmlgoodies.com/>

ISEL/DEI/TC

41



HTML (cont)

Cada etiqueta ou elemento HTML é composto por:

- Uma etiqueta de início e outra de fim

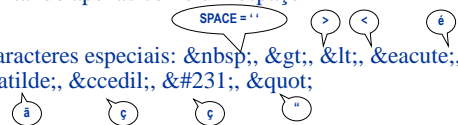
<Elemento> ... </Elemento>

- Pode conter vários atributos

<ELEMENTO atributo1="valor1" atributo2="valor2"> ... </ELEMENTO>

Os browsers truncam espaços e linhas em branco, contando apenas como um espaço

- Caracteres especiais: , > , < , ´ , ã , ç , ç , "



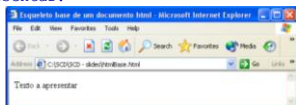
ISEL/DEI/TC

42



HTML – Estrutura

- A estrutura base de um documento HTML é a seguinte:
- `<html> ... </html>` Etiquetas indicativas do início e do fim do documento HTML
 - `<head> ... </head>` Delimitam a seção de cabeçalhos do documento
 - `<title> ... </title>` Cabeçalho que permite especificar o título da página (visualizado na barra superior do browser)
 - `<body> ... </body>` Corpo do documento, onde será colocado o conteúdo a ser formatado

[illegible]

HTML – Estrutura

- ## Atributos da etiqueta *body*
- *background* – especifica a imagem de fundo
 - Ex: `<body background="fundo.gif">`
 - *bgcolor* – especifica a cor de fundo
 - Ex: `<body bgcolor="green">`
 - *bgproperties* – configura o fundo da página, como fixo ou deslizante
 - Ex: `<body bgproperties="fixed">`



HTML – Etiquetas

| Etiqueta | Descrição |
|-----------------------------|--|
| <html>, </html> | Delimita o início e o fim do documento |
| <head>, </head> | Delimita a zona de cabeçalhos |
| <title>...</title> | Delimita o título |
| <body>, </body> | Delimita o corpo do documento |
| <h>, </h> | Delimita uma secção de nível n 0-6 |
| , | Delimita uma lista não numerada |
| , | Delimita uma lista ordenada |
| , | Delimita um item da lista |
| | Mudança de linha |
| <hr /> | Inserir uma linha horizontal |
| <p>, </p> | Delimita um parágrafo |
| | Inserir uma imagem indicada no atributo src |
| ... | Delimita um hyperlink para a página indicada em href |
| <div>...</div> | Define um grupo de elementos (Divisão) |
| ... | Define um grupo de elementos (inline) |
| <!--...--> | Delimita uma zona de comentários |



HTML – Etiquetas (cont)

| Etiqueta | Descrição |
|----------------------|--|
| <pre>...</pre> | Delimita texto cujo formato será respeitado mantendo espaços e mudanças de linha |
| ... | Delimita texto enfatizado |
| ... | Delimita texto carregado |
| <code>...</code> | Delimita código fonte |
| <i>...</i> | Delimita texto em itálico |
| ... | Delimita texto em negrito |
| <object>...</object> | Define um objecto embelhado (ex: animação flash) |

Est
lóg

- ## Applets
- ▀ <applet> Permite carregar e correr uma applet (pedaço de código java). Alguns dos seus atributos são:
- Ⓢ Code – Indica o código que vai carregar (bytecodes de java)
 - Ⓢ width e height – especifica a área que a applet vai ocupar dentro da página
- Ex: <applet code=“myapplet.class” width=“250” height=“150” > </applet>



HTML – Etiquetas (cont)

- Tabelas
- <table> Delimita uma tabela. Pode ter os seguintes atributos:
 - < border - Indica a espessura da margem exterior da tabela
 - < cellpadding - Espaçamento à volta de cada célula
 - < width - Largura que a tabela vai ter. Pode ter uma percentagem relativa ao tamanho da janela
 - <thead> Delimita a zona do cabeçalho dentro da tabela
 - <tbody> Delimita o corpo da tabela
 - <tfoot> Delimita o fim da tabela
 - <tr> Delimita uma linha da tabela
 - <th> Delimita uma coluna que será cabeçalho da tabela
 - <td> Delimita uma coluna dentro de uma linha (célula)
 - O atributo Colspan permite indicar que a célula vai ocupar mais que uma coluna
- Ícones
- Listas ordenadas
- type - Define o tipo de numeração
 - type="A" (A,B,...,Z)
 - type="a" (a,b,...,z)
 - type="I" (I, II, III, IV, ...)
 - type="i" (i, ii, iii, iv, ...)
 - type="1" (1, 2, 3, ...) (Usado por omissão)
 - start - Indica o início da numeração
 - start="4" (4, 5, 6, ...)
- Listas não numeradas
- type - Define o símbolo
 - type="disc" (●) (Usado por omissão)
 - type="circle" (○)
 - type="square" (■)



HTML – *Cascading Style Sheets* (Noções)

- Descrição de regras de estilo que indicam ao browser como apresentar os diversos elementos de um documento HTML
- Vêm substituir os atributos de formatação dos elementos HTML.
 - Ex: `<hr width="75%" size="5" align="center">`
- Forma de separar a apresentação do documento da sua estrutura.
- A maioria das propriedades das *style sheets* podem ser utilizadas em qualquer elemento HTML



HTML – Cascading Style Sheets (Noções)

Quando um browser lê uma *style sheet*, vai formatar o documento de acordo com ela. Existem três formas de inserir *style sheet*'s:

- ☐ **Style Sheet Externa** – Ideal para quando o estilo é para ser aplicado a muitas páginas. Torna-se possível alterar o aspecto de todo um *web site* alterando apenas um ficheiro. Cada documento tem de se ligar à *style sheet* que está a usar, utilizando a etiqueta `<link>`. Esta etiqueta é colocada na secção dos cabeçalhos.
...<head> <link rel="stylesheet" type="text/css" href="meuEstilo.css"> </head>...
- ☐ **Style Sheet Interna** – Uma *style sheet* interna deve ser usada quando um único documento tem um estilo único. São definidos na secção dos cabeçalhos utilizando a etiqueta `<style>`.
...<head> <style type="text/css"> body {background-color: red} p {margin-left: 20px} </style> </head> ...
- ☐ **Inline Styles** – Deve ser utilizado quando um estilo é aplicado a uma única ocorrência de um elemento. Para utilizar *inline styles* é utilizado o atributo *style* da etiqueta do elemento respectivo. O atributo *style* pode conter qualquer propriedade CSS. O exemplo mostra como alterar a cor e a margem esquerda de um parágrafo:
...<p style="color: red; margin-left: 20px"> This is a paragraph </p>...

ISL-DE/TC

49



Exemplo de uma página em html

ISL-DE/TC

50



Exemplo de uma página em HTML

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html">
<title>Página de Sistemas Computacionais Distribuídos</title>
</head>
<body>
<h1>Página de SCD (Heading 1)</h1>
<p>Texto normal sem formatação</p>
<p style="text-align: center">Texto centrado</p>
<p><em>Texto</em> <strong>com</strong> <code>e</code> <del>tipos</del> </p>
<b><u>formatação</u></b>
<p><a href="http://www.isl.ipt.pt" title="Entrada do ISL">Link para a página inicial do ISL</a>
<h2>Lista não numerada (Heading 2)</h2>
<ul>
<li>Item 1</li>
<li>Item 2</li>
<li>Item 3</li>
</ul>
<h2>Lista numerada (Heading 2)</h2>
<ol>
<li>Item numerado 1</li>
<li>Item numerado 2</li>
<li>Item numerado 3</li>
</ol>
</body>
```

Acrescenta ao HTTP o cabeçalho: "Content-Type: text/html"

Heading 1

Título da página

Formatações várias

Link para outra página

Lista de itens não numerados

Lista de itens numerados

continua

ISL-DE/TC

51



Exemplo de uma página em HTML (cont)

```
<p>Exemplo de uma pequena tabela 2x2:</p>
<table border="1">
<tr>
<th>Heading</th>
<th>Another Heading</th>
</tr>
<tr>
<td>Posição</td>
<td>Posição</td>
</tr>
<tr>
<td>Posição</td>
<td>Posição</td>
</tr>
</table>
<p>Exemplo de utilização de imagens:</p>
<p style="text-align: center">



</p>
<p style="text-align: center"><b>FIM</b></p>
```

Cabeçalhos da tabela

Primeira linha

Segunda linha

Imagem

ISL-DE/TC

52



HTML – Frames

- ☐ A página HTML que contém a definição das molduras não pode ter corpo (`<body>`)
- ☐ Criação de molduras dentro de uma página
 - ☐ **<frameset>** – Permite definir em quantas regiões vamos querer dividir a janela do browser e a largura de cada uma delas.
 - ☐ **Cols** – Define a lista com as larguras para as *frames* verticais (ex: `<frameset cols="25%, 25%, *">`)
 - ☐ **Rows** – Define a lista com as larguras para as *frames* horizontais (ex: `<frameset rows="100, *">`)
 - ☐ **Border** – Indica a espessura da linha exterior da moldura ("5" usado por omissão)
 - ☐ **Bordercolor** – Configura a cor da linha exterior (cinza por omissão)
 - ☐ **<frame>** – Define a moldura
 - ☐ **Src** – Especifica o URL da página que será carregada nesta moldura
 - ☐ **Name** – Define um nome único para a região
 - ☐ **Frameborder** – Indica se existirá fronteira na moldura (ex: 0 ou 1)
 - ☐ **Marginheight**, **Marginwidth** – Permite definir em pixels a margem no topo e no fundo da moldura (ex: `noresize`)
 - ☐ **Noresize** – Indica se a moldura pode ser redimensionada pelo utilizador
 - ☐ **Scrolling** – Indica se a *frame* tem as barras de *scroll* (yes, no ou auto)

ISL-DE/TC

53



Exemplo de frames

- ☐ A página tem definidas duas molduras verticais:

```
[testeframe.html]
<html>
<head>
<frameset cols="150, *" noresize="noresize">
<frame src="mercuro.html" noresize="noresize">
<frame src="sun.html" noresize="noresize">
</frameset>
</html>

[mercuro.html]
<html>
<HEAD></HEAD>
<BODY>

</BODY>
</HTML>

[sun.html]
<html>
<HEAD></HEAD>
<BODY>

</BODY>
</HTML>
```

testeframe.html

mercuro.html

sun.html

ISL-DE/TC

54



“Bibliografia” utilizada

- RFC's
 - RFC 1945 – HTTP 1.0
 - RFC 2616 – HTTP 1.1
 - RFC 2045 - Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies
 - RFC 822 – Standard for the format of ARPA Internet text messages
 - RFC 2396 - Uniform Resource Identifiers (URI): Generic Syntax
 - RFC 1738 - Uniform Resource Locators (URL)
- HTML 4.0 Sourcebook
 - Graham, Ian S. – John Wiley & Sons; 4th edition
- Special Edition Using XHTML
 - Molly Holzschlag ; Que
- Links:
 - HTTP – <http://www.w3.org/Protocols/>
 - HTML – <http://www.w3.org/TR/html401/>
 - Style Sheets – <http://www.w3.org/TR/REC-CSS1>

