

Aplicações Web



Simão Paredes
sparedes@isec.pt

Sistema Hipermédia

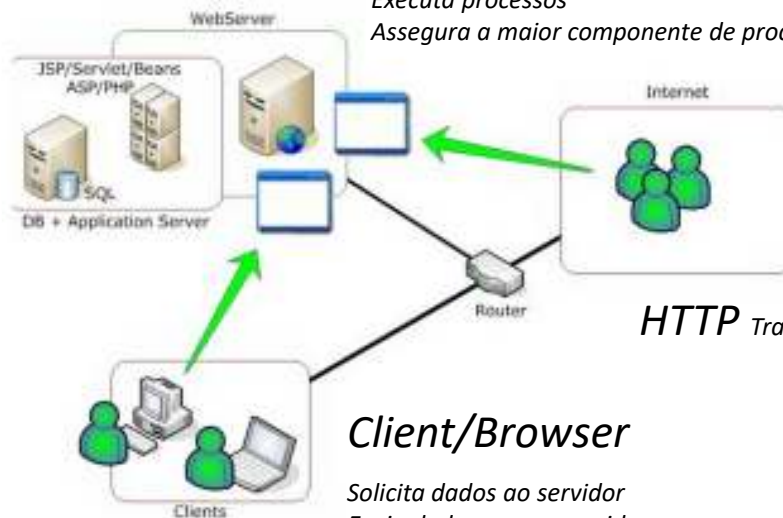
- Possui uma **organização não linear** da informação e pode ser descrito da seguinte forma:
 - Um conjunto de nós onde reside um ou mais tipos de dados:
 - Texto; áudio; imagem; vídeo; ...
 - Um conjunto de ligações (links) entre nós que se podem percorrer de forma não linear;
 - Em 1989, Tim Berners-Lee criou um sistema de gestão de informação que deu origem à **world wide web (www)**
 - Maior sistema hipermédia



<https://twitter.com/infobaeAmerica/status/1105464985956818949>

Web Server

*Envia dados ao cliente
Acede a bases de dados
Executa processos
Assegura a maior componente de processamento*



HTTP Transferência de dados

Client/Browser

*Solicita dados ao servidor
Envia dados para o servidor
Solicita ao servidor uma execução específica
Executa um processo específico*

Internet

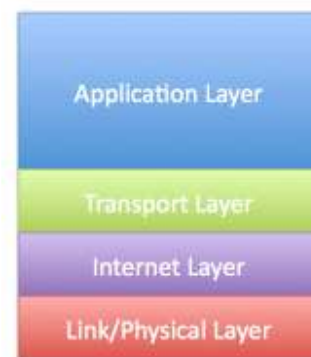
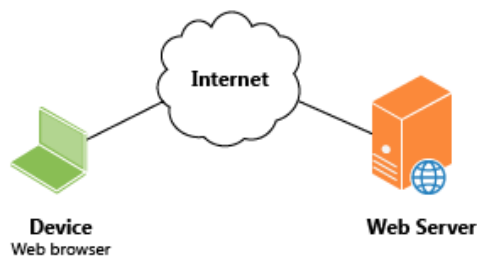
Internet

- Infra-estrutura de rede + protocolo de comunicação
 - Suporta vários serviços:
 - **world wide web** [*HTTP (hypertext transfer protocol)*]
 - O HTTP é um protocolo de comunicação, o qual é um elemento nuclear da *www*
 - e-mail (SMTP, POP3, IMAP,...)
 - VoIP
 - file sharing (ftp) ...



<http://pplware.sapo.pt/informacao/cisco-preve-crescimento-no-trafego-global-de-dados-moveis/>

Internet Layers



http://www.cellbiol.com/bioinformatics_web_development/

■ Rede partilhada

- Otimiza a utilização da rede, várias mensagens para vários destinatários em simultâneo
 - Múltiplos emissores / Múltiplos recetores
- Mensagens são divididas em pacotes
 - Pacotes enviados separadamente.
 - Cada pacote tem que conter dados que permitam identificar os respetivos emissor e recetor (endereços **IP**)
 - Tem que existir forma de assegurar a integridade da informação (**TCP**)

Internet layer



Hospedeiros (*hosts*)

Emissores/Recetores de pacotes

Routers

Interligação entre as várias redes que constituem a internet

Cada *host* é identificado pelo seu endereço IP:

Conjunto de campos numéricos que identifica o computador e a rede a que se encontra ligado

193.137.78.242	IPV4 (32 bits)
0:0:0:0:ffff:c189:4ef2	IPV6 (128 bits)

A atribuição de IP's é um processo controlado hierarquicamente por:

Internet Assigned Numbers Authority (IANA) <https://www.iana.org/>

RIPE Network Coordination Centre (Europa) www.ripe.net

Local Internet Registries <https://www.ripe.net/membership/indices/PT.html>

Internet layer

■ Domain name servers (DNS)

■ Serviço de mapeamento dos endereços IP em domínios :

- Nomes mais fáceis de memorizar e que permitem otimizar a interação com o utilizador

■ <http://193.137.78.36/>

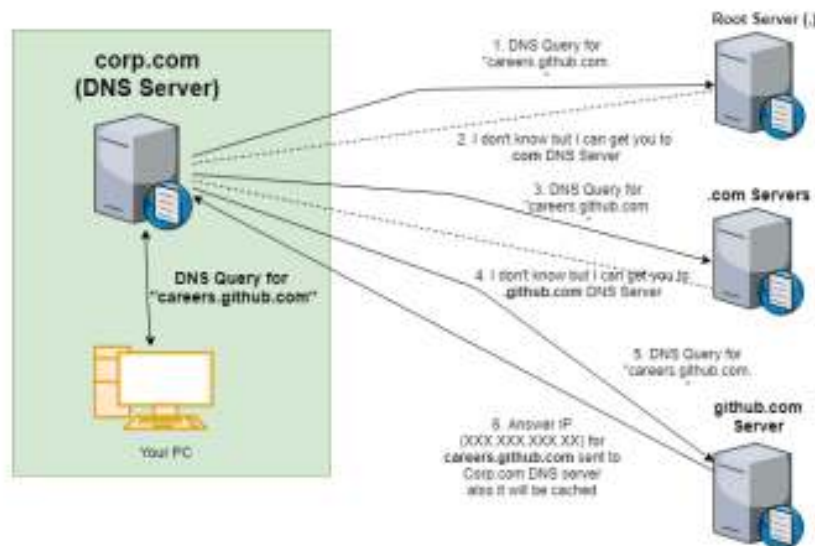
■ www.isec.pt



<https://medium.com/@openmohan/dns-basics-and-building-simple-dns-server-in-go-6cb8e1cf461>

Internet layer

■ Domain name servers (DNS)



<https://medium.com/@openmohan/dns-basics-and-building-simple-dns-server-in-go-6cb8e1cf461>



→ **Internet layer:** protocolo de endereçamento não reconhece:
qualquer relação entre pacotes
qualquer informação ao nível da aplicação

como tal:

não assegura que todos os pacotes que constituem a mensagem são entregues com sucesso
não assegura que os pacotes são recebidos pela ordem correta

→ **Transport layer:** assegura a entrega fiável dos pacotes
todos os pacotes foram entregues pela ordem correta (integridade da informação)
Acknowledgments

Outros protocolos de transporte:
UDP (User Datagram Protocol)
RTP (Real-Time Transport Protocol)

world wide web / HTTP

Hyper Text Transfer Protocol

Application Layer - HTTP

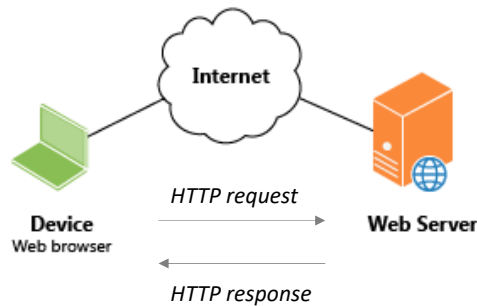
Hyper Text Transfer Protocol (HTTP):

Protocolo cliente – servidor

Programa cliente (web browser) interage com um servidor web (web server)

Cliente envia um pedido (*HTTP request*) de um determinado recurso a um servidor

Sempre que recebe um pedido o servidor web localiza ou constrói o recurso solicitado e envia como resposta ao cliente (*HTTP response*)



Application Layer - HTTP

■ Universal Resource Locator (URL)

`scheme: // host /path/.../[;url-params][?query-string][#anchor]`

■ Elementos Obrigatórios

■ *scheme*

■ Protocolo: http

■ *host*

■ Endereço IP / domínio (DNS)

■ *path*

■ O caminho da raiz do servidor para a directoria que contem o documento desejado.

`http://www.isec.pt/alunos/guia-estudante/index.html`

■ Elementos Opcionais

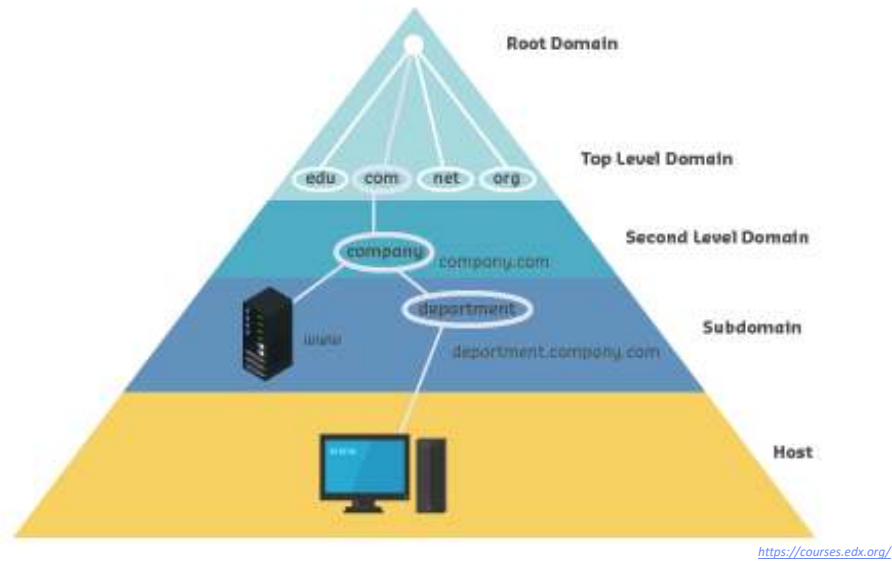
■ *url-params; query-string; anchor*: utilizados para passar informação através do URL (ex: identificadores de sessão, dados de formulários,...)

■ ...

<http://www.google.com/support/webmasters/bin/answer.py?answer=1235687>

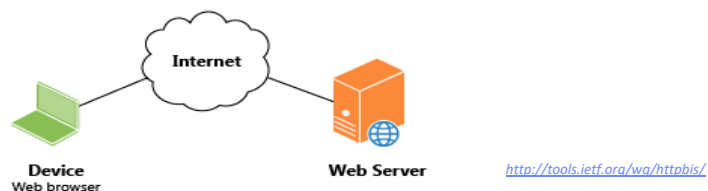
Application Layer - HTTP

- **Universal Resource Locator (URL)**
 - **host**
 - Endereço IP / domínio (DNS)
 - <http://www.isec.pt>



Application Layer - HTTP

- **Pedido HTTP (request)**
 - Pode ser uma página HTML, um ficheiro, uma execução de um programa, ...
 - Implica um mecanismo de identificação de recursos
 - **URL (Uniform Resource Locator)**
 - É efetuado pelo web browser
 - HTTP **não permite** que a ligação inicial se estabeleça por iniciativa do servidor



- **https (hyper text transfer protocol secure)**
 - Utiliza certificados SSL para encriptar a comunicação cliente/servidor.

Evolução

world wide web

Evolução (www)

web 1.0

"content creators were few in Web 1.0 with the vast majority of users simply acting as consumers of content."

web 2.0

"... In the Web 2.0 era, the customers are **the creators**. The creators build web pages, upload audio, video and images, and tag multimedia files providing other users with useful information about their creations. This material is known as **"user-generated content"**."

http://blogs.voices.com/thebiz/2006/09/web_20_definition.html

web 3.0

Knowledge discovery
Semântica
Relevância dos Dados
Inteligência Artificial
Personalização



- web 3.0 & IOT (*internet of things*)



<https://www.expressvpn.com/blog/what-is-the-internet-of-things-iot/>

Principais Desafios

world wide web

Principais Desafios

Modern web applications have higher user expectations and greater demands than ever before. Today's web apps are **expected to be available 24/7 from anywhere in the world**, and usable from virtually **any device or screen size**. Web applications **must be secure, flexible, and scalable** to meet spikes in demand. Increasingly, complex scenarios should be handled by **rich user experiences** built on the client **using JavaScript**, and communicating efficiently **through web APIs**

...

<https://docs.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/modern-web-applications-characteristics>

Principais Desafios

■ Segurança

- <https://www.websitehostingrating.com/cybersecurity-statistics-facts/>



■ Principais vulnerabilidade nas aplicações web:

- <https://www.toptal.com/security/10-most-common-web-security-vulnerabilities>
- <https://www.commonplaces.com/blog/6-common-website-security-vulnerabilities/>

Principais Desafios

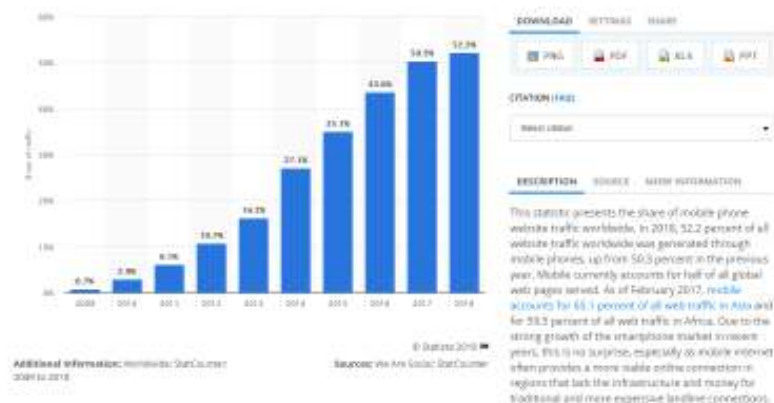
■ web mobile

- Deve ser considerada a diversidade de dispositivos para acesso à web

*The widespread deployment of Web-enabled mobile devices (such as phones) make them a **target of choice** for content creators. Understanding **their strengths and their limitations**, and using technologies that fit these conditions are key to create success mobile-friendly Web content.* <https://www.w3.org/standards/webdesign/mobileweb>



Percentage of all global web pages served to mobile phones from 2009 to 2018



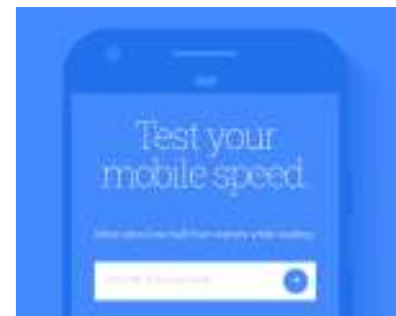
<https://www.statista.com/statistics/241462/global-mobile-phone-website-traffic-share/>

Principais Desafios

■ Desempenho

- Single Page Application (SPA)
 - Otimizar a transferência de dados (AJAX, ...)
- Reduzir o número de pedidos (*requests*) ao servidor
- Limitar a dimensão dos ficheiros (imagens, vídeos, ...)

• <https://techbeacon.com/app-dev-testing/23-front-end-performance-rules-web-applications>



<https://testmysite.withgoogle.com/intl/en-gb>

■ Visibilidade

- Search Engine Optimization (SEO)
 - Título
 - keywords
 - Descrição do documento

• <https://support.google.com/webmasters/answer/7451184?hl=en>



<http://www.misowebdesign.com/>

■ Portabilidade

- Os *browsers* tem diferentes suportes às diversas tecnologias (tendência para uma progressiva uniformização).
- *progressive enhancement* deve condicionar a estratégia para lidar com uma capacidade variável dos diferentes *browsers*:



Browser Support

The numbers in the table specify the first browser version that fully supports the property.
Numbers followed by -webkit- specify the first version that worked with a prefix.

Property	Chrome	Firefox	Internet Explorer	Opera	Safari
align-content	21.0	11.0	20.0	9.0	12.1
				7.0 -webkit-	

http://www.w3schools.com/cssref/css3_pr_align-content.asp

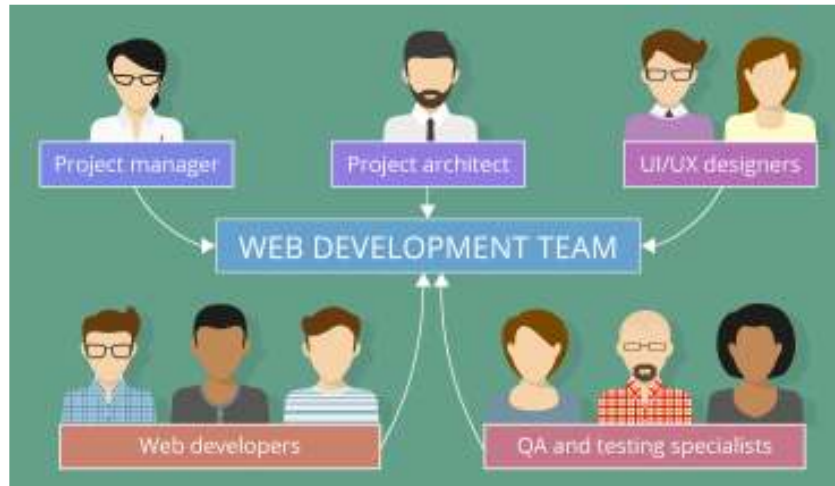
■ Acessibilidade

- Reduzir as barreiras ao acesso à informação por parte dos utilizadores com necessidades especiais (invisuais, deficientes motores, ...)
- *Web Content Accessibility Guidelines* (WCAG) <http://www.w3.org/WAI>

web development

- O desenvolvimento de uma aplicação web com alguma dimensão necessita de uma equipa composta por especialistas em diferentes áreas:

■ <https://www.scnsoft.com/blog/how-to-assemble-a-good-web-development-team>



<https://coderseye.com/programming-web-developer-salaries>

Front-end development

world wide web

Front End Development: Tecnologias

- O browser, independentemente da tecnologia aplicada para criação da aplicação, consegue interpretar apenas HTML, CSS e JavaScript



= HTML, CSS, and
JavaScript

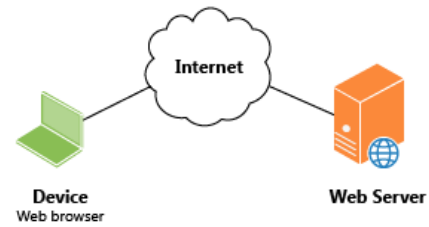
Front End Development: Tecnologias

- Diretamente interpretadas pelo browser (tecnologias base)

- HTML5
- CSS3
- JavaScript **

- Frameworks / Pré-processadores

- Bootstrap
- Foundation, ... <https://geekflare.com/best-css-frameworks/>
- SASS (pré-processador CSS)
- jQuery
- AJAX,
- React
- Angular
- Vue, ... <https://hackr.io/blog/best-javascript-frameworks>



Client Side Web Technologies



Front End Development

- Apesar de por vezes se interceptarem é possível perspetivar o *front-end development* em diferentes camadas:

- Estrutura
- Apresentação
- Comportamento



■ Content Management System (CMS)

- Ferramenta de alto nível que permite criar, editar e gerir o conteúdo de um site.
 - Dispensa o conhecimento das tecnologias base



Natureza do Conteúdo

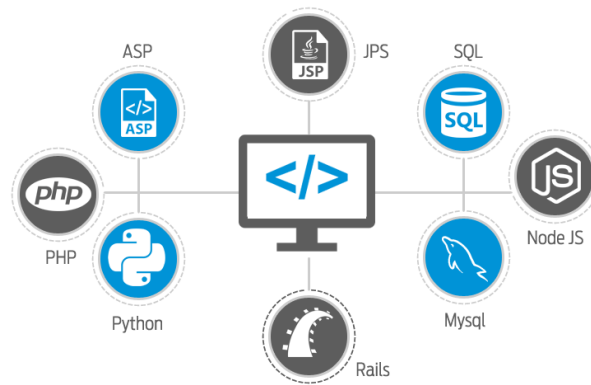
Natureza do Conteúdo

■ Estático

- O conteúdo não pode ser actualizado/alterado sem que haja uma alteração no código-fonte do site.

■ Dinâmico

- Tendo como suporte linguagens de script/bases de dados (*server side*) os conteúdos podem ser alterados sem necessidade de alterações no código-fonte.

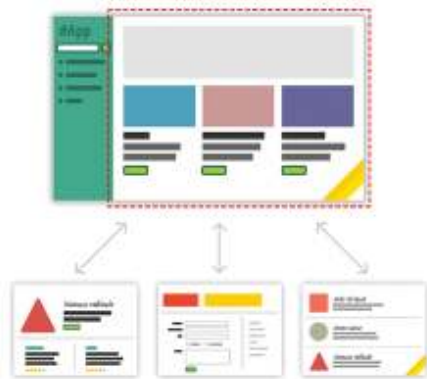


<https://www.addonsolutions.com/blog/selection-dilemma-static-vs-dynamic-web-development.html>

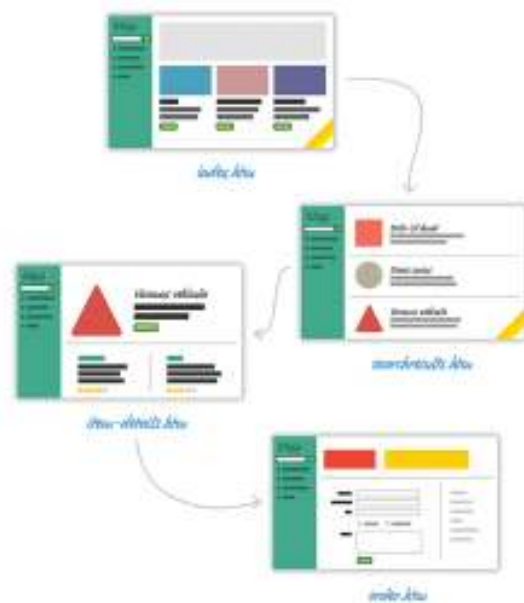
- Pode ser gerado conteúdo de forma dinâmica tendo por base exclusivamente linguagens de script no cliente (*client side scripting languages*)

Single Page Application vs. Multi Page Application

Single Page Application



Multi Page Application



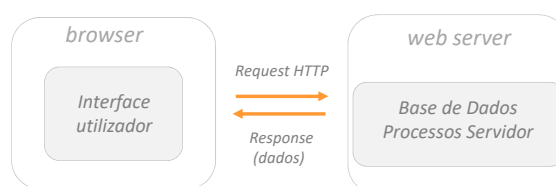
Chinnathambi, K., Learning React, ISBN: 970-0-13-484355-1, Addison-Wesley, 2018

Multi Page web Applications (MPA)

- Estrutura clássica - múltiplos ficheiros *.html
 - Acesso a um novo ficheiro implica um novo *request* e espera pela respetiva *response*

Interação Clássica:

1. Pedido HTTP
2. Resposta servidor com o pedido processado (tempo de espera)
3. Novo pedido
4. Nova resposta servidor (tempo de espera)



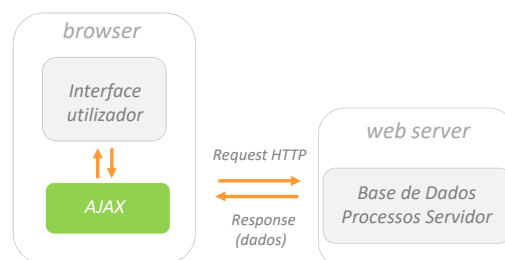
Multi Page Application (MPA)

- Forma tradicional
 - cada alteração exige um novo request ao servidor para envio de nova página
- Navegação sólida
- Facilita a Search Engine Optimization (SEO).
 - Por exemplo ao definir keywords apropriadas para as diferentes páginas
- Navegação menos fluída (maior tempo de espera/latência) do que uma SPA.
- O desenvolvimento do frontend e backend estão fortemente acoplados o que condiciona a evolução da própria aplicação.

Single Page Application (SPA)

Interação:

1. Início da sessão
 2. Servidor envia aplicação cliente (módulo AJAX) que permanece activo toda a sessão
 3. Script (cliente) gere informação introduzida pelo utilizador (links, forms, ...) e determina se é necessária uma chamada assíncrona ao servidor ou se é um processo local (cliente).
- Só são solicitados os dados necessários à actualização pretendida



Single Page Application (SPA)

- Um único ficheiro *.html que vai sendo atualizado de acordo com a interação do utilizador
 - A aplicação é transferida para o cliente sendo que a gestão dos conteúdos a disponibilizar é maioritariamente feita no lado do cliente (script, ...):
 - inicialmente é transferido para o cliente o ficheiro *.html assim como todo o *.css e *.js necessários ao funcionamento do site;
 - não existe necessidade de efetuar o *reload* de uma nova página;
 - a atualização do conteúdo é feita de forma assíncrona o que torna a interação muito mais fluída;
 - são solicitados apenas os dados necessários para a atualização de conteúdo.
 - A grande desvantagem prende-se com Search Engine Optimization uma vez que o conteúdo é gerado dinamicamente diretamente no cliente
 - Gmail, Google Maps, Facebook, GitHub, ...

SPA vs. MPA

- Vantagens (SPA):
 - Melhor experiência do utilizador
 - Menos *refresh* da página, maior fluidez
 - Permite uma experiência semelhante a uma aplicação móvel (nativa)
 - Performance
 - Melhora a performance do sistema tornando o acesso a conteúdos muito mais rápido
 - Facilidade de manutenção
- Desvantagens (SPA):
 - Novo Framework:
 - Curva de aprendizagem de um novo framework (AngularJS, React).
 - Execução de scripts *JavaScript*:
 - Caso a execução de scripts esteja bloqueada no browser implica a interrupção do funcionamento de todo o site
 - *Search Engine Optimization*:
 - A indexação das páginas pelos motores de busca pode ser dificultada

Comunicação entre Aplicações/Plataformas (web services)

web services

- Interface entre diferentes aplicações / plataformas
 - permite a comunicação entre aplicações desenvolvidas em diversas tecnologias
 - é a base tecnológica que permite que uma aplicação possa consumir dados de outras aplicações
 - web service pode ser definido como:

“Web services provide a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks.”

<https://www.w3.org/TR/ws-arch/>



<https://codenuclear.com/web-services-with-jax-rs-example/>

Restful web services

- Baseados na arquitetura REST (**R**epresentational **s**tate **t**ransfer)
- Nem todos os web services são baseados em REST (ex:SOAP)
- Muito utilizados na www devido à simplicidade na sua implementação
- Cada comunicação é totalmente independente da anterior e da seguinte
 - Todos os dados são enviados de uma vez
 - Todos os dados são recebidos de uma vez
 - *Stateless*

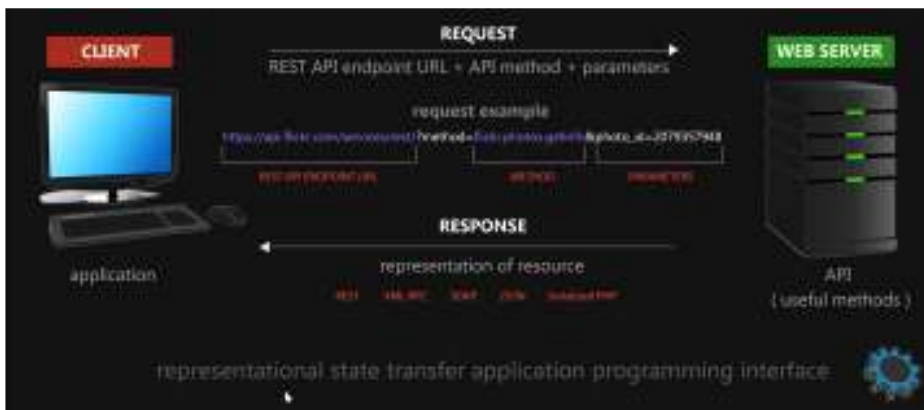


https://www.youtube.com/watch?v=LooL6_chvN4

Restful web services



Numa interação cliente - servidor é solicitado ao servidor um determinado recurso (*.jpg; *.html)



Num *REST Request* podem ser invocados um conjunto de métodos/funções disponibilizados pela REST API aos quais são passados parâmetros de forma a obter os dados desejados (*REST Response*).

Human Computer Interaction (HCI)

HCI

*Human-computer interaction (HCI) is a discipline concerned with the **design, evaluation and implementation of interactive computing systems for human use**, and with the study of the major phenomena surrounding them.*

(Saul Greenberg)



- Alguns dos aspetos a considerar:
 - Consistência Gráfica
 - Estrutura de Informação
 - Navegação
 - Conteúdos
 - Cor
 - ...