

Tecnologias Web

1ª AULA, 5 de fevereiro

WWW

Roberto Lam, Instituto Superior de Engenharia,
Universidade do Algarve

rlam@ualg.pt

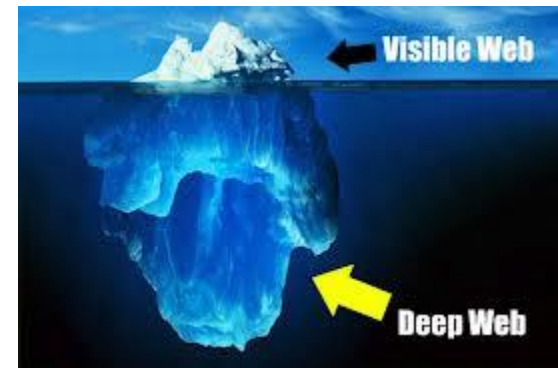
2023/24



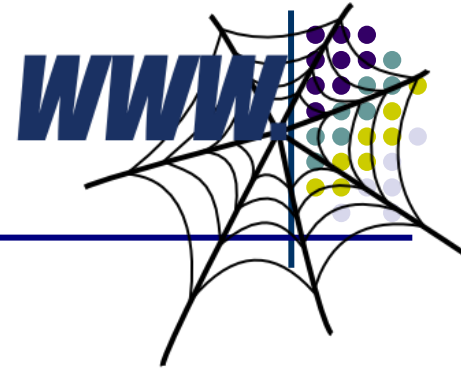
Motivação



- Técnicos em Tecnologias da Informação
- Conhecimentos sobre “World Wide Web”
 - Arquitectura base,
 - Protocolos,
 - Serviços,
 - Etc.



Avaliação



- Componente prática: 3 trabalhos (60%)
 - Peso de cada trabalho na avaliação prática: 20% do 1º Trabalho, 30% do 2º trabalho e 50% da cotação, do 3º trabalho.

DATAS dos trabalhos?

TP1 – 16 março; TP2 - 27 abril, TP3 - 25 maio 2024

Grupos: 2 alunos máximo, com apresentação e defesa individual

Penalização – máximo de 30%, da cotação do trabalho (na proporção linear ao intervalo compreendido entre a data de entrega e o final das aulas, 29 de maio de 2024).

- Componente teórica: teste | exame (40%)
- Nota final: 40% nota do teste | exame +
60% nota obtida nos trabalhos

Nota mínima em qualquer dos componentes 7 valores.

Histórico



- WEB *versus* INTERNET

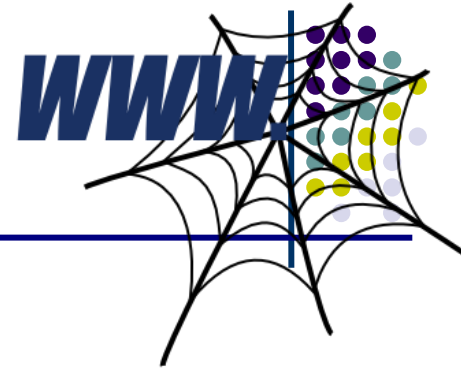
- INTERNET

- Ligação 'física' entre computadores
- Típicas aplicações Cliente-Servidor (Telnet, FTP, Email, etc.)
- Aplicações desenvolvidas ao nível TCP/IP

- WORLD WIDE WEB

- Aplicações sobre INTERNET
- Aplicações desenvolvidas sobre HTTP

Histórico



1993 – CERN, Organização Europeia para Pesquisa Nuclear, Berners-Lee (1989/90).

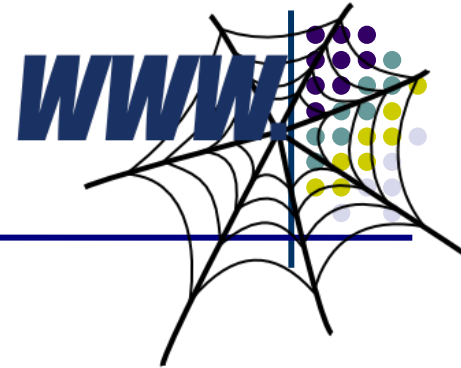
Surge conceito de WEB para divulgação de documentos científicos, através da utilização do conceito de “hipertexto”.

1993 – NCSA, National Center for Supercomputing Applications. Aplicação cliente em X-Windows **mosaic**.

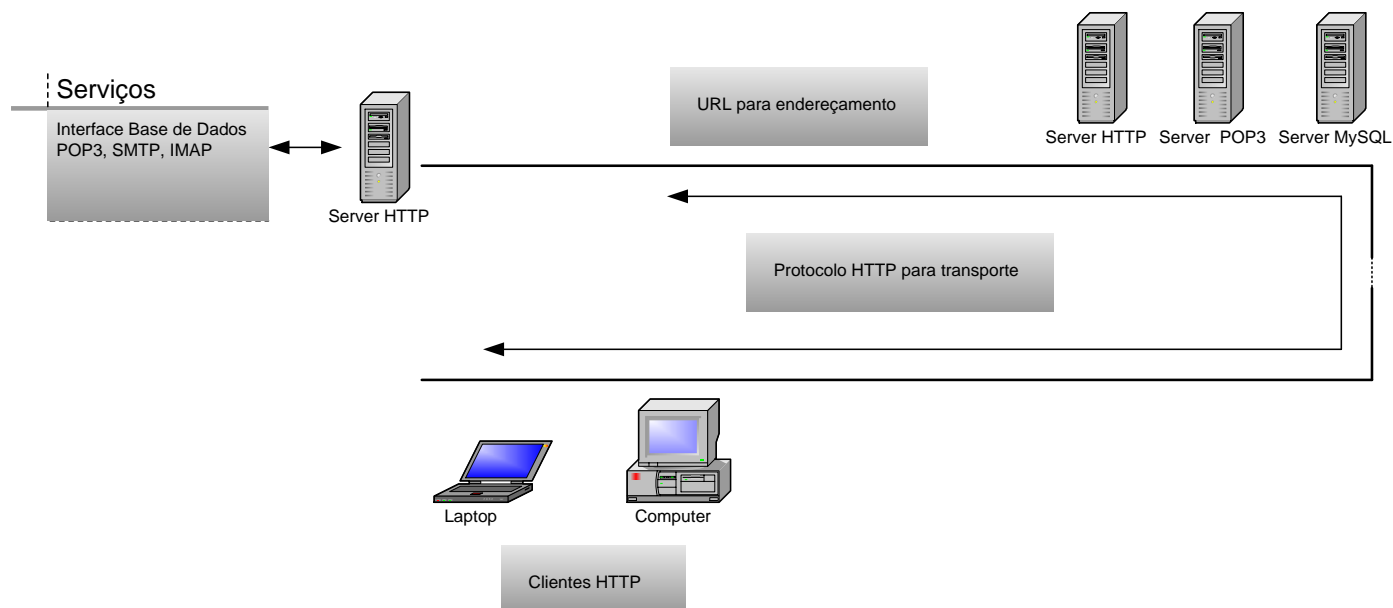
1994 – Netscape Communications apresenta aplicação cliente **navigator**.

1994 – o MIT e CERN apresentam o World Wide Web Consortium W3C (www.w3c.org) promove o desenvolvimento técnico e propostas de standards para a WEB (RFC) .

Modelo WEB



Objectivo: Transmissão, partilha de documentos, recursos electrónicos na INTERNET



Componentes do modelo WEB



URL – ***Uniform Resource Locators.***

Esquema de endereçamento de recursos na WEB

HTML – ***HyperText Markup Language.***

Meta linguagem que permite a construção de páginas para a WEB. As Páginas de HTML podem conter: formulários, hiperligações, etc.

HTTP – ***HyperText Transfer Protocol***

Protocolo HTTP permite a interacção Cliente-Servidor na difusão de páginas na WEB.

Componentes do modelo WEB



Servidor HTTP

Utilização do protocolo HTTP na difusão de documentos HTML, XML (*Markup Languages.*). Integração de serviços de Base de Dados, correio electrónico, *News, etc.*

Apache, IIS, NSCA HTTPd

Clientes HTTP

Aplicações que permitem a visualização de documentos HTML, XML.
Interacção dinâmica via javascript, vb script, java, .NET

Processamento

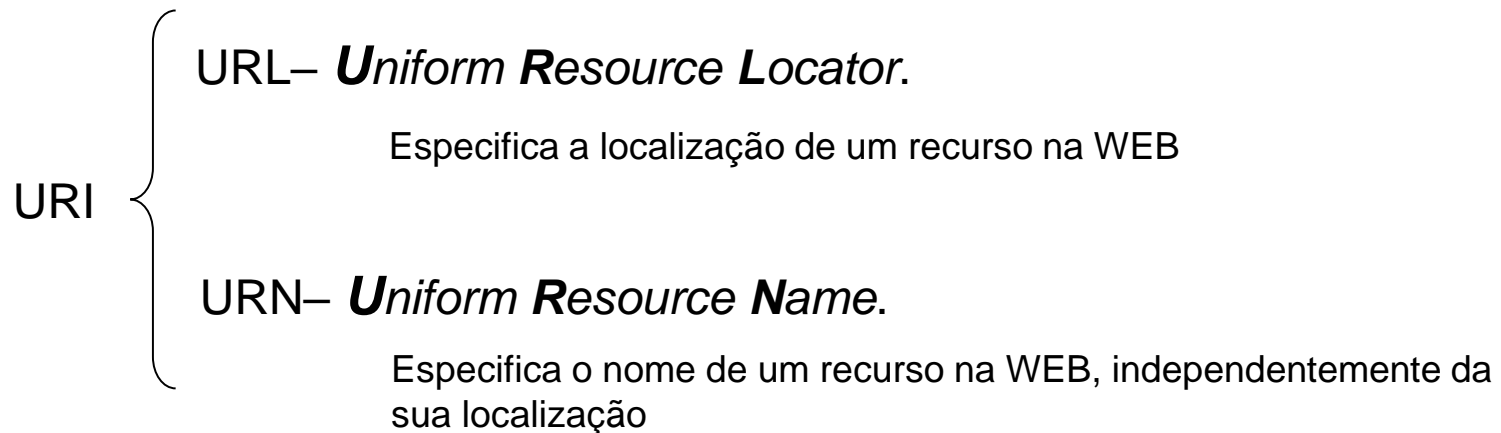
Servidor *versus* Cliente

URI: URL e URN



URI – **U**niform **R**esource **I**dentifier.

Esquema de endereçamento de recursos na WEB (w3C)



Registo de esquema de endereçamento:

<http://www.iana.org/assignments/uri-schemes.html>



URL



URL – **U**niform **R**esource **L**ocator.

Protocolo a utilizar no acesso ao servidor

Domínio onde está situado o servidor

Login e Password caso o protocolo o exija

Numero de porta de acesso

Portos:

Telnet 23

SSH 22

HTTP 80

FTP 21

http://www.iana.org:80/assignments/uri-schemes.html

protocolo domínio porto detalhes do recurso
pastas e ficheiro a solicitar

URL



ftp://login:password@10.11.140.223

protocolo
ftp

autenticação do utilizador

domínio (servidor ftp)

URL com identificador de marca (*tag*).

<http://algueres.no.alentejo/pub/grandola.html#local>

` Local de encontro `

HTML básico



Elemento raiz

HTML Hyper Text Markup Language

<html>

<head>

Elemento head

<title> :: Página pessoal....:</title>

</head>

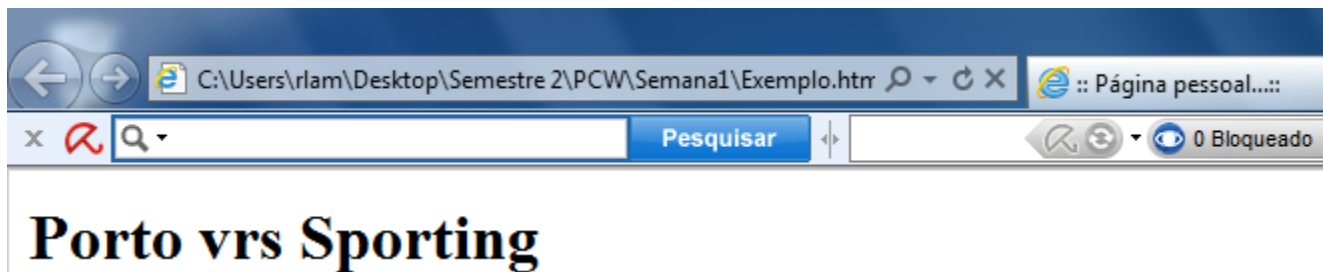
Elemento body

<body>

<h1> Porto vrs Sporting </h1>

</body>

</html>



Protocolo HTTP



HTTP - Hypertext Transfer Protocol

Protocolo HTTP 1.0

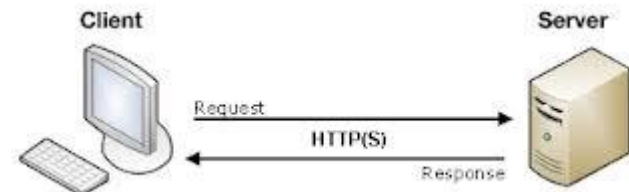
IMPORTANTE!

- solicitação / resposta (solicitação sempre efectuada pelo cliente)
- *stateless* (uma vez satisfeita a solicitação do cliente a conexão é encerrada)

Sessão típica HTTP 1.0

1. Aplicação cliente (browser) abre conexão
2. Aplicação cliente efectua a solicitação

A solicitação contém *request header*, onde é especificado o método HTTP a usar na transacção (GET, POST) . Também possui Informação das capacidades do browser.



Protocolo HTTP



3. Servidor envia o solicitado

A resposta contém *response header*, onde é especificado o estado da transacção (OK, ERROR,...) . Também possui Informação do tipo de dados enviados e os dados.

4. Servidor encerra a conexão

O servidor não mantém nenhuma informação sobre do processo
Existem browsers que mantém a conexão aberta através **keep-alive**

Uma transacção por conexão

Se uma determinada página possuir 10 imagens serão embelecidas 11 transacções

Conexões independentes

Para se manter alguma informação entre as diferentes solicitação / resposta
o uso <INPUT TYPE="hidden"... ou variáveis de sessão (PHP, ASP,..)

Protocolo HTTP



MÉTODOS DE SOLICITAÇÃO

GET

solicita um recurso

POST

envia dados (informações) ao servidor

HEAD

solicita o cabeçalho sobre um determinado recurso do servidor

PUT, DELETE, OPTIONS, TRACE, CONNECT

Protocolo HTTP



Exemplo sessão cliente/servidor HTTP 1.0

http://www.somehost.com/path/file.html

1) O *browser* abre um *socket* para **www.somehost.com** e envia

GET /path/file.html HTTP/1.0

Accept: text/plain

Accept: image/*

Accept: image/jpeg

Accept: */*

User-Agent: Mozilla/1.0

[blank line here]



REQUEST HEADER
Identificadores tipo

POST /path/script.cgi HTTP/1.0

From: frog@jmarshall.com

User-Agent: HTTPTool/1.0

Content-Type: application/x-www

Content-Length: 32

home=Cosby&favorite+flavor=flies

Protocolo HTTP



Exemplo sessão cliente/servidor HTTP 1.0

2) o servidor devolve:

HTTP/1.0 200 OK

Date: Fri, 31 Dec 1999 23:59:59 GMT

Content-Type: text/html

Content-Length: 1354

<html>

<body>

<h1>Happy New Millennium!</h1>

(more file contents) . . .

</body>

</html>

e termina fechando a conexão

Códigos devolvidos pelo servidor:

1xx *informação*

2xx *sucesso*

200 OK

3xx *indicação acções posteriores*

301 Moved Permanently

4xx *erro cliente*

402 Not Found

5xx *erro servidor*

500 erro servidor

Protocolo HTTP



Protocolo HTTP 1.1

- Permite múltiplas transacções numa única conexão,
- Permite enviar o recurso solicitado em fragmentos (recurso gerado *online*),
- Suporte de *cache* associada à transacção
- Múltiplos domínios num único IP

[HTTP Made Really Easy by James Marshall](#)



Perguntas?