



Programação de servidores na Web

Instrutores



Stefan Postolache

stefan.freya@gmailcom



Duarte Almeida

Duarte_Almeida@iscte-iul.pt

Requisitos

- ❑ Programação python básica
- ❑ Tecnologias Web
- ❑ Gestão de bases de dados

Conteúdos do curso

- ❑ Teoria essencial sobre aplicações Web
- ❑ Aprofundamento do uso da Shell
- ❑ Django
 - ❑ Instalação
 - ❑ Configuração
 - ❑ Fundamentos
- ❑ Construção de um Blog
- ❑ Clone do IMDB
- ❑ Implementação de uma Loja online
- ❑ Publicação de aplicações Python em ambiente de produção

Origens da Web

Hипертекст

Hipertexto e Hipermédia

- **Hiper**

- entidade multidimensional

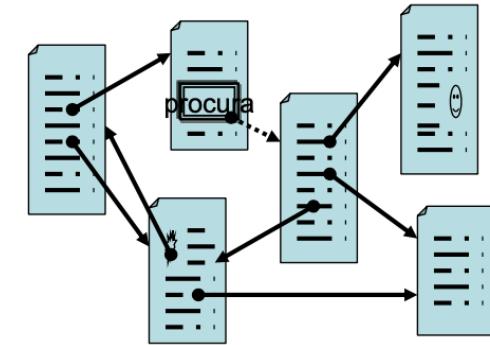
- **Hipertexto**

- Texto movimenta-se em muitas direções independentes
 - Texto aumentado com ligações

Hipertexto e Hipermédia

■ Hipertexto

- Parlavras podem ser estendidas em qualquer momento de forma a obter mais informação
- Extensão através de *hypertext links*



■ Hipermédia

- Conceito de hipertexto aplicado a outros meios
 - Imagens, vídeos, etc.

Hipertexto e Web

- Qual a inovação subjacente?
 - **Navegar** documento de forma **não linear?**
 - **Rapidez** com que se podem seguir as ligações
- Considerações de uma interface Hipertexto
 - Como identificar ligações?
 - Registar ligações seguidas pelo utilizador
 - Funcionalidades de navegação adicionais (para além de voltar para trás)



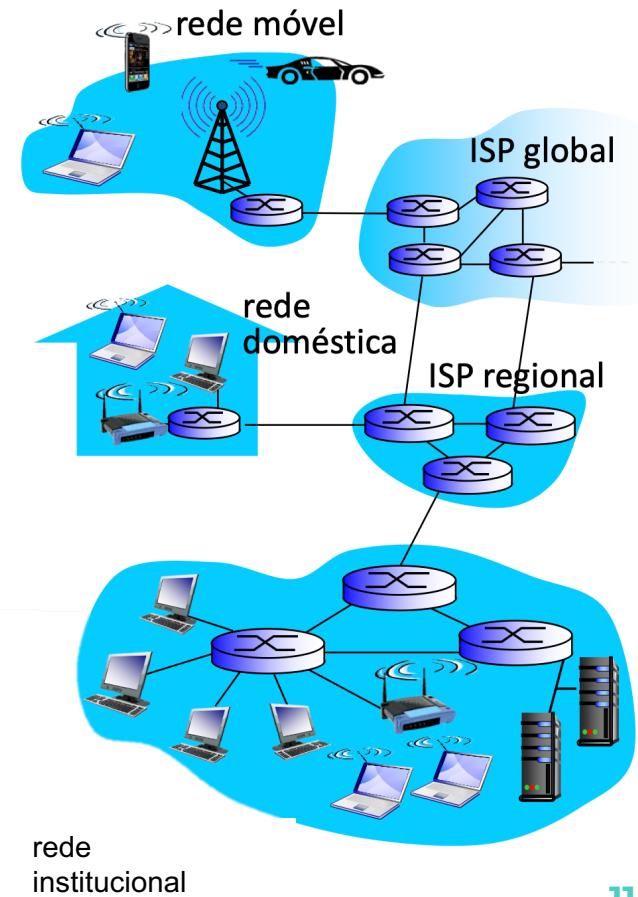
- O exemplo mais comum de hipertexto é a Web e a interface mais comum é fornecida pelos browsers Web

Internet

Rede das
Redes

O que é a Internet?

- Milhões de computadores ligados em rede
- Variados tipos de ligações físicas
 - Ligações sem fio 
 - Ligações com fio 
- Encaminhadores de pedaços de informação
 - Routers e Switches 



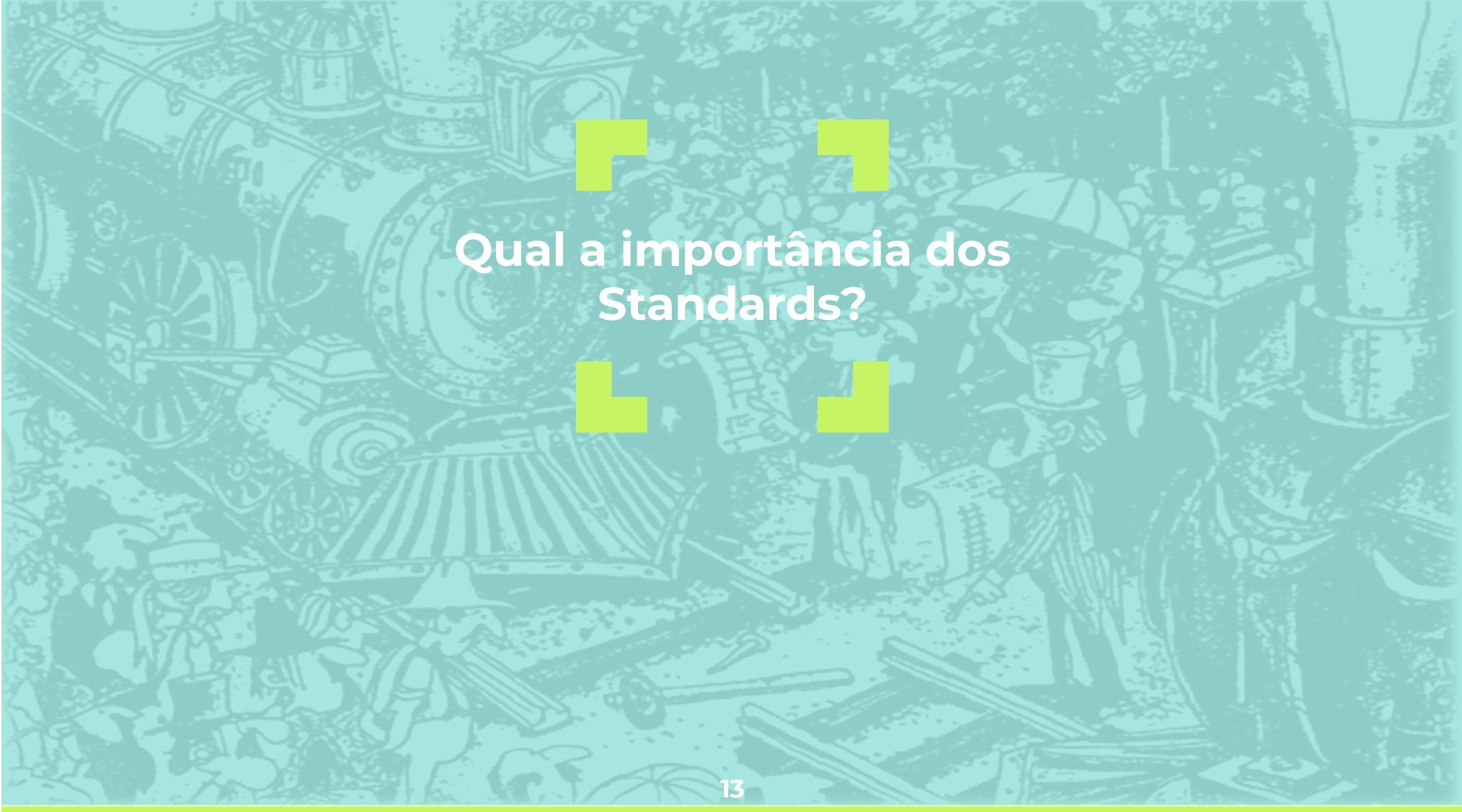
O que é a Internet?

□ **Protocols**

- Controlam o envio e receção de mensagens

□ **Standards**

- Garantem interoperabilidade entre equipamentos
- RFC: Request for comments
- IETF: Internet Engineering Task Force



Qual a importância dos Standards?

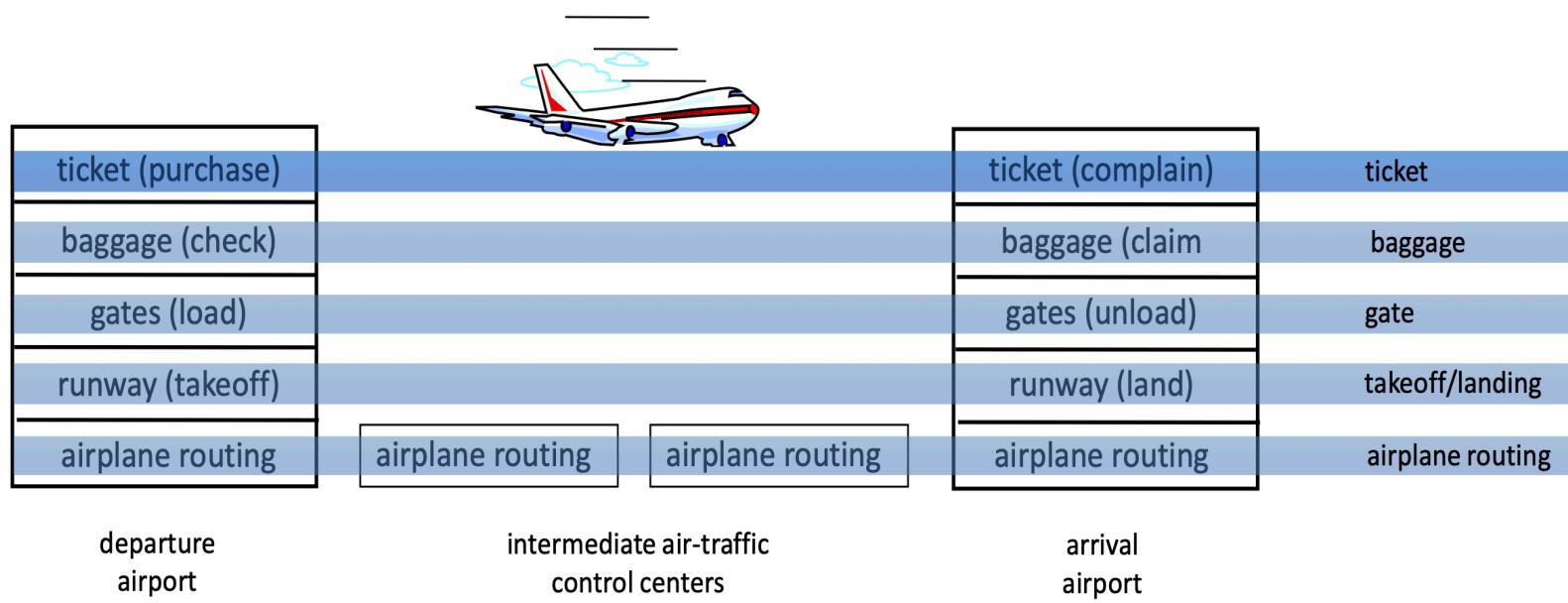
Arquitetura Em camadas

Como lidar com
a complexidade
da rede?

Organização em camadas

- Redes são complexas e compostas por vários elementos
- Organização em camadas permite-nos lidar com esta complexidade
 - Modularização
 - Independência
- Cada Camada concretiza determinado serviço
 - Ações na camada
 - Usando serviços de camadas inferiores
 - Fornecendo serviços a camadas superiores

Analogia com empresa de aviação



Pilha de protocolos da Internet

- ❑ Aplicação: Aplicações distribuídas que usam a rede: FTP, HTTP, etc.
- ❑ Transporte:
Transferência de mensagens entre processos remotos
 - ❑ TCP, UDP



Pilha de protocolos da Internet

- ❑ Rede: encaminhamento de pacotes da origem ao destino: IP, protocolos de encaminhamento
- ❑ ligação: transferência de tramas entre elementos de rede vizinhos.

Ethernet



Pilha de protocolos da Internet

- ❑ Físico meio de transporte de bits

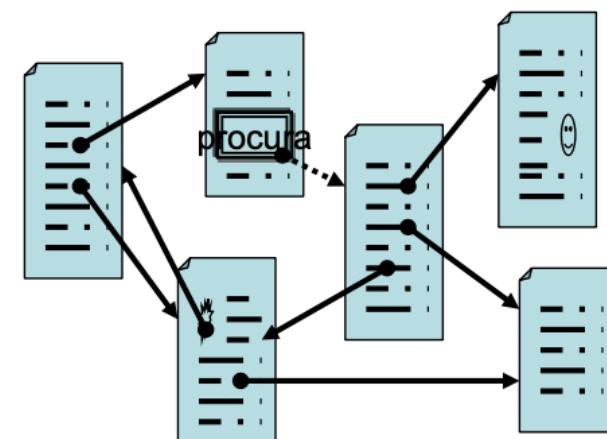


World Wide Web

Conceitos
fundamentais

World Wide Web – O que é?

- ❑ Um sistema hipermédia
- ❑ Um sistema de informação
- ❑ Um sistema distribuído



World Wide Web – principais objetivos

- Acesso global
- Simplicidade
- Versatilidade

Internet vs Web

□ Internet

- Conjunto de redes de transmissão de informação entre sistemas informáticos que estão interligados entre si

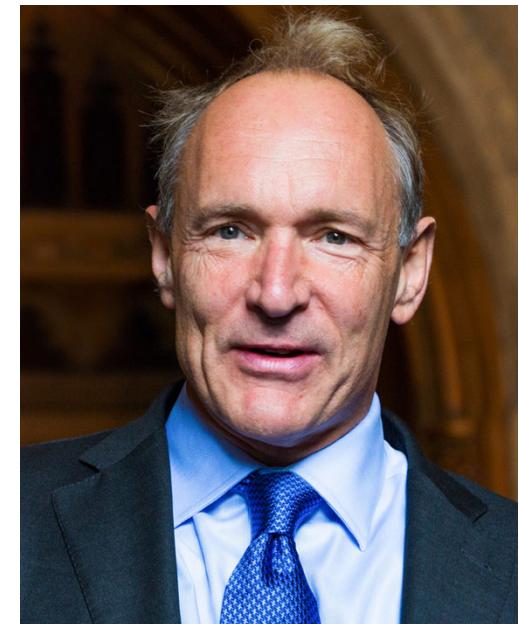
□ Web

- Conjunto de protocolos para criar um espaço hipermédia sobre a internet

Origem

1990

- Ferramentas necessárias para o funcionamento da Web
 - HTTP (HyperText Transfer Protocol)
 - HTML (HyperText Markup Language)
 - Web Browser
 - HTTP Server
 - Web Server
 - Primeira Página Web



Time Berners Lee

Arquitetura

- **Aplicações cliente servidor suportadas pelo protocolo TCP/IP(HTTP)**

- Esquema comum de endereçamento
- Protocolo comum para articulação
- Negociação de formatos

- **Servidores Web**

- Diversas implementações
 - Apache
 - NGINX
 - CERN HTTPD
 - Microsoft Internet Server

Arquitetura Cliente Servidor

■ Servidor

- Sempre ligado
- Endereço IP permanente
- Localizados em grandes data centers

■ Clientes

- Comunicam com o servidor
- Não estão sempre ligados
- Endereços IP podem ser dinâmicos
- Não comunicam diretamente entre si

Cliente

- Interagem por interfaces interactivas orientadas para navegação (browsing em hipertexto)
- Diversas implementações:
 - Mozilla, Opera, Safari, MS Internet Explorer, Chrome, Mosaic, Lynx

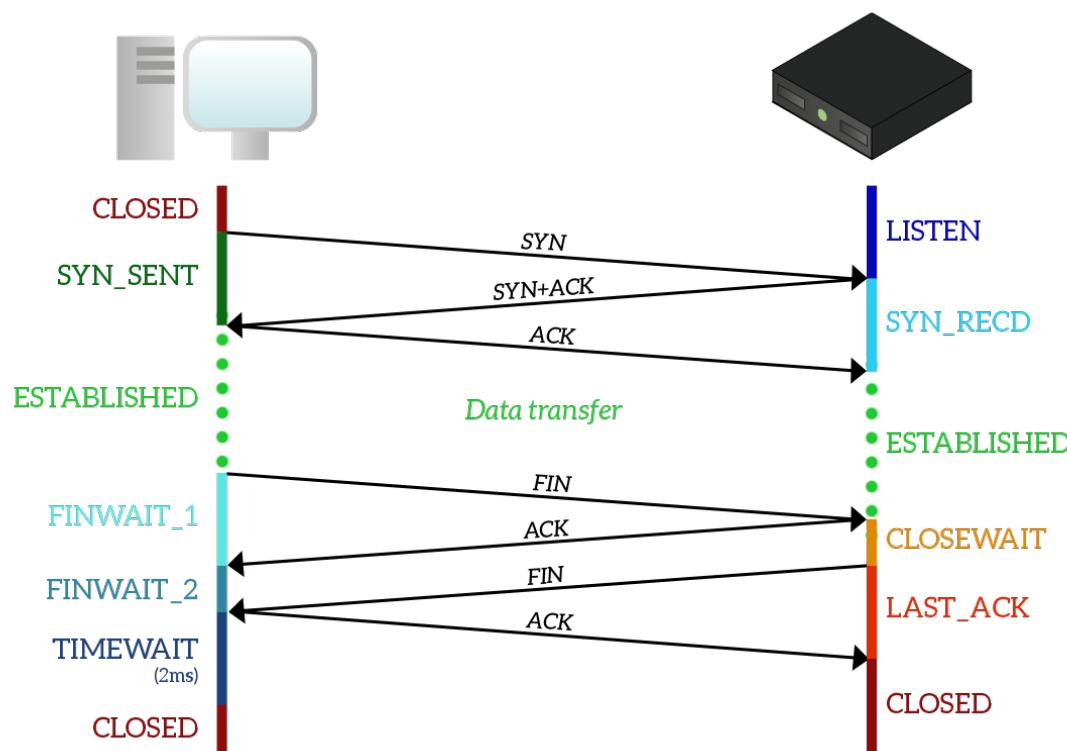
Servidor

- Repositórios de conteúdos de informação multimédia
- Estrutura da informação baseada na ligação de documentos por hipertexto/hipermédia

TCP – Transmission Control Protocol

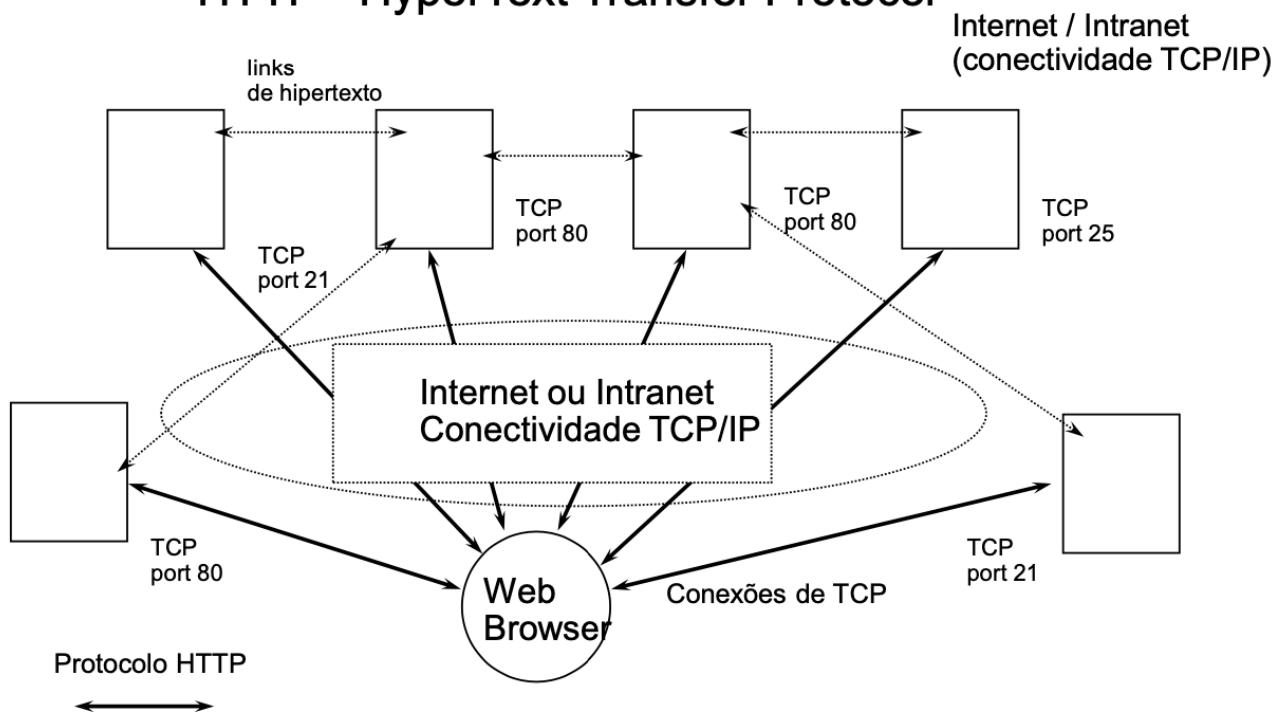
- ❑ Informação entre processos remotos é transmitida de forma confiável
 - ❑ Cada transmissão envolve reconhecimento que conteúdo foi recebido
 - ❑ Não pretende ser eficiente
-
- ❑ Dados são transformados em pacotes:
 - ❑ Cabeçalhos – Destino
 - ❑ Corpo – Dados
 - ❑ Checksum – garantir minimização de erros

TCP – Transmission Control Protocol



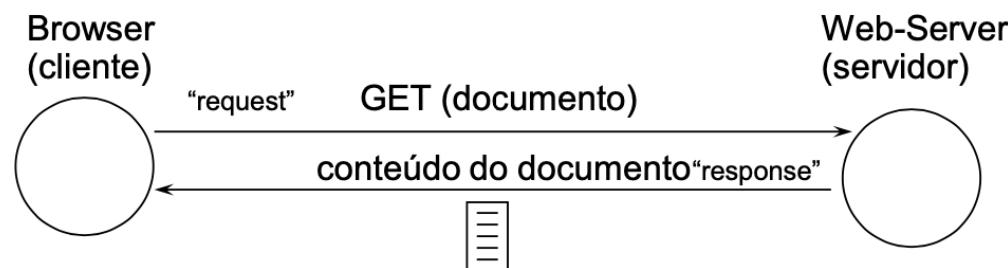
WWW: Arquitetura Geral

HTTP - HyperText Transfer Protocol

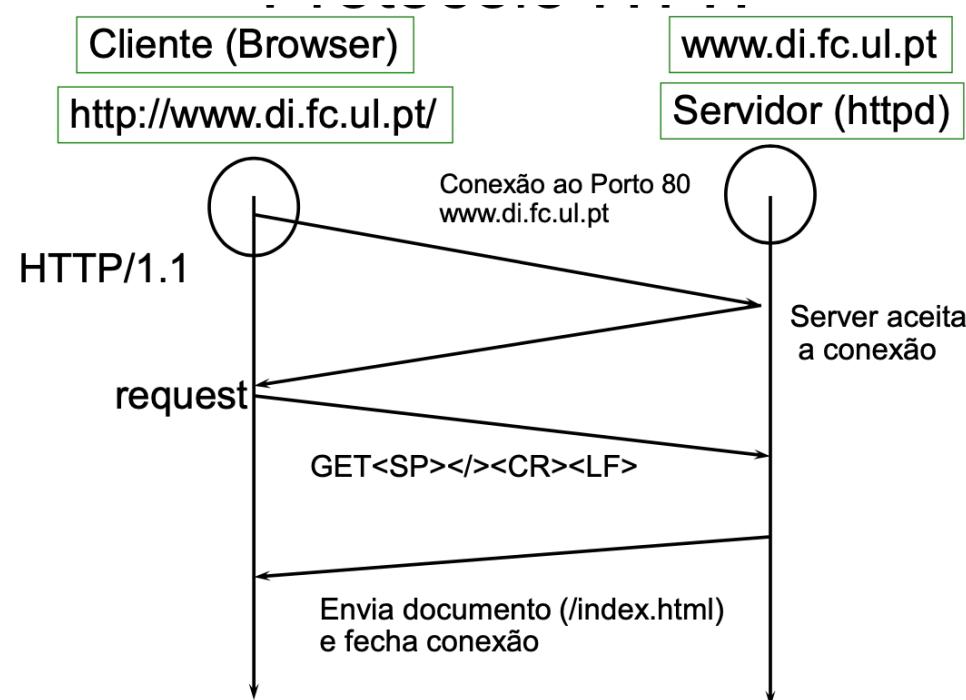


Protocolo HTTP

- ❑ Protocolo de base da WWW que é usado nas trocas de documentos entre clientes e servidores
- ❑ Suportado por TCP



Protocolo HTTP



Protocolo HTTP – Métodos mais relevantes

- **GET:** submete um pedido simples de um recurso ao servidor Web
- **POST:** submete um pedido ao servidor, incluindo informação introduzida pelo utilizador que deve ser processada pelo servidor. A informação do utilizador é anexada ao pedido sob a forma do corpo do pedido

Protocolo HTTP - exemplo

- Exemplo:

- Para se visualizar a página cujo URI é
`http://www.di.fc.ul.pt/~afalcao/exemplo.html`
 - Pedido do Browser ao servidor `www.di.fc.ul.pt` :
 - `GET /~afalcao/exemplo.html HTTP/1.1`
 - RespostaServidor:
 - `HTTP/1.1 200 OK` ou
 - `HTTP/1.1 404 Not Found` –...

Pedido HTTP - Cabeçalhos

□ Cabeçalhos genéricos

- Aplicados a pedidos e respostas, troca de informação entre cliente e servidor
 - Ex: Date: Sun, 03-Dec-00 15:04:12 GMT

□ Cabeçalhos do pedido

- Aplicados a pedidos
- Especificam o tipo de informação, conjuntos de caracteres e idiomas

Pedido HTTP - Cabeçalhos

- **Cabeçalhos de resposta**
 - Aplicados a respostas
 - Informação específica sobre a resposta
- Ex: WWW-Authenticate aparece associado ao código 401 que é enviado quando o utilizador tenta aceder a um recurso protegido

Pedido HTTP - Cabeçalhos

- **Cabeçalhos de entidade**

- Aplicam-se ao conteúdo que é transferido no corpo de um pedido ou resposta
 - Permitem caracterizar o conteúdo

- Ex: o cabeçalho **content-length** permite especificar o no de Bytes associados ao conteúdo; **content-type: text/html** indica que o tipo de dados da resposta é texto

Pedido HTTP - Exemplos

```
GET /tutorials/other/top-20-mysql-best-practices/ HTTP/1.1
Host: net.tutsplus.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US;
rv:1.9.1.5) Gecko/20091102 Firefox/3.5.5 (.NET CLR 3.5.30729)
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: PHPSESSID=r2t5uvjq435r4q7ib3vtdjq120
Pragma: no-cache
Cache-Control: no-cache
```

Pedido HTTP - Exemplos

```
HTTP/1.x 200 OK
Transfer-Encoding: chunked
Date: Sat, 28 Nov 2009 04:36:25 GMT
Server: LiteSpeed
Connection: close
X-Powered-By: W3 Total Cache/0.8
Pragma: public
Expires: Sat, 28 Nov 2009 05:36:25 GMT
Etag: "pub1259380237;gz"
Cache-Control: max-age=3600, public
Content-Type: text/html; charset=UTF-8
Last-Modified: Sat, 28 Nov 2009 03:50:37 GMT
X-Pingback: http://net.tutsplus.com/xmlrpc.php
Content-Encoding: gzip
Vary: Accept-Encoding, Cookie, User-Agent

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

Códigos de resposta

- ❑ 1xx: **informação**
 - ❑ 100 Continue (continuar a enviar o pedido)
- ❑ 2xx: **sucesso**
 - ❑ 200 OK (pedido bem sucedido, recurso no corpo)
- ❑ 3xx: redireccionamento
 - ❑ 301 Moved Permanently (recurso movido para outro URL)
 - ❑ 302 Found (recurso movido temporariamente para outro URL)
 - ❑ 304 Not Modified (documento não modificado)

Códigos de resposta

- ❑ 4xx: **erro no cliente**
 - ❑ 400 Bad Request (servidor não percebeu pedido, erro sintaxe)
 - ❑ 401 Unauthorized (este pedido necessita de autenticação)
 - ❑ 404 Not Found (recurso não existe no servidor)
- ❑ 5xx: **erro no servidor**
 - ❑ 503 Service Unavailable (servidor não consegue responder ao pedido por estar sobrecarregado ou em manutenção)

E continua...

- Motores de busca cruciais
 - Torna-se fundamental saber como encontrar a informação
- Redes sociais dominam a Web
- Os conteúdos deixam cada vez mais de ser centralizados
 - O utilizador é o principal produtor de informação (blogs, wikis, posts, tweets)
- **Novos dados:** Satélite, mapas, vídeos

E continua...

- Acesso móvel lança novos desafios
 - Largura de banda (ainda) limitada
 - Interfaces diferentes

Páginas Web vs Sites Web

- Web sites mantém consistência temática
- Preocupações estéticas. Além do conteúdo temático deve ter uma aparência comum ao longo das suas páginas
- Sites web apresentam uma arquitectura planeada e definida
- Não é comportável ter todas as páginas web de um website criadas estaticamente

Web sites e aplicações Web

- O inicio da Web era sobretudo conteúdo estático
- Mais tarde começaram a aparecer scripts que executavam conteúdo dinâmico:
 - Procuras de informação
 - Pedidos a bases de dados
 - Processamento de informação
 - E-Commerce

Web sites e aplicações Web

- Mudanças de paradigma
 - (2004) Web 2.0 – Utilizadores criam o conteúdo (blogs, YouTube, Tweets, Facebook, etc)
 - (2005) AJAX (Asynchronous JavaScript and XML)
 - Permite modificar páginas já presentes no Browser com conteúdo dinâmico proveniente do servidor

WWW: Arquitectura Geral

□ Intervenientes

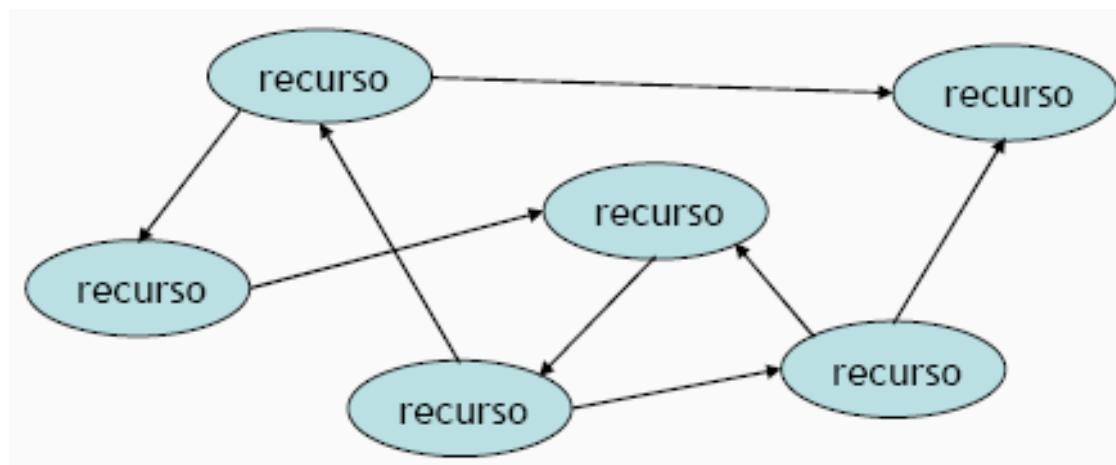
- Browser – cliente HTTP
 - Programa que estabelece uma conexão com o objectivo de formular pedidos
- Servidor Web – servidor HTTP
 - Programa que aceita ligações e responde aos pedidos recebidos
- Proxy
 - Programa que serve de intermédio e que realiza cache de conteúdos

Identificação de Recursos

Como
Identificar
websites

Identificação de Recursos

- O utilizador, através de um browser, “vê” a Web como um conjunto de recursos interligados, através de URLs



Identificação de Recursos

- URI - Uniform Resource Identifier
 - String de caracteres utilizada para identificar ou nomear um
 - Sintaxe para identificadores de recursos
- Sintaxe genérica de um URI
 - <esquema> : <parte específica esquema>

Tipos de URLs

- URL - Uniform Resource Locator
 - Identificação por localização
 - Exemplo: <http://www.w3c.org>
- URN - Uniform Resource Name
 - Identificadores de recursos pelo nome, independentes da localização
 - Exemplo: <urn:isbn:0-395-36341-1>

Sintaxe URI

<URI nome esquema> : <parte específica do esquema>

- ❑ Nome do esquema
 - ❑ Exemplos: http, ftp, mailto, URN, tel, file, ...
- ❑ Parte específica do esquema
 - ❑ A sintaxe e a semântica é determinada pela especificação do esquema
- ❑ Existem algumas regras quando o esquema tem uma estrutura hierárquica
- ❑ Reserva de certos caracteres especiais

Sintaxe URI

□ Caracteres e regras:

- Nome do esquema - letra seguida de uma combinação de letras, dígitos e o sinal "+", ".", ou "-"
- Caracteres US-ASCII
- Caracteres que podem ser reservados: ":", "/", "?", ":", "@", "=", "&", "#"
- No caso de aparecerem fora do contexto para o qual foram reservados, devem ser codificados: %<índice_hexadecimal>
- Se o esquema tiver links relativos e uma estrutura hierárquica, os componentes da hierarquia devem ser separados por "/"

URI Sintaxe Genérica

- **Scheme** – Designa o protocolo a usar (e.g. http, ftp,...) seguido de ://
- **Host** - Endereço IP (numérico ou via DNS) do serviço a ser acedido (e.g. 93.111.8.156 ou www.fc.ul.pt)
- **Port** – parte opcional do URL, a indicar o porto do servidor que receberá o pedido. Cada protocolo tem um porto por omissão: http: 80, ftp: 21,...)
- **Path** - Caminho lógico. Indica o local onde o recurso pretendido está

URI Sintaxe Genérica

- **url-params** – parametros do url, usado ocasionalmente para identificadores de sessão (pares nome: valor)
- **Query-string** – pares nome-valor que representam os parâmetros dinâmicos associados ao pedidos
- **anchor** – marcador posicional referente a posição específica do documento

Cookis

Como guardar
informação
do cliente

Protocolo HTTP - Limitações

- O protocolo não possibilita a existência de uma sessão de utilizador. Não guarda informação entre dois pedidos sucessivos.
- A interacção é sempre iniciada por um pedido do cliente.

Protocolo HTTP - Limitações

- Existem duas formas de resolver o problema
 - Armazenar a informação de estado no servidor Web sobre a forma de dados da sessão.
 - Armazenar a informação de estado no cliente sobre a forma de cookies.
- **Dados de sessão**
 - Associação de um identificador de sessão ao 1º pedido HTTP de um novo cliente
 - Nos pedidos subsequentes o browser comunica esse identificador.

Uso de dados de sessão no servidor

- ❑ Usa-se a informação de sessão no servidor quando
 - ❑ Esses dados não podem estar no cliente
 - ❑ Dados apenas relevantes para aquela sessão
 - ❑ Não há necessidade de segurança
- ❑ Linguagens de desenvolvimento web permitem a criação de variáveis de sessão:
 - ❑ E.g. PHP: \$_SESSION

Protocolo HTTP - Cookies

□ O que são Cookies?

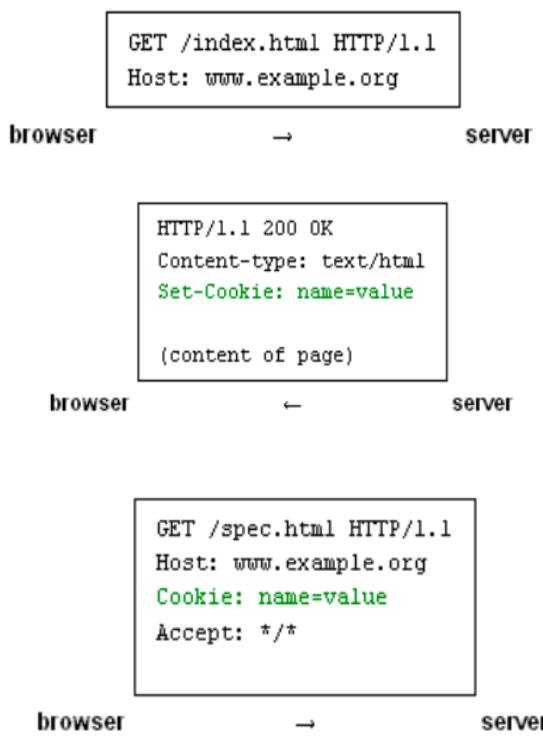
- Mecanismo que permite a aplicações HTTP servidoras guardar e obter informações sobre o cliente.
- Manutenção de informação de estado sobre o cliente.
- Entidades criadas por um programa executado pelo servidor e enviadas ao cliente para serem armazenadas no respetivo SF e num local acessível pelo Web Browser

Que informação contém um Cookie?

- Informação sobre o estado do cliente na forma de um par **nome=valor**
- Conjunto de URLs para os quais são válidos
- Data de validade (para o caso de Cookies persistentes)

Exemplo de cookies

- ❑ Pedido HTTP
- ❑ Resposta HTTP
- ❑ Novo Pedido
 - ❑ O servidor sabe que este pedido é relacionado com o anterior





O futuro profissional
começa aqui

iscte

INSTITUTO
UNIVERSITÁRIO
DE LISBOA



emprego
digital



UPskill