

# **TECNOLOGIAS E APLICAÇÕES WEB**



**TELEMÉDIA**

**2006/2007**

# Sumário

---

- A iniciativa World Wide Web
- Protocolos usados no WWW
- Extensões aos clientes e aos servidores
- Proxy, Caching e Mirroring
- Pesquisa no WWW
- Segurança no WWW

Texto auxiliar para consulta (acetatos não são baseados nesse texto):

**Multimédia e Tecnologias Interactivas**, *Nuno Ribeiro*, FCA, 2004

(Capítulo 9)

# Introdução

---

## *WWW – The Killer App*

As recently as 1993, it was common to read of a coming revolution in communication and computing technologies. Authors predicted a future information economy, the emergence of digital libraries and newspapers, the prospects of commerce over the network, and so forth. Yet the press was also filled with skeptical articles, suggesting that although there might well be a trend toward an information superhighway, it seemed to lack on-ramps accessible to normal computer users.

In an astonishingly short period of time, this situation has reversed itself.

Kennet P. Birnam, in “Building Secure and Reliable Network Applications”

# World Wide Web

---

## O que é?

- Nasce em 1989, CERN, Suíça...
- A designação é a do projecto que lhe deu origem...
- Concretiza o conceito de **hipermédia na Internet**

## Definição

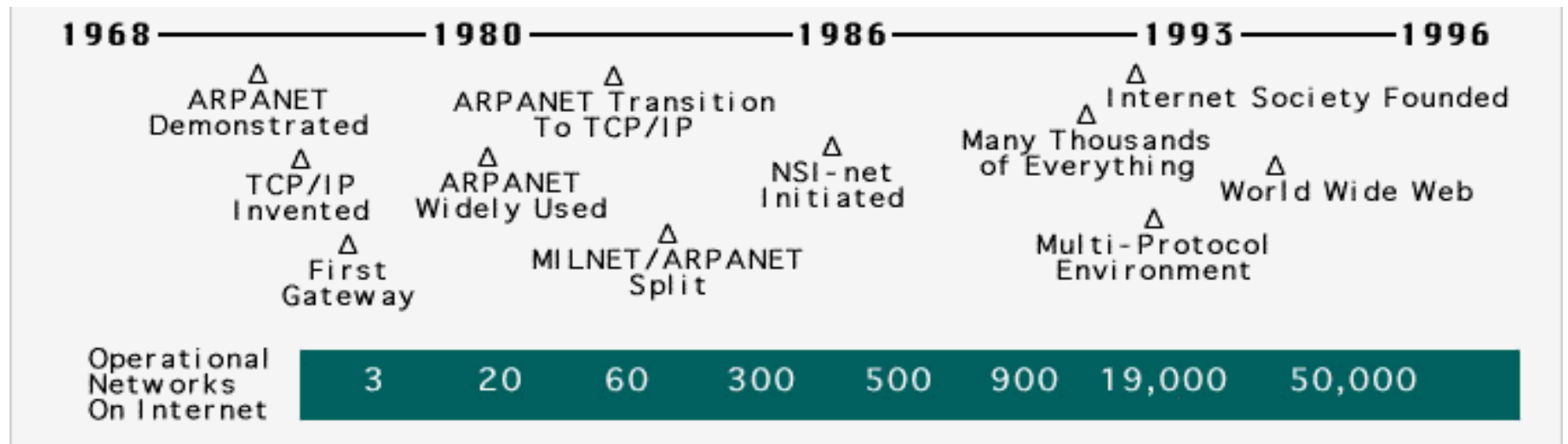


Sistema de informação *hipermédia*,  
*cooperativo, distribuído e heterógeneo*

## Outros significados:

- designa o conjunto dos *protocolos* desenvolvidos pelo projecto
- designa o *hiperespaço* de informação disponível na Internet e suportado por *servidores de informação* interligados...

# World Wide Web



# World Wide Web

---

- Ideias base do WWW:

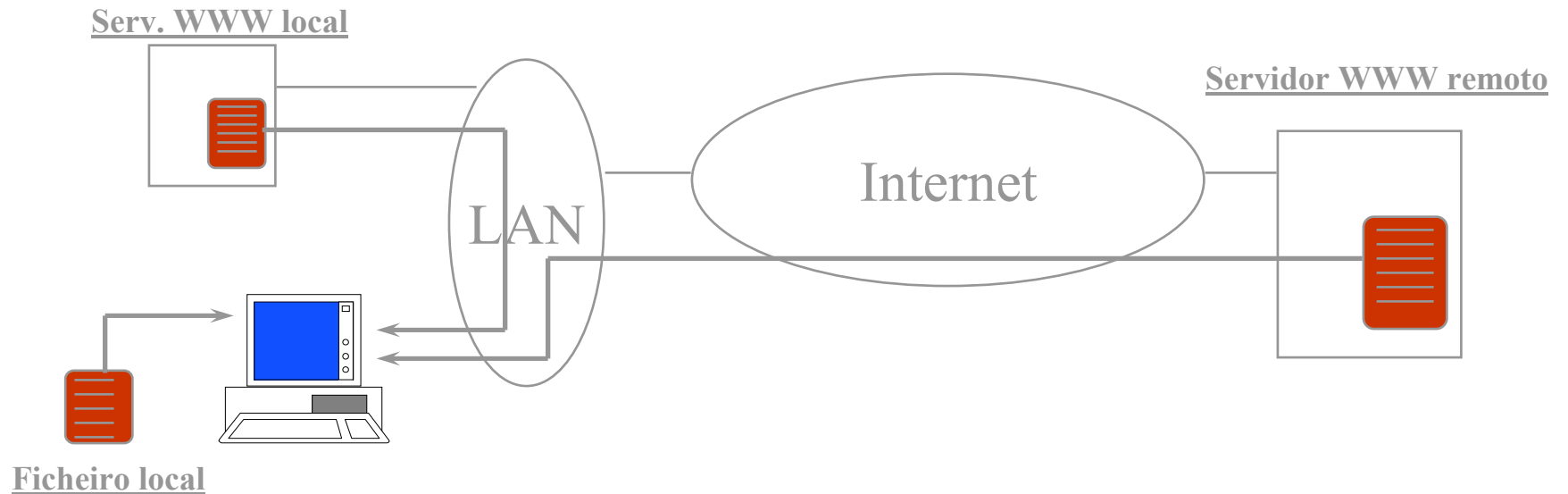
1. Todos *podem criar* documentos e inseri-los na *teia!*

- qualquer um pode ser autor, incluir referências para qualquer outro documento, e ser referenciado livremente...
- não há nenhuma autoridade administrativa, centralizante, ou restritiva de qualquer outra forma...

2. Uma forma *uniforme* de *localizar* documentos em *todo o mundo!*

- foram introduzidos métodos para definir com exactidão onde e como encontrar documentos na Internet...
- estas *moradas*, designam-se por Uniform Resource Locators (URLs)
- não são apenas *endereços*, porque incluem *métodos de acesso*

# World Wide Web



### 3. Um *interface* com o utilizador, único e uniforme!

- similaridade de operação entre clientes distintos...
- transparência da localização (não interessa onde está!)
- esconder incompatibilidades entre sistemas na representação dos dados...

# World Wide Web

---

## 4. Acesso a qualquer “*base de dados*” de informação

- é preciso garantir acesso a dados que não tenham sido produzidos especificamente para o WWW...
  - mesmo que forma mais limitada, e sem *links*
- são necessárias aplicações que sirvam de intermediários no acesso:
  - designam-se por *gateways de aplicação*

## 5. Um suporte para realizar *transacções*!

- o utilizador pode introduzir dados em formulários e enviá-los...
- ...o que permite a realização de vários tipos de transacções.
- Uma das utilizações, entre muitas, é o comércio electrónico!



# WWW - Os protocolos

---

- *Uniform Resource Locators (URLs)*

Formato geral:



- 1 protocolo de acesso aos dados (HTTP, FTP, etc..)
- 2 nome completo da máquina (ou endereço IP!)
- 3 pastas dentro das quais se encontra o documento
- 4 nome do documento e respectiva extensão!
- 5 referência a uma âncora de destino dentro do documento.

Exemplo:

☞ <http://med-amsa.bu.edu/Gutenberg/Welcome.html>

# WWW - Os protocolos

---

- *Uniform Resource Locators (URLs)*

## Outros exemplos:

- ftp://ftp.di.uminho.pt/pub/README (arquivo ftp)
- http://marco.uminho.pt/MCS/index.html (servidor www)
- telnet://orpheu.ci.uminho.pt (acesso remoto por telnet)

## Exemplos com formatos especiais:

- news:pt.internet (grupo de *news*)
- mailto:aXXXXX@correio.ci.uminho.pt (envio de e-mail)

Estes exemplos usam a **configuração E-Mail** do utilizador!

# WWW - Os protocolos

---

- Os URLs podem ser:
  - **Absolutos** – Seguem sempre o formato geral, ou seja, começam sempre pelo protocolo de acesso...
    - Ao mudar um site Web de servidor pode ser necessário corrigir as referências absolutas (nome do servidor)
  - **Relativos** – Não usam o formato geral... começam apenas na parte que referencia pastas ou ficheiros. É sempre necessário transformá-los em URLs absolutos herdando as restantes partes da página base onde são inseridos.
    - Mais fácil de mudar um site completo de máquina, quando se usam internamente URLs relativos



# Exercício sobre URLs

- Construa o URL de acesso ao documento **Exercicio.html** que está numa pasta de nome *aulas* dentro de outra pasta de nome *telemidia*. O ficheiro está disponível num servidor cujo endereço IP é 193.137.19.21 e que se chama *telemidia*. O referido servidor faz parte do domínio do departamento de informática (*di.uminho.pt*)
- Identifique todas as partes constituintes do seguinte URL  
[http://www.google.com/search?sourceid=navclient-ff&ie=UTF8&rlz=1B2GGGL\\_enPT176&q=webboom](http://www.google.com/search?sourceid=navclient-ff&ie=UTF8&rlz=1B2GGGL_enPT176&q=webboom)  
<http://marco.uminho.pt/disciplinas/TELEMEDIA/#material>
- Transforme o seguinte URL relativo num URL absoluto, supondo que iria ser usado como *ligação* no documento *Exercicio.html* acima referido:  
*../exames/teste-exemplo.html*
- Diga, justificando, se o seguinte URL é relativo ou absoluto:  
*/index.html*

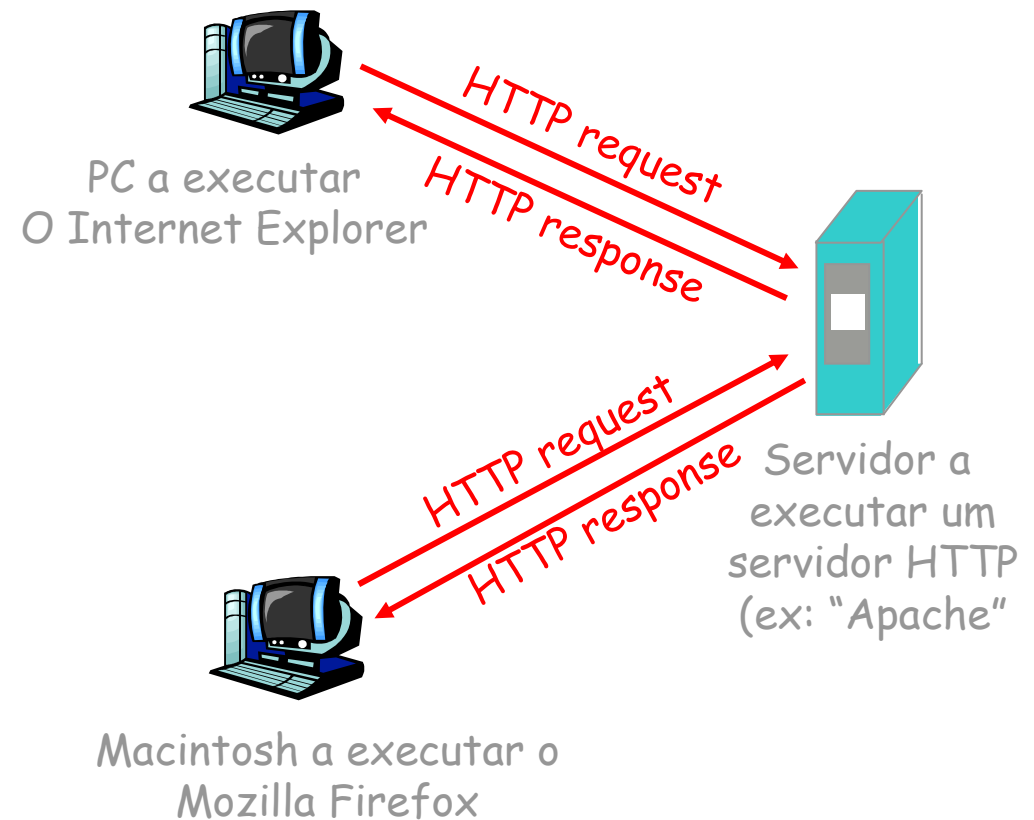
# WWW - Os protocolos

- *HyperText Transfer Protocol (HTTP)*

Segue o modelo cliente – servidor, como a maioria das aplicações Internet:

**cliente**: browser que faz pedidos, recebe “objectos” Web de volta e exhibe-os no monitor;

**servidor**: servidor HTTP que aceita pedidos e envia objectos na resposta;  
Atende numa porta 80

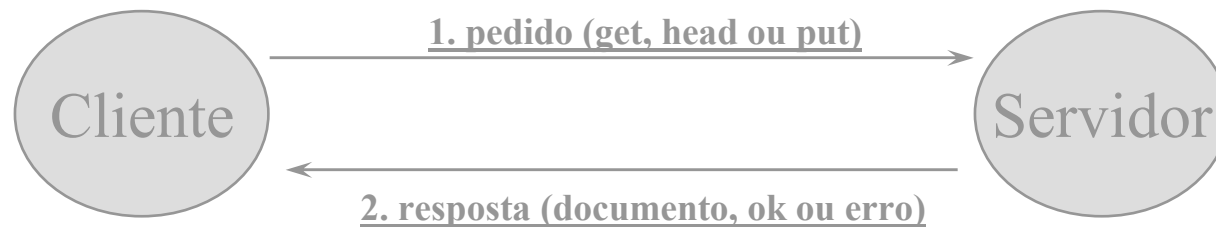


# WWW - Os protocolos

---

- *HyperText Transfer Protocol* (HTTP)

- Não é indispensável: podem ser usados outros protocolos...
- Optimizado para hipertexto interactivo
- Rápido: pedidos satisfeitos numa única interacção (ida e volta)



- Permite esquecer o “estado” - Diz-se **não orientado ao estado** (*stateless*)
  - Não se estabelece nenhuma sessão entre servidor e cliente
  - Torna os servidores muito mais simples e menos susceptíveis a erros
  - O URL é mais que suficiente para aceder ao documento!
- **Extensível**: suporta transferência de qualquer tipo de dados

# WWW - Os protocolos

---

- *HyperText Transfer Protocol* (HTTP)

Dois tipos de mensagens: **HTTP Request** e **HTTP Response**

São mensagens em texto normal, e portanto “legíveis”

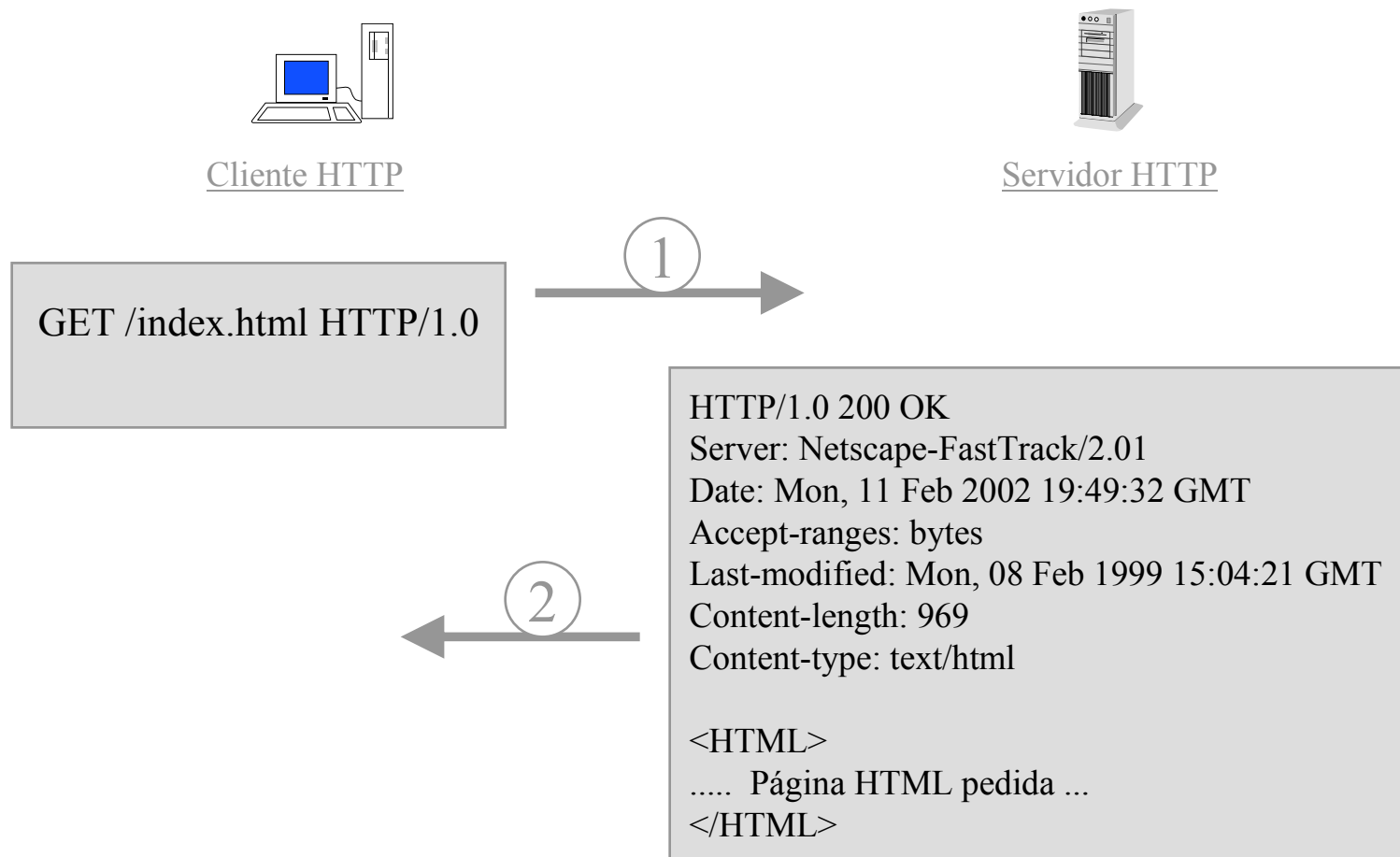
**Pedidos**: uma linha com o tipo de pedido (GET, PUT ou POST) e o URL, seguido de um cabeçalho, uma linha em branco e um corpo, normalmente vazio;

**Respostas**: uma linha com o estado (código de erro ou sucesso) seguido de um cabeçalho, uma linha em branco e o corpo com o ficheiro requerido;

Códigos: *200 OK, 404 Not Found, 301 Moved Permanently, etc.*

# WWW - Os protocolos

- *HyperText Transfer Protocol (HTTP)*



**Exemplo de obtenção de uma página**



# WWW - Os protocolos

---

- Como os browsers lidam com as conexões HTTP:
  - **HTTP não persistente** – no máximo um pedido por cada conexão a um servidor, e um único objecto transferido; habitual na versão HTTP/1.0, mas pouco comum hoje em dia; algo ineficiente;
  - **HTTP persistente** – aproveitar uma única conexão com o servidor para pedir vários objectos, sequencialmente; habitual na versão HTTP/1.1
  - **Múltiplas conexões em simultâneo** – em qualquer dos casos, o browser pode realizar um conjunto de **N** conexões em simultâneo. Uma página que inclua duas imagens, pode implicar uma primeira conexão para descarregar a página principal e de seguida duas em paralelo para descarregar as duas imagens!

# Exercício

---



- Abrir uma janela "Linha de Comandos" (Programas->Acessórios) e escrever **netstat -a** (alternativamente **netstat -a -n**)
- Visite algumas páginas Web, ao mesmo tempo que repete o comando anterior, para observar o número de conexões que vão sendo estabelecidas

# WWW - Os protocolos

---

- *HyperText Markup Language* (HTML)
  - um tipo simples derivado da linguagem *SGML*
  - suporta algumas estruturas lógicas simples...
    - cabeçalhos, parágrafos, listas numeradas e não numeradas, tabelas, *frames*, inserção de imagens, etc.
    - formulários, com elementos de selecção do tipo escolha múltipla, botões on-off, etc. para permitir transacções...
  - e também naturalmente a inserção de hiperligações :
    - âncoras e *links*
  - é o único tipo que os *browsers WWW* têm obrigatoriamente que conhecer
  - os browsers *WWW* *convertem* os códigos HTML em instruções de visualização apropriadas...

# WWW - Os protocolos

---

- *HyperText Markup Language (HTML)*

**<HTML>**

**<HEADER>**

**<TITLE>** Título do Documento **</TITLE>**

**</HEADER>**

**<BODY>**

**<H1>** Cabeçalho principal **</H1>**

O texto é em formato livre e deve ser estruturado em parágrafos. **<P>**

Novo parágrafo com texto.

**<PRE>** ... Texto pré-formatado... **</PRE>**

Clique

**<A HREF="ficheiro2.html">** aqui **</A>**

para ver outra página!

**</BODY>**

**</HTML>**

*Texto original em HTML*

Título do Documento

Cabeçalho principal

O texto é em formato livre e deve ser estruturado em parágrafos.

Novo parágrafo com texto.

... Texto pré-formatado ...

Clique aqui para ver outra página!

*O que um cliente WWW exibiria*

# WWW - Extensões

---

- Extensões mais comuns nos clientes WWW
  - Imagens com áreas clicáveis (*Client Side Image Maps*)
  - Suportar objectos “ativos”, com código executável
    - JavaScripts/JavaApplets, ActiveX, etc..
    - permite páginas mais elaboradas, e formulários mais complexos
  - *Publishing* (também tem de ser suportado no servidor!)
    - Um caso particular do anterior para publicação de páginas nos servidores...
- Extensão das funcionalidades dos servidores
  - CGI Scripts (*Common Gateway Interface*)
    - Ex: contadores do nº de acessos às páginas, tratamento de dados submetidos via formulário, *gateways* de acesso a bases de dados e outras aplicações...
  - *Server Side Includes*
    - Variáveis (*Ex: LastModified*) que se podem embutir no código HTML..
    - Não é necessário saber programar para os poder usar...
  - *Server Side Scripts*

# Extensões nos clientes WWW

---

- Como acrescentar funcionalidades novas a um browser Web?
  - acrescentando novas “*tags*” ao HTML
    - atropelos às normas... ou empurrões?
  - executando aplicações externas, designadas por “*Helper Applications*”
    - forma básica que implica riscos de segurança!
  - incorporando módulos especiais designados por “*Plug-Ins*”
    - integração perfeita com os *browsers* com alguns riscos de segurança e estabilidade...
  - interpretando instruções embebidas nas páginas HTML
    - JavaScript (Netscape), VBScript (Microsoft)
    - *Applets*

# Extensões nos clientes WWW

---

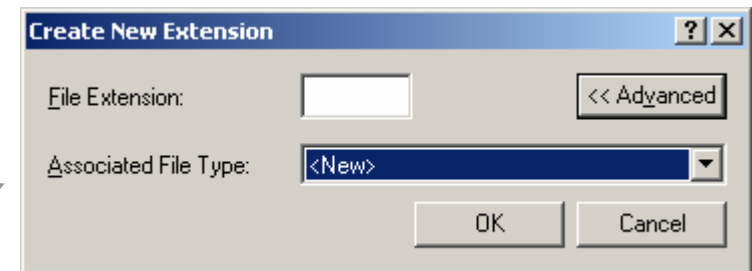
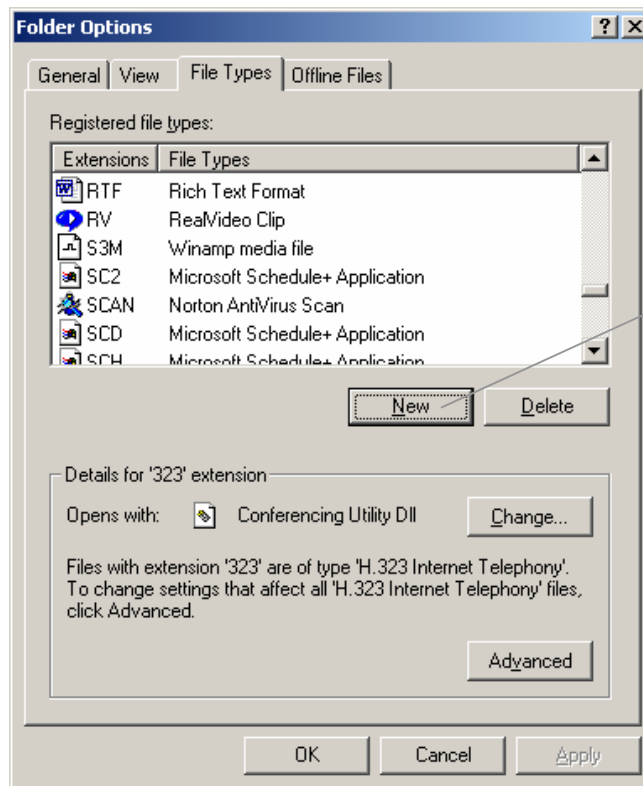
## *Helper Applications*

- A diversidade de formatos de documentos nunca poderá ser normalizada na totalidade:
  - obrigatória a utilização de tipos MIME
- Solução não-proprietária... suportada por qualquer browser
  - na configuração o utilizador indica duas coisas:
    - o tipo MIME associado a cada extensão de ficheiro
    - o programa que deve lidar com esse tipo MIME
- Permite “manipular” qualquer tipo de documento standard ou não!
- Desvantagens:
  - baixo desempenho
  - aplicações executadas devem coexistir em memória - “fat client” (clientes obesos)
  - dados não serão exibidos na janela no browser mas sim em janelas exteriores...
  - execução de aplicações externas, constitui um risco para a segurança dos clientes

# Extensões nos clientes WWW

## *Helper Applications*

- *exemplo para Internet Explorer (Folder options do Windows):*

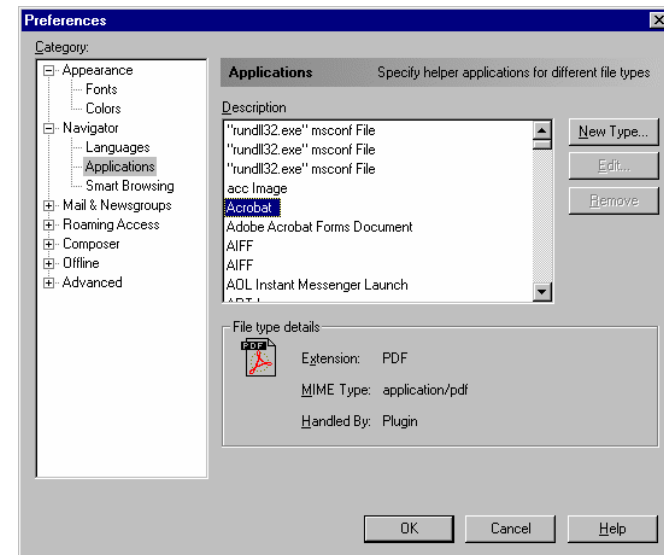




# Extensões nos clientes WWW

## Plug-Ins

- *Plug-Ins* são programas externos, construídos para “encaixarem” bem nos browsers respectivos...
  - respeitam um interface modular padrão definido pelo fabricante do browser
  - no entanto são produzidos por outras empresas (*third-parties*)...
  - os documentos/dados podem ser “manipulados” na própria janela do browser
    - solução mais integrada...
- Riscos:
  - um “*plug-in*” defeituoso pode bloquear o browser...
  - segurança: “*plug-in*” tem acesso a todos os dados disponíveis...



# Extensões nos clientes WWW

## *JavaScript, VBScript e JScript*

- As “*scripts*” parecem-se com macros (Lotus 1-2-3, Word, Excel,...) e são variantes adaptadas e simplificadas de linguagens de programação populares: JAVA e Visual BASIC
- Uma *script* pode ser embebida numa página HTML
- Os *browsers* integram um interpretador para as “*scripts*”:
  - *descodificam e executam as instruções após o download*

```
<HTML>
  <HEAD> <TITLE> Teste de JavaScript </TITLE> </HEAD>
  <BODY>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        RightNow = new Date();
        document.write("Olá");
        document.write("Hoje é dia " + RightNow.getDay() + "-" + RightNow.getMonth() + "-" +
          RightNow.getFullYear() + ". Chegou a esta página exactamente às : " + RightNow.getHours() +
          ":" + RightNow.getMinutes() + " e " + RightNow.getSeconds() + " segundos");
      -->
    </SCRIPT>
```

.....

# Extensões nos clientes WWW

---

## *Applets*

- Um *Applet* é uma espécie de pequena aplicação JAVA concebida para ser executada por um *Browser*
- Um *Applet* difere de uma aplicação em variados aspectos:
  - Por razões de segurança há muitas coisas que um applet não pode fazer, por exemplo aceder ao Sistema de Ficheiros da máquina local
  - Normalmente os métodos que o constituem não são invocados sequencialmente, é o browser ou *viewer* que, dependendo das circunstâncias invocam este ou aquele método. O *Applet* responde e retorna ficando a aguardar novas invocações.

# Extensões nos clientes WWW

---

## *Applets*

- Um applet muito simples:

```
import java.applet.*;
import java.awt.*;
public class FirstApplet extends Applet {
    public void paint(Graphics g)
        {g.drawString("Hello World",25,50);}
}
```

- A Applet tem de ser “compilada” com um compilador JAVA:

```
javac FirstApplet.java
... gerando um ficheiro “FirstApplet.class”
```

- Para visualizar um Applet é necessário que um ficheiro html o referencie

```
<APPLET code=“FirstApplet.class” width=150 height=100>
</APPLET>
```

# Exercício

---



- Consulte o menu “Ferramentas” do seu browser Internet Explorer para verificar as ditas cujas “Extensões” que estão instaladas e/ou a ser usadas pelo browser;
- Identifique uma extensão que permita mostrar documentos PDF, e diga qual o seu tipo;
- Que outros **tipos** de extensões existem disponíveis?

# Outros formatos

---

- Linguagens de definição de páginas comuns no WWW:

- **PDF** *Portable Document Format*

- formato universal da Adobe
- requiere um “Adobe Acrobat Reader” que é gratuito!...



- **DigitalPaper**

- requiere também um *Plugin* gratuito...

- *PostScript*

- formato mais adequado para impressão...

Para obter um melhor *layout* e melhores formatos, muitos utilizadores preferem disponibilizar os documentos em formatos de página... adequados tanto para consultas *on-line* como para impressão... Muito usual em *papers* na comunidade académica.

# Servidores Web

---

- O mercado tem vindo a criar as seguintes categorias de servidores Web (HTTP):
  - *Basic Web Servers*
    - servem apenas documentos (páginas HTML, Imagens, e outros)...
    - suportam **extensões** para serviços adicionais...
  - *Web commerce ou merchant servers*
    - orientados para o comércio electrónico: compras, vendas, transações financeiras, etc.
    - segurança (recorrendo a técnicas criptográficas) é fundamental!
  - *Web exchange servers*
    - integram funcionalidades associadas a troca de mail, fax, news, comunicação em grupo em diferido ou em tempo real, etc...
  - *Web-oriented database servers*
    - são servidores de bases de dados, preparados para o Web

# Extensões aos servidores HTTP

---

Apesar da oferta actual de produtos Web ser grande, torna-se muitas vezes necessário acrescentar pequenas funcionalidades aos servidores...

- Técnicas mais comuns:
  - CGI *Common Gateway Interface*
  - SSI *Server Side Includes*
  - SSS *Server Side Scripts*
  - API *Application Programming interfaces*



# Extensões aos servidores HTTP

---

## *SSI Server Side Includes*

- forma mais simples de interacção com o servidor...
- não exige conhecimentos de programação...
- Trata-se de linhas de texto incluídas nas páginas HTML, que o servidor deverá interpretar antes de devolver a página:

```
<HTML>
  <P> Page last modified on <!--#echo="LAST_MODIFIED"-->
  </P>
  ...
</HTML>
```

## *SSS Server Side Scripts*

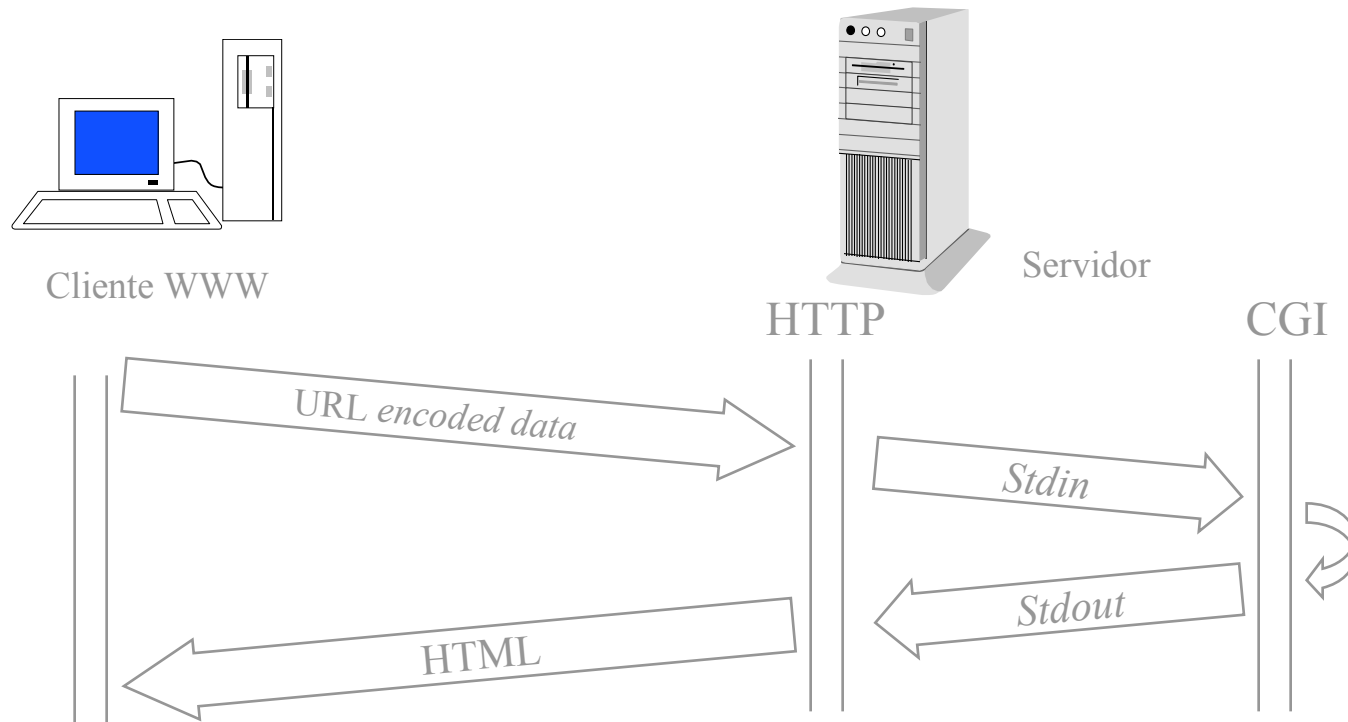
- as páginas com *scripts* para executar do lado do servidor costumam ter uma extensão diferente: .shtml, .php, .asp, .jsp; Exigem conhecimentos de programação;

JSP *Java Server Pages*  
ASP *Active Server Pages*  
PHP *Hypertext Preprocessor*

# Extensões aos servidores HTTP

## CGI *Common Gateway Interface*

- mecanismo que permite a um servidor HTTP aceitar pedidos de processamento vindos dos browsers, passando-os para programas que os vão executar e devolver os resultados na forma apropriada



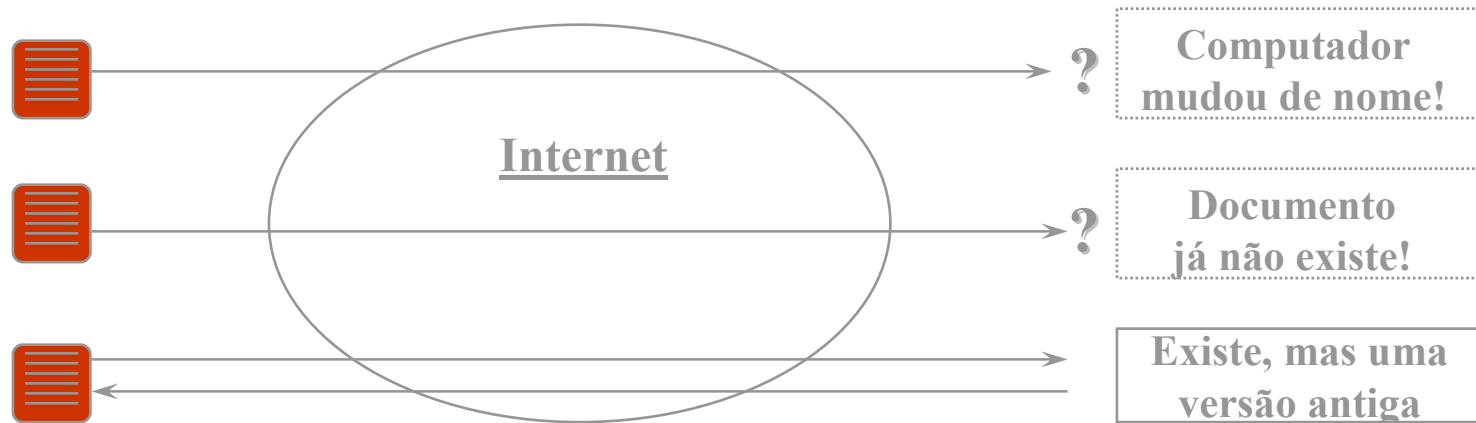
# World Wide Web

---

- Problemas com o excesso de informação:
  - Existe uma enorme quantidade de informação...
  - ... mas:
    - *muitas páginas não são actualizadas...*
    - *muitas tem incorrecções...*
      - Produzidas à pressa! Afinal é tão fácil publicar...
    - *informação não documentada...*
    - *muito difícil “pesquisar”, apesar dos “engenhos de pesquisa”*
    - *os “engenhos” devolvem muitas páginas indesejadas!*
      - Por vezes devido a estratégias dos seus próprios autores!...

# World Wide Web

- Problema dos *links* obsoletos:



## Soluções (?):

Encontrar formas de identificar os documentos,  
que sejam independentes da localização e do nº de cópias!

Uniform Resource Name (URN) (parecido com ISBN dos livros)

Arquivar toda Internet (Ex: Internet Archive e a máquina do tempo)

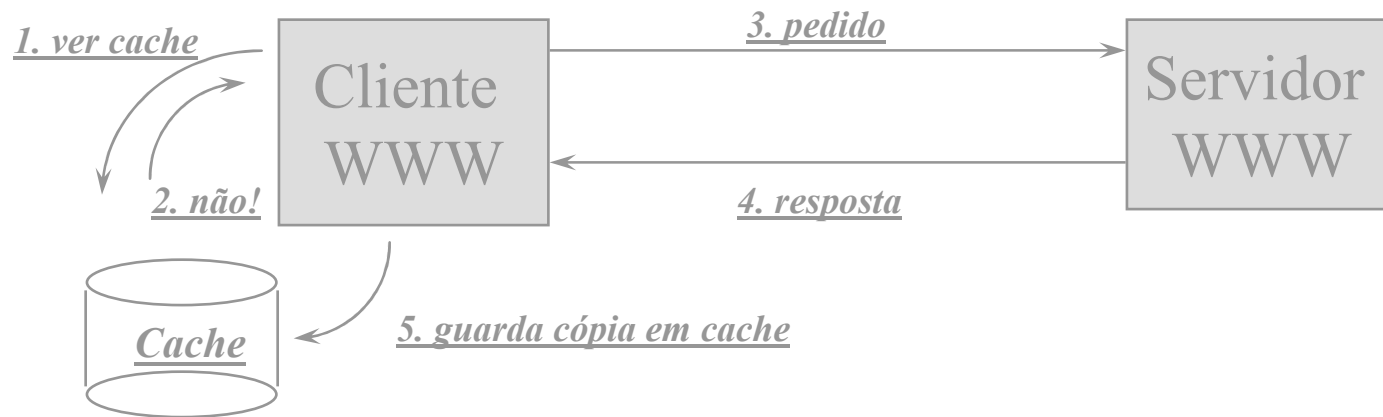
# World Wide Web

---

- Problema da **sobrecarga da rede**
  - Assume-se uma rede com cobertura mundial e capacidades infinitas de transferência que não existem!
  - Os utilizadores nem sempre são avisados do tamanho do documento antes da transferência
  - Os utilizadores e autores não estão conscientes das limitações da rede
  - O mesmo documento pode ser transferido mais do que uma vez pelo mesmo cliente:
    - *Proxy, Caching... e Mirroring*

# World Wide Web

- *Caching* no Cliente WWW

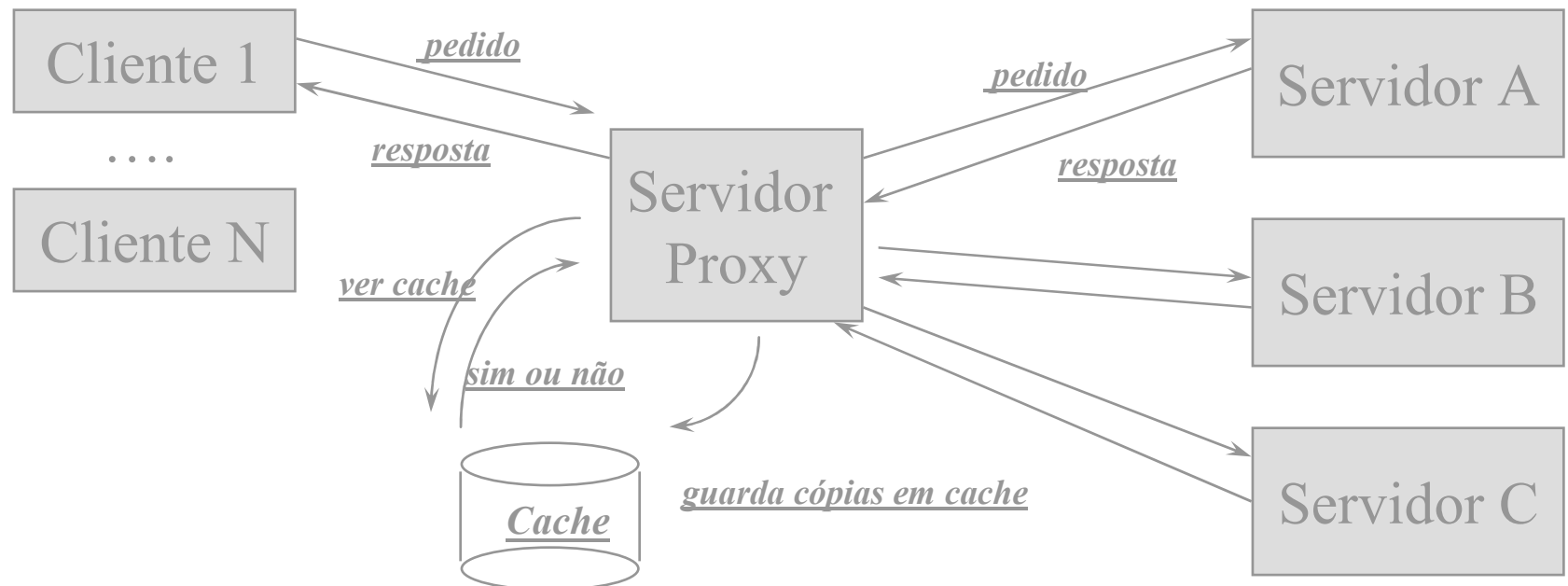


- o utilizador passa várias vezes pelo mesmos documentos...
- ...se os documentos mais recentes forem guardados em *cache*,
- podem posteriormente (dias) ser prontamente acedidos em disco
- contribui-se para diminuir a carga na rede e os tempo de acesso

# World Wide Web

- Será possível ter uma *cache* partilhada por todos os clientes de uma organização?

*Caching* num servidor próprio - Proxy - usado por todos



# World Wide Web

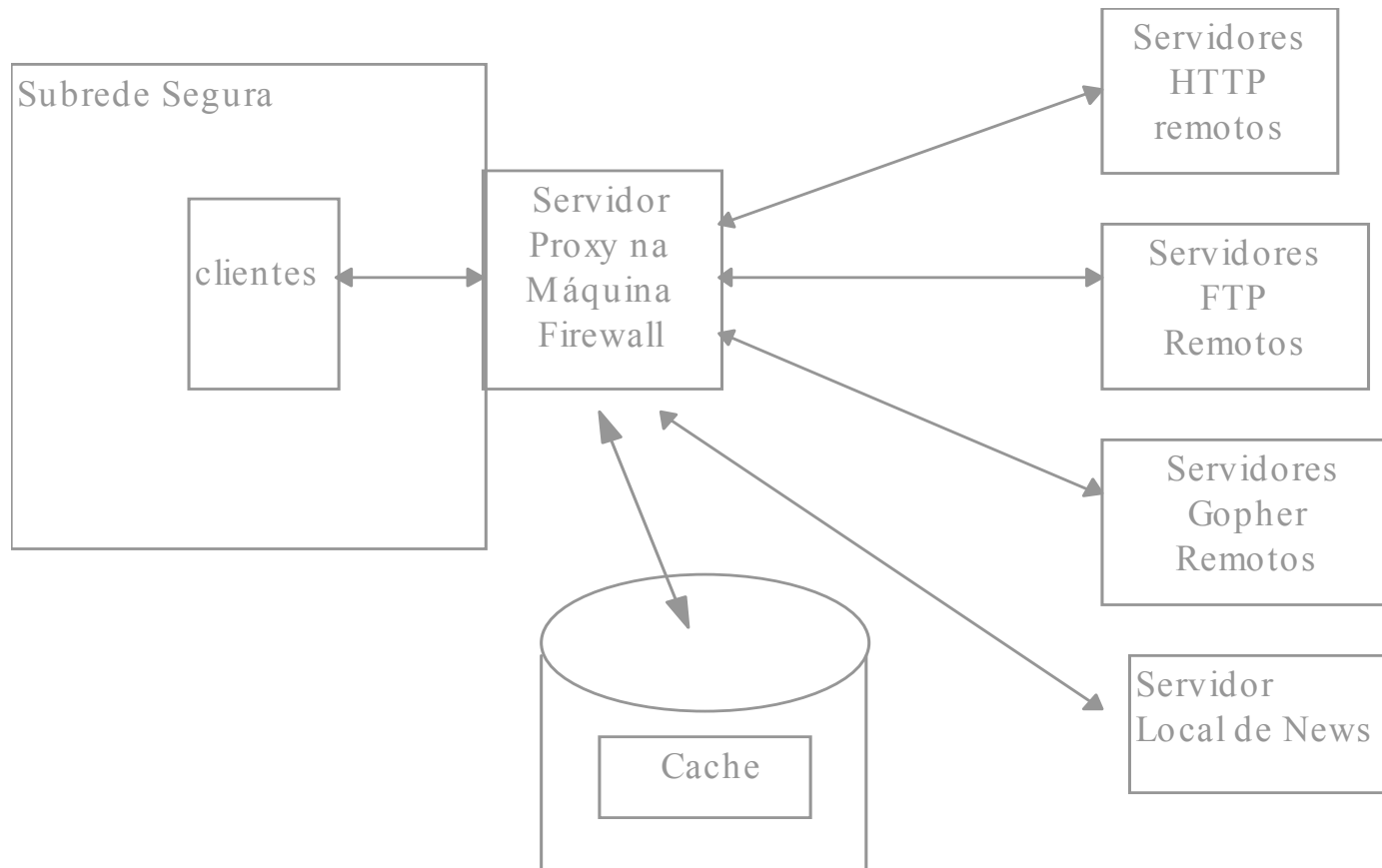
---

- Para que o *caching* seja possível ao nível da organização é necessário que os pedidos de todos clientes passem por um mesmo servidor - o servidor *proxy*
- Ao receber um pedido, o *proxy* encaminha-o ao servidor original e remete a resposta obtida ao cliente, mantendo uma cópia na *cache*
- Quando é pedido um documento repetido devolve ao cliente a cópia em *cache*
- Ao configurar o seu *browser* só tem vantagens em indicar qual é o *proxy* da organização
- O *proxy* também pode permitir a *browsers* colocados em redes sem conectividade Internet aceder a documentos sem quaisquer restrições



# World Wide Web

- *Proxy e Caching*



# World Wide Web

---

- Exercício de familiarização com os conceitos de *cache* e de *proxy*

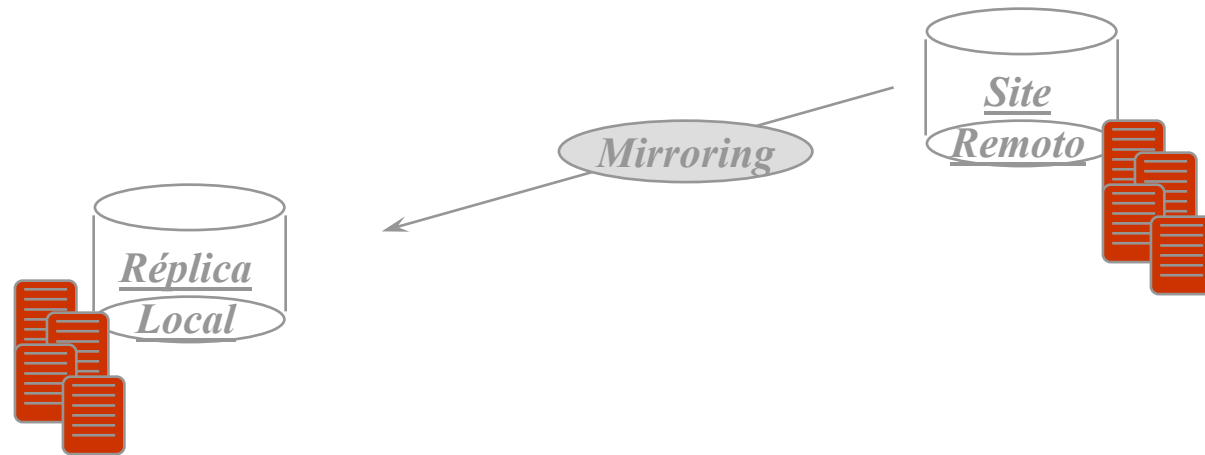


- Consulte o menu “Opções da Internet” do seu browser Internet Explorer e determine:
  - Se utiliza *cache* e que memória está reservada para o efeito
  - Se utiliza *proxy* e qual é esse *proxy*
  - Quais as vantagens/desvantagens de se usar *proxy* neste laboratório?

# World Wide Web

---

- *Mirroring*



- Objectivos:

- Disponibilizar *réplicas* mais próximo dos utilizadores, optimizando a utilização dos recursos da rede
- Distribuição de “carga” (nº pedidos) entre servidores...
- Tolerância a faltas: se um não estiver disponível, há substitutos

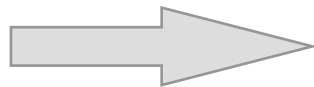
# World Wide Web

---

- *Robots* ou *Spiders*

Estes clientes especiais, surgem para tentar realizar tarefas que com o crescimento do WWW deixaram de poder ser feitas manualmente...

São **Agentes de Software** que exploram autonomamente alguma porção da Web, seguindo todas as ligações (*links*) existentes!




- Os percursos cíclicos são evitados...
- A porção a ser explorada pode ser demarcada por:
  - *domínios* ou *sites*
  - n° de *níveis*
  - combinações dos dois anteriores

# World Wide Web

---

- *Robots* - Utilização

Todos os *Robots* realizam as seguintes tarefas, continuamente:

- Obter página...
  - Procurar e recolher todos os URLs contidos nessa página
  - Fazer algum tipo de processamento
  - Próxima página ainda não visitada...
- 

É no processamento que fazem que os *Robots* se distinguem entre si

- Ex: **Robot de replicação** (*mirror*) cuja tarefa é gravar todas as páginas de um *site* no disco local, corrigindo todas as referências para continuarem válidas localmente!...

# World Wide Web

---

- Além da **replicação**, as tarefas mais comuns são:

## Estatísticas WWW

- Recolher dados estatísticos: n° de *sites*, documentos por *site*, tamanho médio dos documentos, n° de imagens por pág., etc.

## Manutenção

- Testar a integridade das ligações internas e externas (apenas os primeiros links para fora) e alertar para grandes alterações realizadas em documentos. Auxiliam a gestão de um *site*...

## Construção de Índices

- Criar enormes bases de dados que indexam os documentos e ficheiros existentes no WWW. Alguns catalogam todo o texto, outros apenas títulos ou resumos... São os mais populares!

# World Wide Web

---

Os chamados Agentes Inteligentes, são os novos robots pessoais que podem ser “treinados” pelo seu dono (assuntos de interesse) e depois “buscam” informações semelhantes na rede. Os mais sofisticados usam mesmo técnicas de Inteligência Artificial...

## Problemas com os *Robots*:

- “*inundam*” os servidores com pedidos
  - disparam pedidos muito rapidamente e por vezes em paralelo!
- “*entopem*” a rede
  - consomem largura de banda, prejudicando todos (sites visitados e também a rede local!)

## ● Soluções:

- ficheiro *robots.txt* contendo o nome dos indesejados!...
- detectar e impedir acessos em massa da mesma origem!...

# WWW - Pesquisa avançada

---

Ferramentas de pesquisa são programas que fornecem informação...

- Classificação das ferramentas de pesquisa:

Catálogos	Motores de Busca	Mistos	Multi-Engenho
Pesquisa hierárquica “por assunto”	Pesquisa “por palavras chave”...	Combinação dos anteriores	Ferramenta que interroga várias outras em paralelo

- Ou, do ponto de vista da indexação:

Indexação manual	Indexação total ( <i>full</i> )	Indexação parcial
Feita por utilizadores, que classificam em categorias!	Indexação automática de todas as palavras!	Indexação de apenas algumas palavras chave!



# WWW - Pesquisa avançada

---

- Dicas e truques para uma boa pesquisa
  - Adote uma estratégia baseada naquilo que sabe:
    - Use os catálogos, se sabe em que tópicos gerais se inclui aquilo que procura...
    - Use os motores de busca (c/ indexação total) se conhece coisas específicas como nomes ou palavras chave...
    - Características adicionais, como autor, localização, língua, etc, podem ser relevantes para auxiliar a pesquisa
  - Utilizar palavras pouco comuns e se possível diferenciadas...
  - Utilize vários motores de busca e procure sempre entender os resultados, distinguindo os “falsos acertos”

# WWW - Pesquisa avançada

---

- Utilização de alguns operadores para construir perguntas mais sofisticadas:
  - Operadores lógicos (AND, OR, NOT) e de proximidade (NEAR)
  - Sinais mais (+) e menos (-) para incluir ou excluir termos
  - Entre aspas (“...”) para definir uma frase exacta ou palavras muito relacionadas que devem ser procuradas em conjunto...
  - Parentesis rectos [..., ..., ] para agrupar palavras a serem pesquisadas quer isoladamente quer em conjunto
  - Operadores “mais como este” ou “more like this” (pesquisas por exemplos)
  - Letras maiúsculas, usadas para definir nomes próprios ou acrónimos...
- Nem todos os operadores são suportados por todos os serviços de pesquisa!

# WWW - Pesquisa avançada

---

- **Exemplo 1:** financiamento de investigações sobre o cancro
  - utilizando operadores lógicos para definir sinónimos:  
*(cancer OR oncol\*) AND research AND (grant\* OR fund\*)*
  - utilizando parentesis, ou aspas:  
*[cancer, oncol, research, grant, fund]*  
*"cancer research grant"*
  - operadores de proximidade  
*cancer NEAR research NEAR grant*
- **Exemplo 2:** informação sobre Warner-Lambert ou sobre o seu grupo de investigação farmacêutica Parke-Davis
  - Letras maiúsculas para nomes próprios e aspas para agrupar:  
*"Parke Davis" OR "Warner Lambert"*  
*"Warner-Lambert", "Parker-Davis"*

# WWW - Publicação

---

- As fases da publicação multimédia na Web:
  - Fase 1: Conceptualização.
    - definir objectivos
    - caracterizar a audiência
    - esquematizar o conteúdo
    - escolher uma estrutura
      - plana, *frames*, multimédia, ...
  - Fase 2: Construção das páginas.
    - recolher os recursos: scan das imagens, captura de vídeo e som
    - escolher uma ferramenta de autoria HTML
    - codificar e rever as páginas
    - adicionar links internos e externos

# WWW - Publicação

---

- Fase 3: Teste.
  - pelo próprio e por outros se possível...
- Fase 4: Publicar.
  - enviar ficheiros para o servidor... (por HTTP ou por FTP)
  - repetir os testes, com as páginas no servidor...
- Fase 5: Anunciar a publicação
  - junto de ferramentas de pesquisa e catalogação bem conhecidas...
  - outros: listas de distribuição, news, etc...
- Fase 6: Iniciar manutenção e actualização permanente
  - “download” das páginas a serem revistas
  - efectuar as alterações localmente
  - repetir publicação com envio para o servidor
  - teste das alterações...

# World Wide Web

---

- Principais aplicações do WWW:
  - Disseminação da cultura e da memória humana
    - Reconstituições históricas virtuais
    - Muses, galerias de arte e exposições virtuais
  - Educação (Ensino à distância)
  - Comércio (marketing, compras através da rede)
  - Trabalho cooperativo
    - conferências e edição simultânea
  - Bibliotecas electrónicas virtuais
  - Publicação electrónica...

# Segurança no WWW

---

## ● Que riscos?

### ● Nos servidores:

- “Bugs” ou configuração deficiente dos servidores podem levar a que utilizadores (remotos) não autorizados:
  - acedam a documentos confidenciais...
  - executem comandos no servidor que alterem o sistema...
  - obtenham informações que lhes permitam conduzir ataques mais tarde...
  - promovam ataques de “negação de serviço”...

### ● Nos clientes:

- programas “embebidos” que abortam o browser WWW, danificam o sistema, quebram a privacidade do utilizador ou simplesmente incomodam...
- utilização indevida de informação pessoal fornecida consciente ou inconscientemente pelo utilizador...

### ● Na comunicação entre cliente e servidor:

- interceptação dos dados enviados de e para o servidor, por observação dos pacotes, em qualquer ponto do percurso entre cliente e servidor...
- interceptação da identidade...
- manipulação dos dados trocados...

# Segurança no WWW

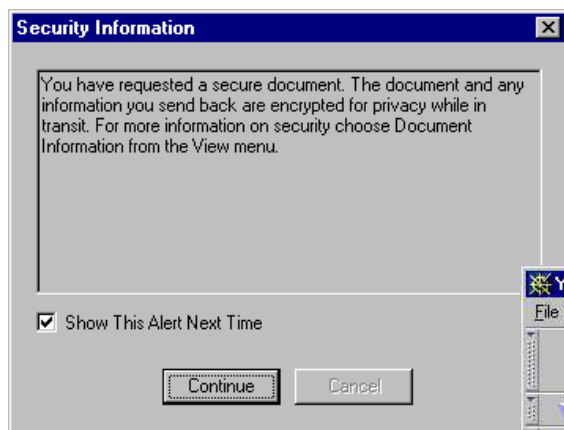
---

- Como minimizar os riscos?
  - é preciso ter consciência dos riscos...
    - .. sem paranóias ou despreocupações excessivas...
  - mecanismos de segurança
    - tornam os sistemas mais caros e mais difíceis de usar...
    - os mecanismos devem ser adequados aos riscos!
  - Na comunicação entre cliente e servidor, através de uma rede pública, os mecanismos de segurança de hoje em dia já dão garantias...
    - ... de modo a permitir até comércio electrónico!



# Segurança no WWW

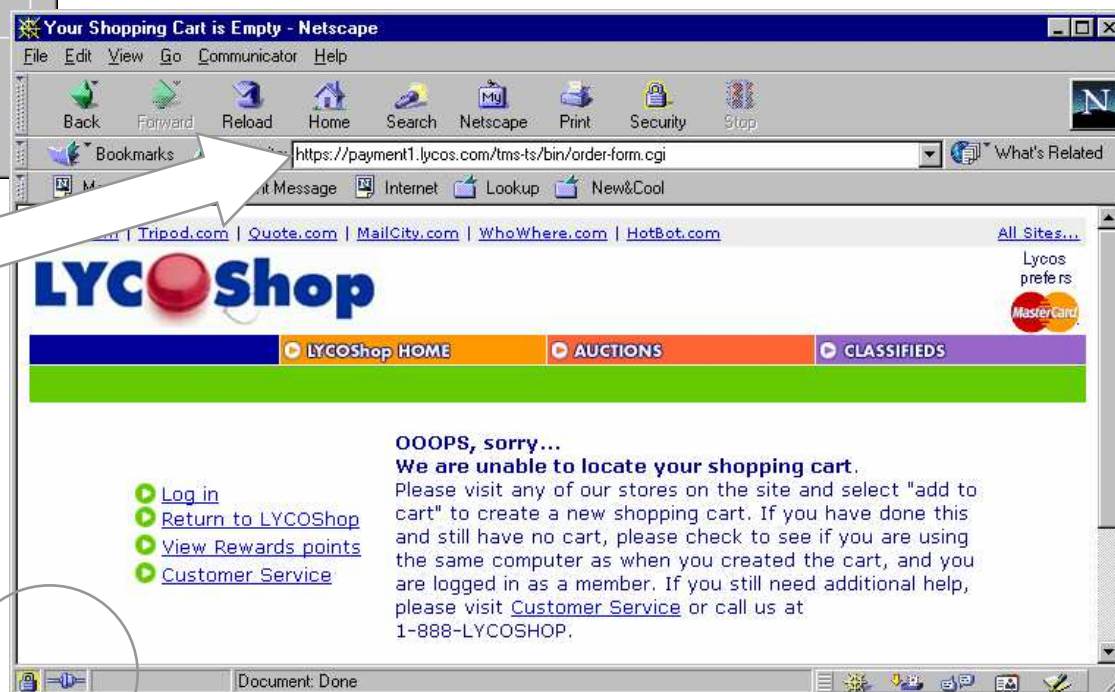
- Como identificar um acesso “seguro”?



O browser alerta sempre que o utilizador entra ou sai de um site seguro!

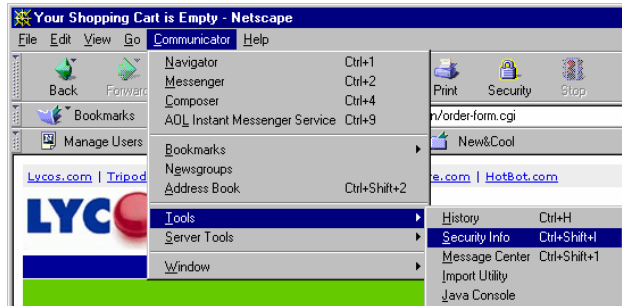
O URL de um site seguro começa sempre por https... (em vez do normal http)

“Cadeado” no canto inferior direito...



# Segurança no WWW

- Como identificar um acesso “seguro”? (cont.)



O browser indica que a página foi cifrada e mostra o “certificado digital” da organização...

# Segurança no WWW

---

## ● Que garantias para o utilizador?

- a página Web exibida não foi “observada” por ninguém no seu trajecto do servidor para o cliente, porque foi cifrada...

Garantia de confidencialidade

- a página Web não foi “alterada” no seu conteúdo original enquanto viajou na rede

Garantia de integridade

- o servidor que enviou a página pode ser identificado pelo “certificado digital” exibido...

Garantia de autenticidade da origem

- é preciso ter certeza que o certificado é válido!!
- Pode ser certificado por outra entidade conhecida (cadeia de certificação)

# Segurança no WWW

---

## “Cookies”

- O que são “cookies”?
  - É um mecanismo desenvolvido pela Netscape e rapidamente adoptado por todos, para manter sessões em HTTP:
    - natureza “stateless” do HTTP não permite sessões
    - cada interacção cliente-servidor consiste apenas num pedido e numa resposta
  - Um “cookie” é um pedaço de informação, que funciona como um testemunho:
    - o servidor gera um “cookie” e envia-o ao cliente...
    - o cliente deve enviar esse “cookie” de volta ao servidor em todos os pedidos que efectuar daí para o frente até expirar...
    - os cookies são guardados em disco...
  - Os “cookies” podem conter qualquer tipo de informação...
  - Cada “cookie” é válido para um *domínio* e para um *path*
    - exemplo:

www.shopping.com    FALSE /    FALSE 975239112    PGsi    Ka9II

# Segurança no WWW

---

## “Cookies”

- Os *cookies* e a privacidade do utilizador:
  - os cookies não podem ser usados para roubar informação armazenada no computador...
    - é o servidor que os envia!
  - ...mas permitem deixar “rasto” sobre o percurso do utilizador...
    - que páginas visitou, com que ordem, durante quanto tempo
    - em que dias e em que horas...
    - de que computador, com que sistema operativo,
    - que browser, etc...