UPT UNIVERSIDADE PORTUCALENSE

CONTEÚDO

1. História da Internet
2. A World Wide Web (www)
3. Funcionamento da Web
4. Web 1.0, 2.0, 3.0 e 4.0
5. A Internet é uma rede global
6. Modelo Cliente-Servidor
7. URLs
8. Protocolo HTTP
9. Mensagens HTTP
10. Linguagens Web
11. Front-end vs Back-end
12. Frameworks Web

História da Internet

A evolução da comunicação

por Carlos Diniz

1957 – A Russia lança o satélite Sputnik. Em resposta os EUA criam a Advanced Research Projects Agency (ARPA)

1961 – Desenvolvimento da transmissão de dados por pacote

1962 – J.C.R. Licklider cria o conceito da chamada "Rede Galáctica"

1969 – Nasce a ARPANET.

1969 – Foi estabelecida com sucesso ligação entre os modems de dois computadores distantes.

1969
Tenex

1971 – A Xerox lança o protótipo de um computador pessoal.

1972 – Surge o The Creeper, o primeiro vírus para computador.

1972 – Ray Tomlinson cria a primeira aplicação de envio de email.

1973 – É feita a primeira conexão continental entre os EUA e a Noruega.

1974
Apple I

1977 – Criação do protocolo de transmissão TCP/IP

1978 – A empresa DEC envia 393 emails para funcionários da Arpanet. Nasce o SPAM.

1981 – É lançado o PC da IBM.

1983 – A ARPANET cria a MILNET, uma divisão só para assuntos militares.

1988 – Jarkko Oikarinen cria o canal de chat IRC.

1985
Windows 1.x

1989 – Tim Berners-Lee propõe o sistema World Wide Web

WWW

A história continua...

1990 - Portugal adere à Internet.
1991 - É aberto o uso da rede a entidades comerciais.
1994 - Surge o MOSAIC, primeiro navegador gráfico.
1998 - Ano de lançamento do Google e do NAPSTER.
2004 - Mark Zuckerberg e os seus colegas de quarto lançam o Facebook.
2005 - Chad Hurley, Steve Chen, e Jawed Karim fundam o Youtube.

A World Wide Web (www)

WWW: Espaço de Informação global onde os recursos da Web são identificados por URLs, interligados por hiperlinks ou links (hipertexto ou hiperligações) e acessíveis via Internet.

URL: Uniform Resource Locator

Tim Berners Lee, 1990, “agora”

Robert Cailliau, 1990, “agora”

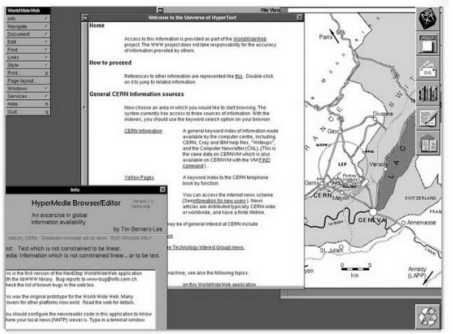
CERN (*Conseil Européen pour la Recherche Nucléaire*), 1990

- Tim Berners Lee e Robert Cailliau
  - Publicaram proposta para gerir informação através de um sistema distribuído de hipertexto (https://www.w3.org/History/1989/proposal.html)

- Para operacionalizar conceitos da proposta, Tim Berners-Lee criou:
  - **WorldWideWeb**, o primeiro navegador Web (https://www.w3.org/People/Berners-Lee/WorldWideWeb.html)
  - **HTML**, *HyperText Markup Language*, linguagem de marcação (https://www.w3schools.com/html/html_intro.asp)
  - **HTTP**, *HyperText Transfer Protocol*, protocolo de comunicação (https://www.w3schools.com/whatis/whatis_http.asp)
  - **CERN httpd**, servidor Web (https://www.w3.org/Daemon/)
  - (4 anos mais tarde) **W3C**, World Wide Web Consortium, para regular e padronizar as tecnologias envolvidas (https://www.w3.org)

DECEMBER 1990

The world's first browser/editor, website and server go live at CERN



By Christmas 1990, Sir Berners-Lee had defined the Web's basic concepts, the html, http and URL, and he had written the first browser/editor and server software. info.cern.ch was the address of the world's first web server, running on a NeXT computer at CERN. The world's first web page address provided information about the World Wide Web project.

https://timeline.web.cern.ch/timeline-header/89

Funcionamento da Web

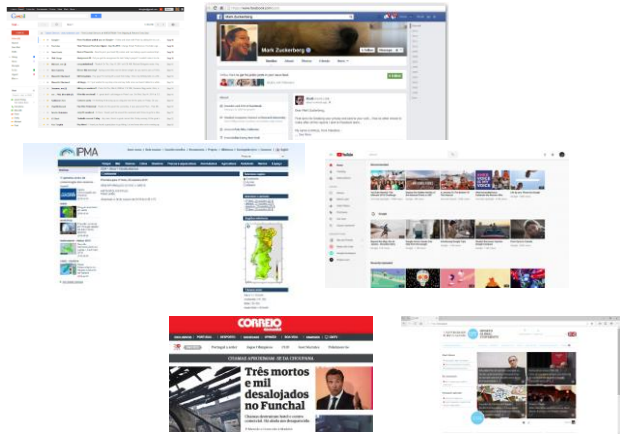
Browser
- Descarregar
- Interpretar
- Renderizar

aceder

Documentos
- Textos
- Imagens
- Vídeos
- Sons
- Hipertexto
- Hiperligações

Permitem “navegar”

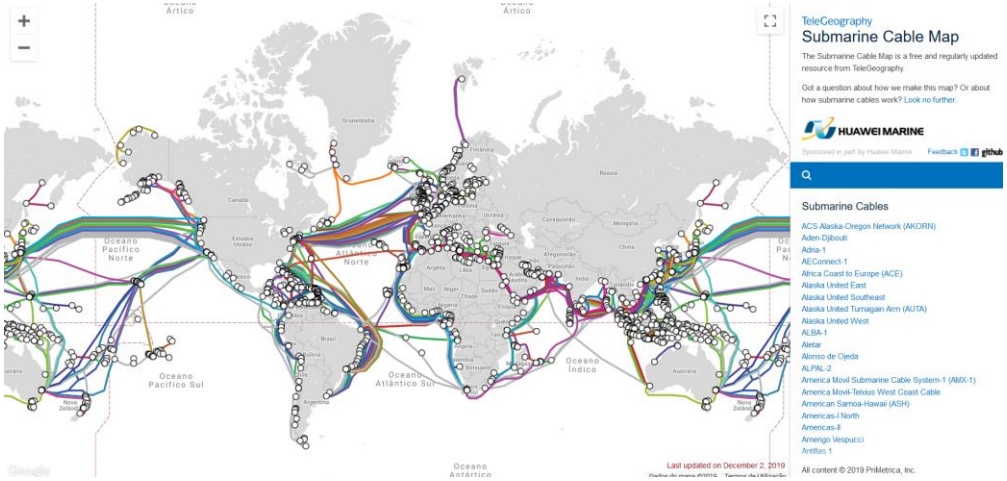
Gerido pelo protocolo HTTP

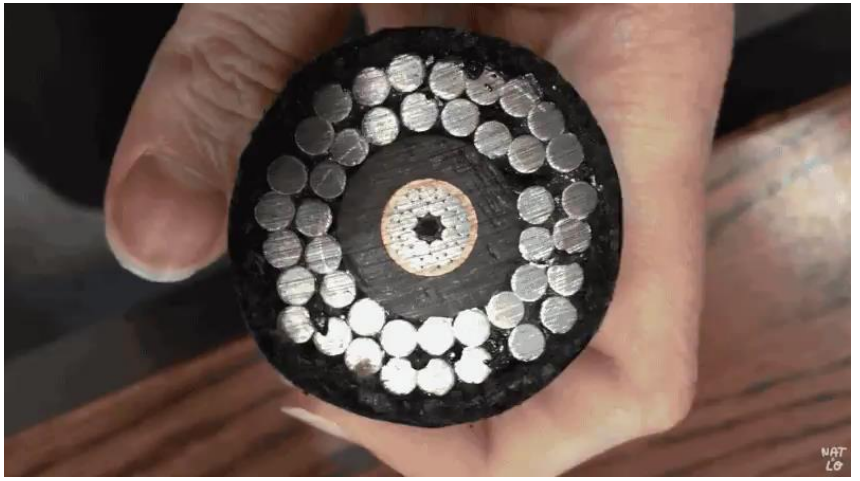


Web 1.0, 2.0, 3.0 e 4.0

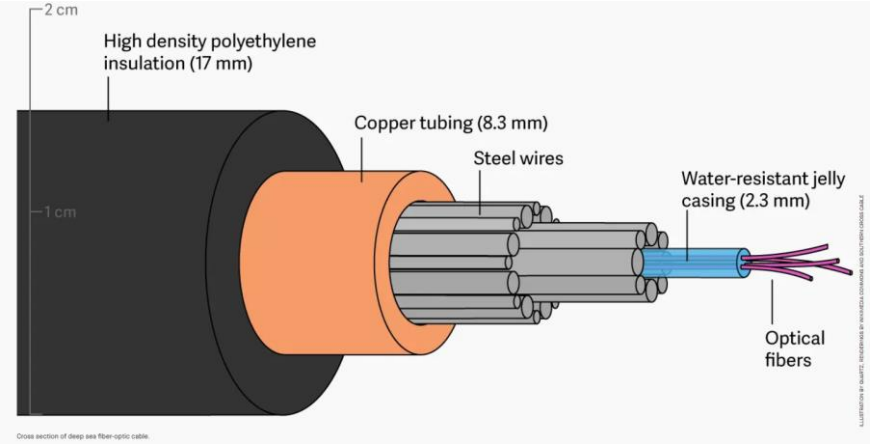
	Web 1.0	Web 2.0	"The Internet of things"	Web 3.0	Web 4.0	Web 5.0
	1994 - 2000	2000 - 2010		2010 – 2020	2020 - 2030	2030 - …
	"The Information Web"	"The Social Web"		"The Semantic Web"	"The intelligent Web"	"The Telepathic Web" "The Symbionet Web"
Foco	Read-only. Conteúdo estático.	Read-write. Conteúdo dinâmico.	Comunicação entre dispositivos	Sistemas de Recomendação	Inteligência Artificial	Implantes cerebrais
Interação	Utilizadores não podem criar nem interagir com websites	Utilizadores podem criar e interagir com websites e também com outros utilizadores	Ligação entre dispositivos "smart" e internet com auxílio da cloud	Ligação inteligente entre pessoas e máquinas	Ligação inteligente entre máquinas	Ligação entre TUDO
Exemplos	Altavista, Geocities, Hotmail, Yahoo! Google	Blogs, Facebook, YouTube, Wikipedia	Ver: https://www.postscapes.com/internet-of-things-examples/	Amazon, YouTube	Computadores como assistentes pessoais, realidade virtual, hologramas, implantes para restaurar visão, …	Capacidade de comunicar com a internet através do pensamento. Pagamentos através de implantes corporais.

A Internet é uma rede global



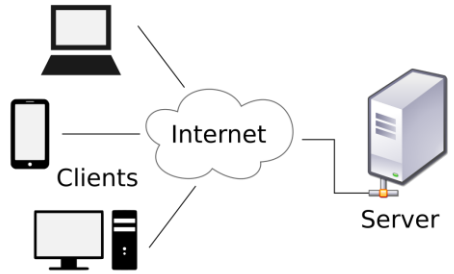


https://www.submarinecablemap.com/

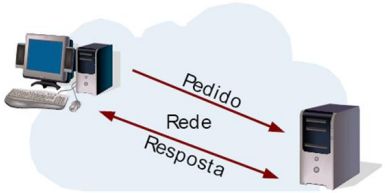


Modelo Cliente-Servidor

- Modelo desenvolvido na Xerox PARC durante os anos 1970
- Descreve relação dos programas numa aplicação

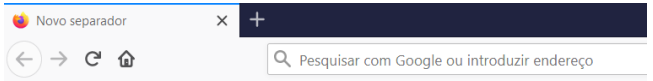


- Funcionamento:



1. Cliente (ex: browser) faz **pedido *I request*** de um recurso (ex: página web) a um servidor
2. Servidor (ex: Apache) recebe o pedido e fornece [dá **resposta *I response***] o recurso desejado

- Clientes e servidor comunicam através da troca de mensagens individuais
  - **Mensagens** atuam sobre recursos no servidor
    - **Recursos** identificados por endereços únicos (URLs)
      - **URL** é o que escrevemos na barra de endereços do browser



URLs

Estrutura

- **Protocolo**: formaliza as regras de comunicação (ex: HTTP, HTTPS*, FTP)
- **Domínio**: endereço do servidor que disponibiliza o recurso solicitado
- **Caminho**: especifica o local onde o recurso se encontra (no sistema de ficheiros) dentro do servidor
- ***Query string*** (opcional): um ou mais pares nome=valor enviados ao servidor para filtrar ou criar um recurso

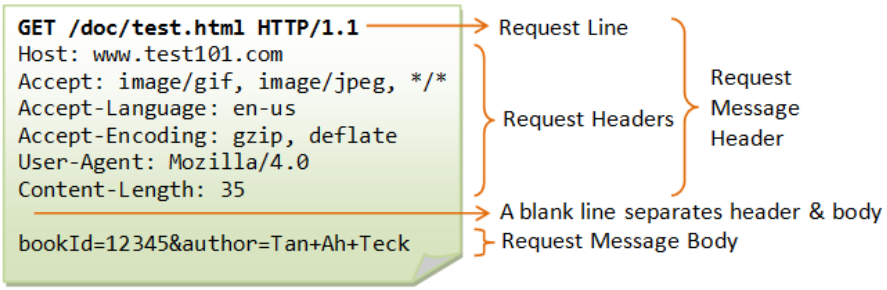
* **HTTPS** (*HTTP Secure*) é uma implementação do protocolo HTTP sobre uma camada adicional de segurança que utiliza o protocolo SSL/TLS (*Secure Sockets Layer* / *Transport Layer Security*), que permite que os dados sejam transmitidos por uma ligação criptografada e se verifique a autenticidade do servidor e cliente através de certificados digitais

Protocolo HTTP

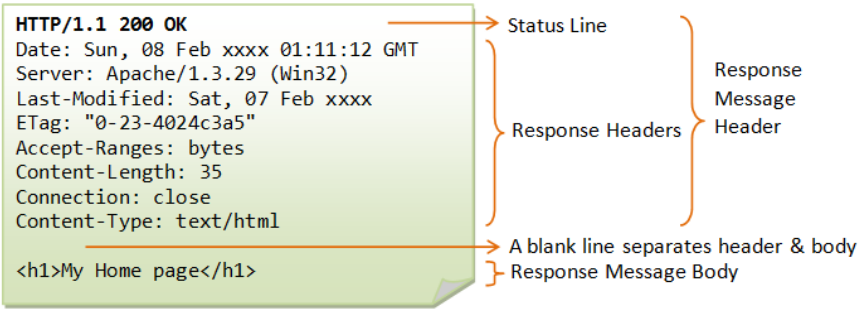
- Protocolo de transferência de hipertexto
- Baseia-se no modelo cliente-servidor
- Permite a troca de mensagens entre os agentes:
  - **Navegadores** (browsers): softwares que pedem os conteúdos em diferentes formatos
  - **Servidores Web**: softwares que funcionam automaticamente e fornecem conteúdos estáticos (ficheiros em disco) e dinâmicos (gerados por programas)
  - ***Proxies***: atuam como intermediários entre o pedido e a resposta, sendo transparentes à comunicação e executam tarefas de encaminhamento de pedidos e respostas, implementação de políticas de acesso e evitam pedidos redundates (*caches*)
- Tem características que o distinguem de outros protocolos. Por exemplo:
  - **Sem conexão** (*connectionless*): a ligação cliente/servidor é refeita a cada pedido. Não permite conexões persistentes para comunicação entre agentes
  - **Sem estado** (*stateless*): não existe estado da interação cliente/servidor. Cada pedido é independente do seu predecente. Uma das formas para simular o estado da interação é através da utilização de cookies.
  - **Independente de *media*** (*media independent*): qualquer recurso pode ser enviado através do protocolo. Os agentes lidam com o tipo de recursos trocados através da leitura de metainformação incluída no protocolo para caracterizar os recursos

Mensagens HTTP

Pedido / Request



Resposta / Response



Linha de pedido – Métodos:

- **get**: solicita recurso. Deve retornar apenas dados
- **post**: submissão de dados para criação de recurso
- **put**: Substitui atuais representações de recurso
- **delete**: remove o recurso do servidor

Cabeçalho:

- **Host**: domínio do servidor
- **Accept**: tipo de conteúdo aceite
- **Accept-Language**: Linguagem esperada
- **Accept-Encoding**: codificação esperada
- **User-Agent**: Descrição do cliente
- **Content-Length**: tamanho (bytes) do pedido

Corpo do pedido

Linha de estado – Códigos:

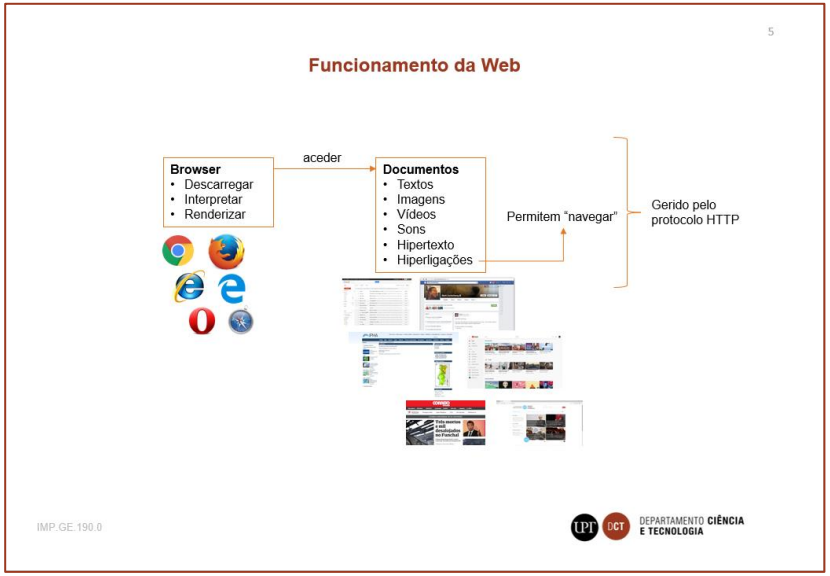
- 1**: respostas informativas
- 2**: sucesso
- 3**: redireccionamento
- 4**: erros do cliente
- 5**: erros do servidor

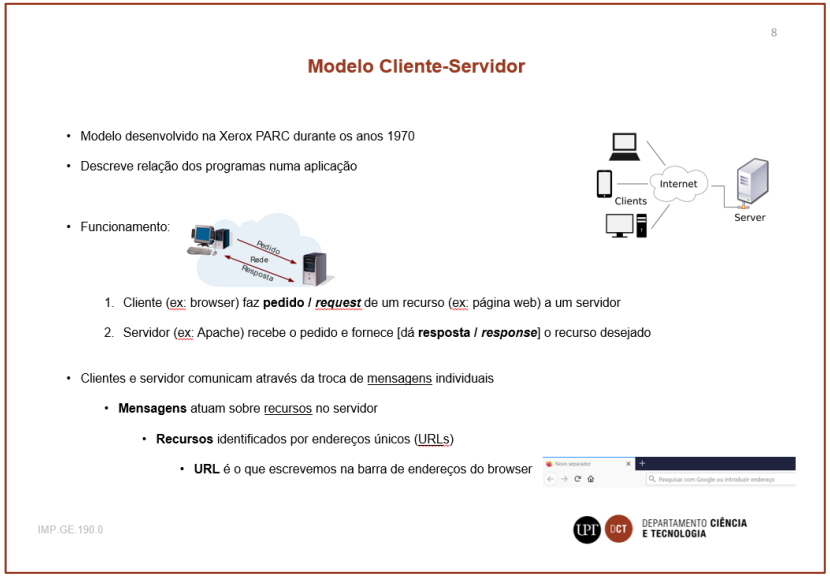
Cabeçalho:

- **Date**: data de início da transferência de dados
- **Server**: software usado no servidor
- **Last-Modified**: última alteração ao recurso
- **ETag**: identificador de versão do recurso
- **Accept-Ranges**: unidade para definir pedidos parciais
- **Content-Length**: tamanho (bytes) da resposta
- **Connection**: controla o que fazer à ligação
- **Content-type**: tipo de conteúdo da resposta

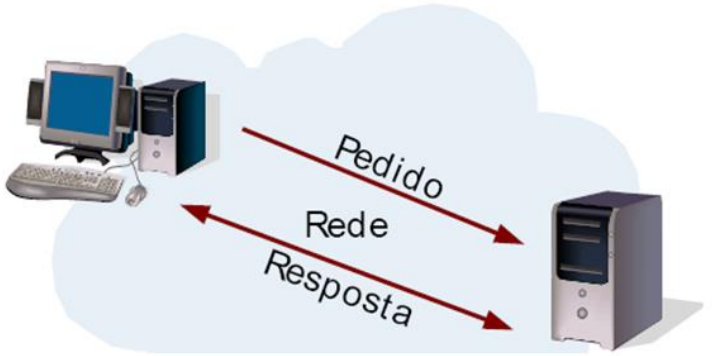
Corpo da resposta

Linguagens Web





Frontend



Backend

DEPARTAMENTO CIÊNCIA E TECNOLOGIA

Front-end vs Back-end

Frontend

- Do lado do cliente
- Parte da aplicação que interage diretamente com o utilizador
- Parte estética do site
- Linguagens:
  - **HTML**: conteúdo
  - **CSS**: aparência
  - **JavaScript**: interatividade

Full Stack

- De ambos os lados
- "Pilha de software que responde de uma forma completa e integrada às necessidades do desenvolvimento frontend e backend"
  (in *Introdução ao Desenvolvimento Moderno para a Web*)
- Exemplo: **MEAN**
  - (**M**ongoDB, **E**xpress, **A**ngular, **N**odeJS)
  - Popular porque permite usar apenas JS tanto do lado do cliente como do servidor

Backend

- Do lado do servidor
- Parte da aplicação que interage com serviços
  - **API RESTful** *
- …e bases de dados:
  - **Relacionais**
  - **NoSQL**

* API RESTful
API: Application Programming Interfaces
REST: Representational State Transfer

Frameworks Web

Framework Web: conjunto padronizado de conceitos e práticas para lidar com os problemas do desenvolvimento Web e servir de referência à solução de novos problemas de natureza similar.

- **Frameworks Frontend**
  - Foco na camada de apresentação: apresentação gráfica e interação (ex: responsividade)
  - Bibliotecas de código eficientes
  - Combinação de linguagens de marcação, estilização e scripting
  - Componentes das **frameworks web responsivas** (Ex: **Bootstrap**, **Foundation**, **Kube**, **Semantic-UI**, **Skeleton**, **Materialize**, **Pure**)
    - **Folhas de estilo CSS**: posicionamento responsivo os elementos, estilização de HTML, modelação de componentes gráficos
    - **Plugins JS**: implementação de componentes de interação avançados.
- **Frameworks Backend**
  - Foco nas camadas lógica e de dados: codificação da lógica de negócio e acesso a dados
  - Bibliotecas para aceder a bases de dados, mapear dados e objetos, gerir sessões, …
  - **Node.js**: interpretador de código JS para facilitar criação de aplicações de alta escalabilidade.
    - Framework Node.js – **Express** – com conjunto robusto de recursos (views, roteamento, execução)
  - **MongoDB**: sistema de bases de dados não relacional
    - Alta capacidade de disponibilidade, escalabilidade e flexibilidade
    - Permite armazenamento de documentos **JSON** (JavaScript Object Notation) sem uma estrutura associada

UNIVERSIDADE
PORTUCALENSE

Do conhecimento à prática.