

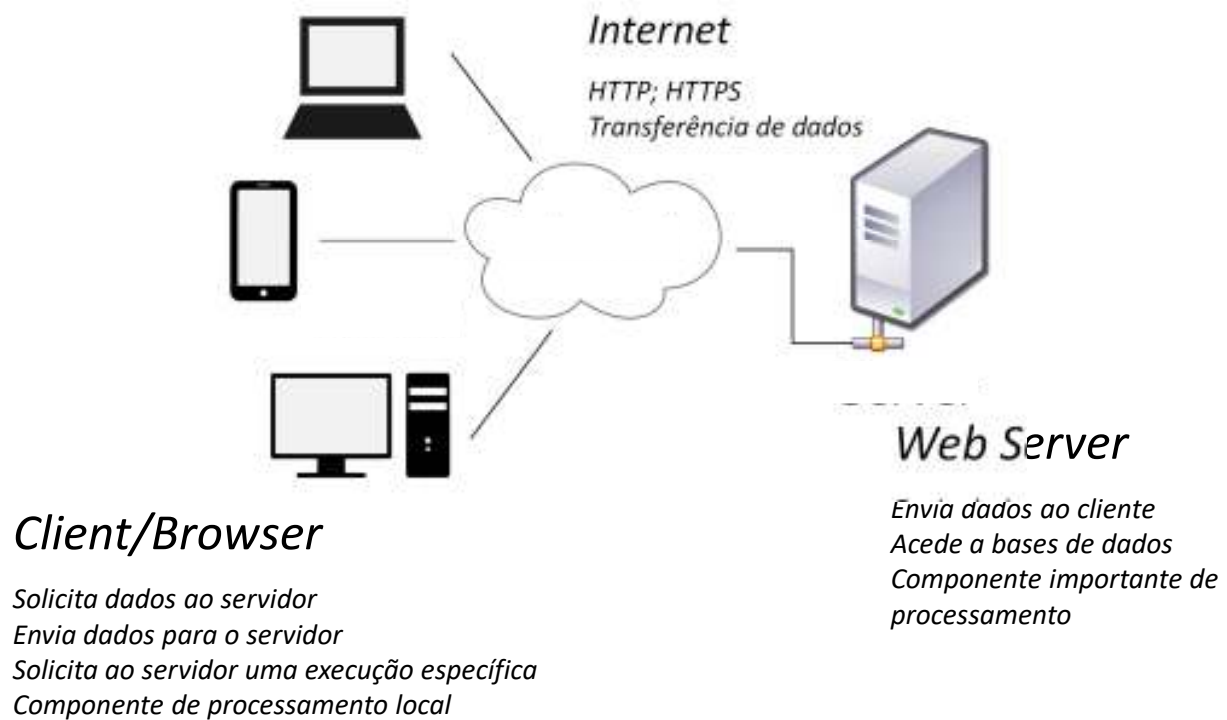
Aplicações Web

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>



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/>

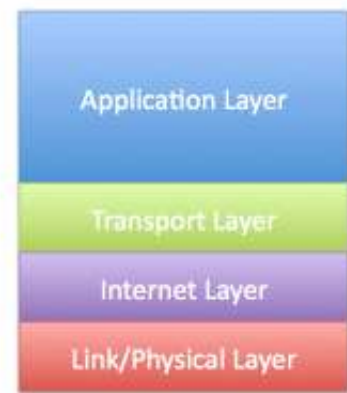
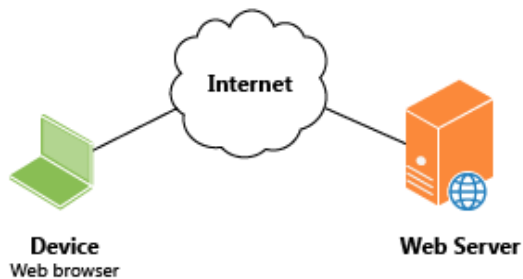
Internet

- Comunicação
 - Interligação
 - cabo
 - wireless
- +
- Protocolos
 - endereçamento
 - transporte
 - aplicação



<https://www.vox.com/2014/6/16/18076282/the-internet>

Internet Layers



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

TCP/IP (Internet layer + transport layer)

■ 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)
0:0:0:0:0: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://193.137.78.72/>

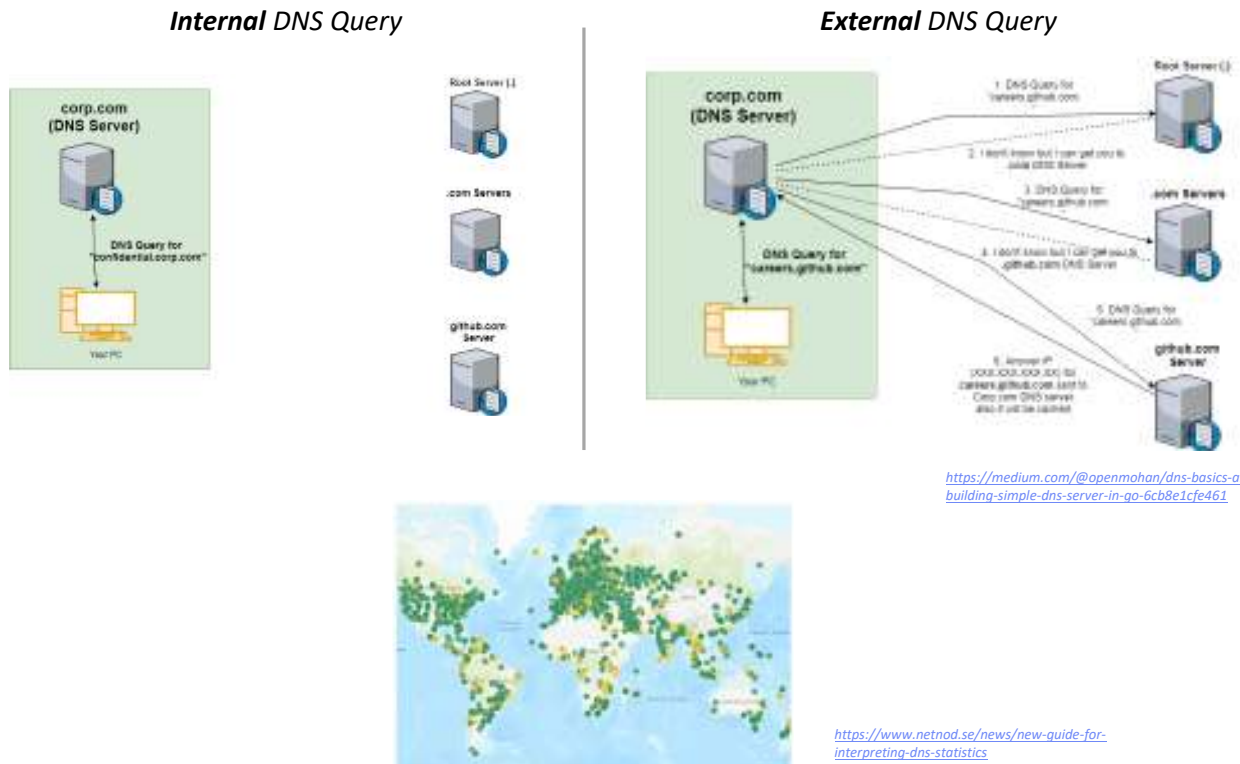
■ www.isec.pt



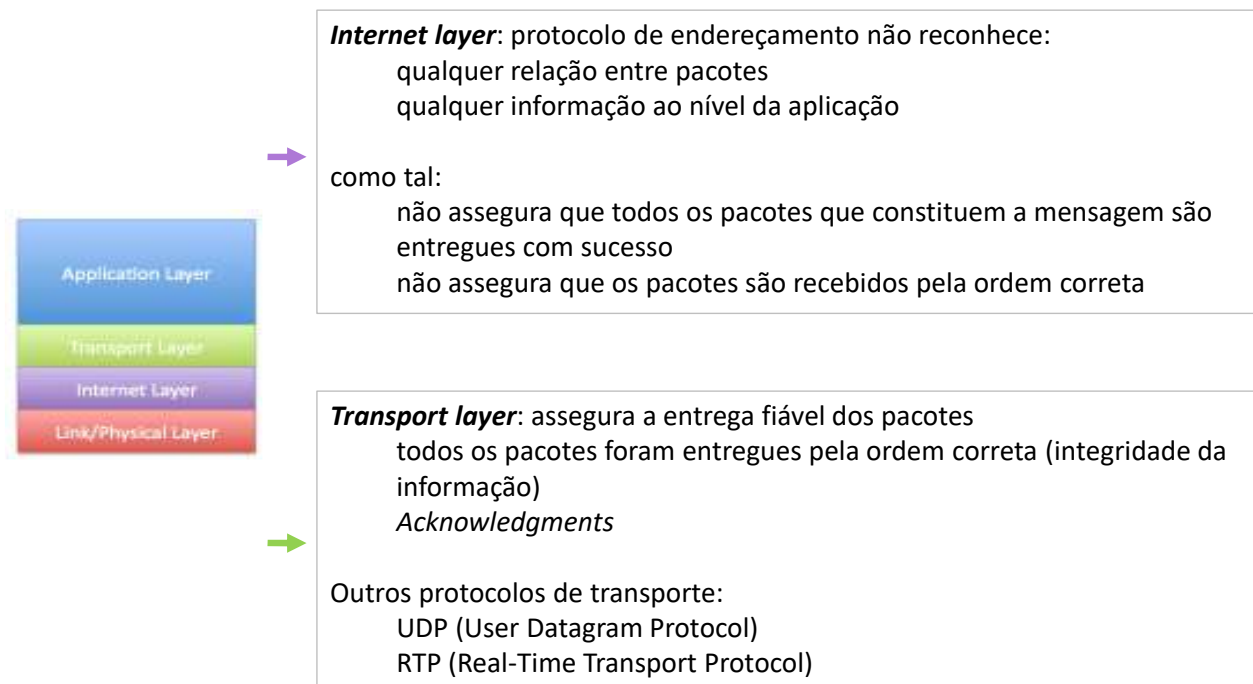
<https://medium.com/@openmohan/dns-basics-and-building-simple-dns-server-in-go-6cb8e1cfe461>

Internet layer

■ Domain name servers (DNS)



Internet Layer / Transport Layer



world wide web / HTTP

Hyper Text Transfer Protocol

Application Layer - HTTP

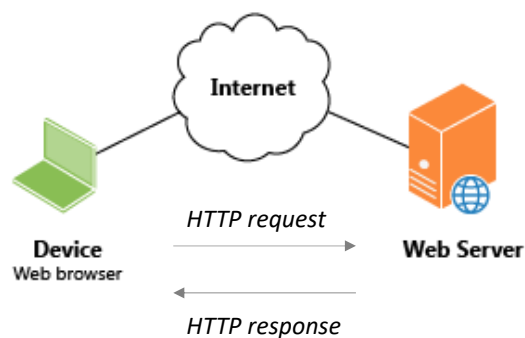
Hyper Text Transfer Protocol (HTTP):

Protocolo cliente – servidor

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

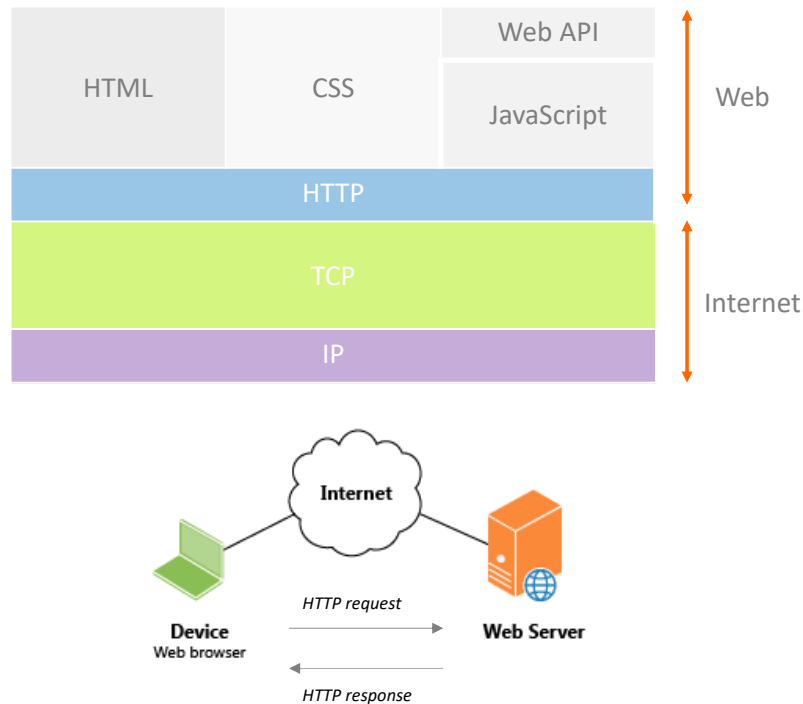
Cliente envia um pedido (*HTTP request*) de um determinado recurso a um servidor

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



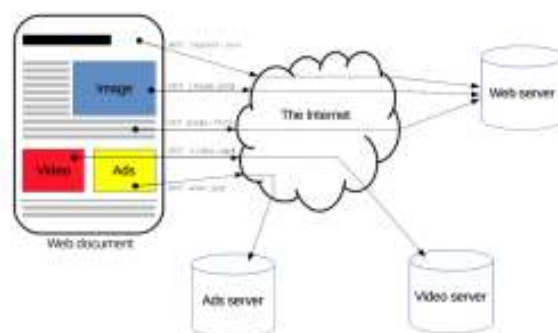
Application Layer - HTTP

- Protocolo de comunicação que permite solicitar/receber os diferentes elementos a ser interpretados pelo browser



Application Layer - HTTP

- Os pedidos HTTP:
 - São sempre iniciados pelo *browser*, nunca são iniciados do lado do servidor
 - Um documento completo é reconstruído a partir dos diferentes sub-documentos obtidos:
 - texto, descrição do layout, imagens, vídeos, scripts.

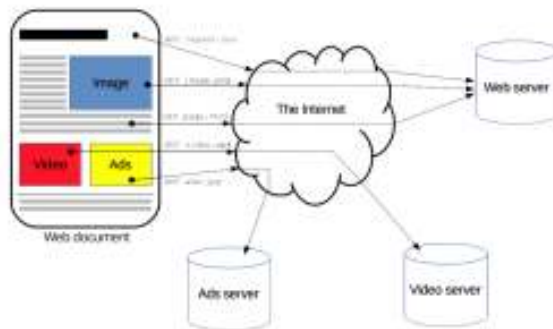


<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview>

- Cliente/Servidor comunicam com base em mensagens individuais
 - HTTP Request / HTTP Response
 - HTTP Request / HTTP Response
 - ...

Application Layer - HTTP

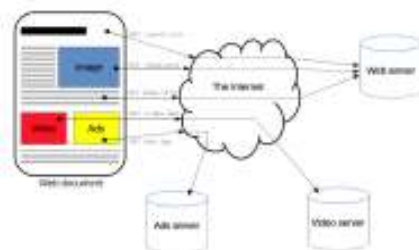
- Cliente
 - Cliente /utilizador
 - *browser*
 - *user agent* (*crawler* usado pelos motores de pesquisa para indexação automática)
 - Para efetuar o *display* de uma aplicação web, o *browser*:
 - Envia um *HTTP request* para obter o documento HTML
 - Analisa a sintaxe do ficheiro HTML para detetar scripts, CSS e conteúdos (imagens, vídeo)



<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview>

Application Layer - HTTP

- Cliente
 - O *browser* interpreta todos esses elementos e mostra o produto final.
 - Existem *scripts* diretamente executados no *browser* que em fases subsequentes e de acordo com a interação do utilizador, necessitam de mais recursos.
 - É feito o *download* destes recursos e a aplicação/página global é atualizada.



- Um ficheiro HTML possui vários links, os quais dependendo da interação com o utilizador, permitem a navegação pela web.
 - O browser traduz esses endereços em HTTP requests e depois interpreta as respetivas HTTP responses para efetuar o display ao utilizador.

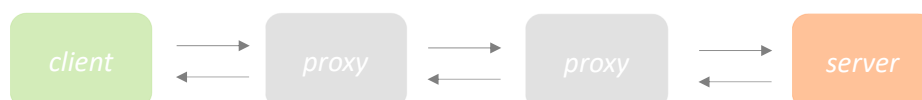
Application Layer - HTTP

■ Servidor

- Disponibiliza o recurso solicitado pelo cliente e virtualmente pode ser perspectivado como uma máquina, que na realidade pode ser:
 - apenas um servidor
 - conjunto de servidores com balanceamento de carga
 - um programa que gere o acesso a outros servidores para gerar parte ou a totalidade dos elementos solicitados:
 - servidor de base de dados,
 - mashups,
 -
- Um servidor não implica necessariamente uma máquina dedicada, mas vários servidores podem estar hospedados na mesma máquina.

Application Layer - HTTP

- Entre o browser e o servidor existem vários intermediários a operar nas várias camadas para transmitir as mensagens HTTP.
 - Nas camadas inferiores são transparentes para a camada da aplicação HTTP
 - Na camada de aplicação são designados por proxies e podem desempenhar várias funções:
 - *cache*
 - Filtragem (antivírus)
 - Balanceamento de carga (permitir que vários servidores possam responder a diferentes requests)
 - Autenticação/Autorização (controlo do acesso aos recursos)
 - Registo de informação (armazenamento de informações de histórico)



Application Layer - HTTP

■ Características do protocolo HTTP

■ Simples

- projetado para ser simples e legível.

■ Extensível

- os cabeçalhos HTTP permitem a extensão de funcionalidades (negociação de conteúdo).

```
GET / HTTP/1.1
Host: developer.mozilla.org
Accept-Language: fr
```

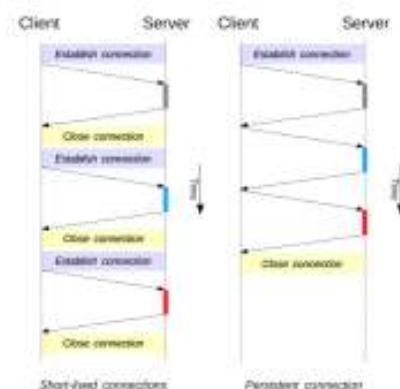
■ Não tem estado, mas tem sessões

- Como tal, não existe uma relação entre dois *requests* diferentes entre cliente e servidor o que provoca uma limitação óbvia para a continuidade da interação com o utilizador
 - pode ser contornada através de cookies que são adicionados ao fluxo do HTTP, permitindo dessa forma que a criação de sessão em cada HTTP Request possua o mesmo contexto

Application Layer - HTTP

■ HTTP

■ Ligação persistente



https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Connection_management_in_HTTP_1.x

- Short-lived connections (ligação não persistente) tem dois grandes problemas:
 - O tempo necessário para estabelecer uma nova ligação é significativo
 - Desempenho da ligação TCP subjacente melhora com a frequência de utilização da ligação.
- Ligação persistente
 - permanece aberta por um período e pode ser reutilizada por vários pedidos, evitando assim a necessidade de um novo handshake TCP.
 - Ligações não utilizadas são fechadas ao fim de um determinado tempo que pode ser parametrizado.
 - O HTTP/1.1, utiliza como default ligações persistentes.

Application Layer - HTTP

Fluxo

- Sequência de operações para se estabelecer a comunicação cliente servidor:
 - Abrir uma ligação TCP de forma a enviar 1 ou mais HTTP request.
 - Request HTTP
 - na versão HTTP/2.0 as mensagens são enquadradas dentro de frames não ficando diretamente visíveis

```
GET / HTTP/1.1
Host: developer.mozilla.org
Accept-Language: fr
```

- HTTP response:

```
HTTP/1.1 200 OK
Date: Sat, 09 Oct 2022 14:28:02 GMT
Server: Apache
Last-Modified: Tue, 01 Dec 2019 20:18:22 GMT
ETag: "51142bc1-7449-479b075b2891b"
Accept-Ranges: bytes
Content-Length: 29769
Content-Type: text/html
```

<!DOCTYPE html... (the requested web page)

- Fecha/reutiliza a ligação para futuros HTTP requests.

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

scheme

host

path

Application Layer - HTTP

■ URL

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

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,...)



<https://www.semrush.com/blog/url-parameters/>

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

Application Layer - HTTP

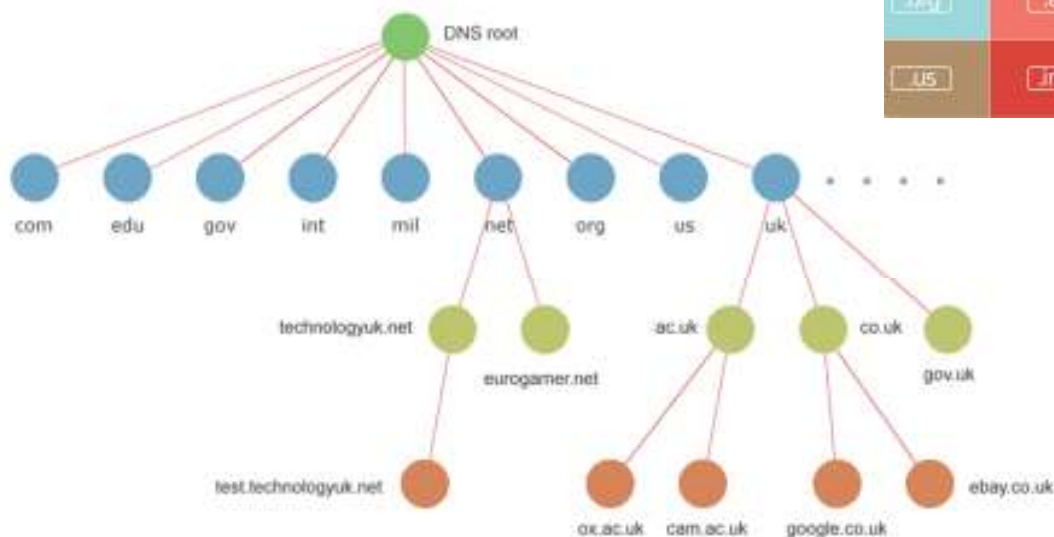
■ URL

■ host

- Endereço IP / domínio (DNS)

- <http://www.isec.pt>

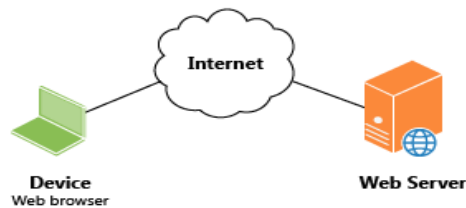
Top Level Domain



<https://www.icann.org/resources/pages/tlds-2012-02-25-en>

■ Pedido HTTP (request)

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



<http://tools.ietf.org/wg/httpbis/>

■ https (hyper text transfer protocol secure)

- Utiliza certificados SSL para encriptar a comunicação cliente/servidor.

https://ptisp.pt/alojamento-web/ssl?qclid=CjwKCAjwrPCGBhALEiwAUl9X076br8o9KjxRtLrL-RtY-H_cpg4Zpl_hetjWZIT1laMqdh3BvLxVhoCOjcQAvD_BwE

Evolução

world wide web

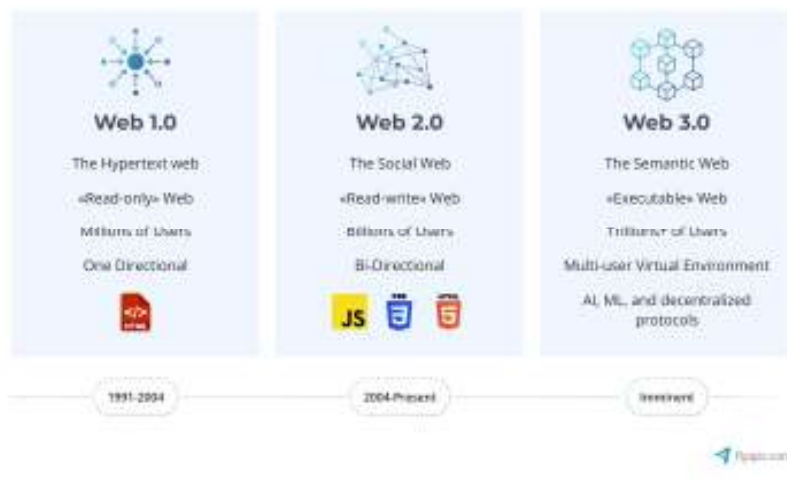
web 1.0

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

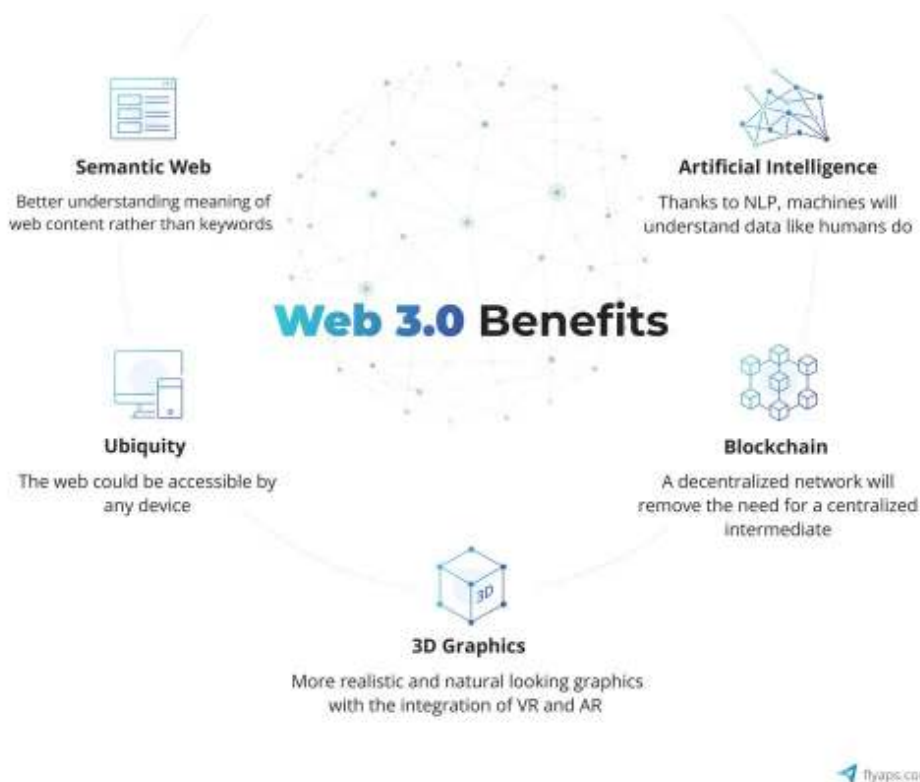
web 2.0

"... 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



<https://flyaps.com/blog/web-3-a-beginner-guide-to-decentralized-future/>



<https://flyaps.com/blog/web-3-a-beginner-guide-to-decentralized-future/>

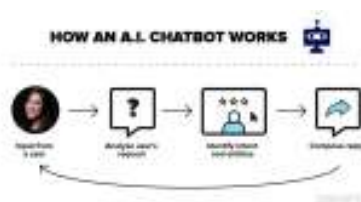
Web 3.0

Knowledge discovery

Semântica
Relevância dos Dados

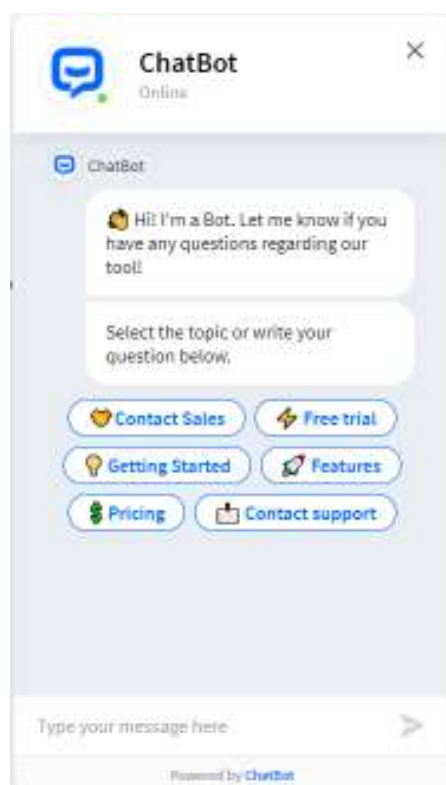
Inteligência Artificial

Personalização
NLP – Natural Language Processing



"An AI chatbot is trained to operate more or less on its own, using a process **known as Natural Language Processing, or NLP**, combined with **artificial intelligence** and the annotation of **human data**."

<https://www.drift.com/learn/chatbot/ai-chatbots/>



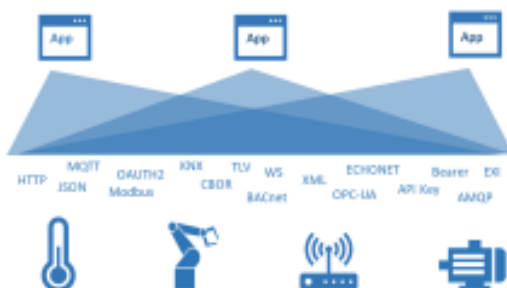
■ IOT (internet of things) & web of things



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

"The Internet of Things (IoT) is the network of physical objects or "things" embedded with electronics, software, sensors, actuators, and connectivity to enable objects to exchange data with the manufacturer, operator, and/or other connected devices."

<https://www.ietf.org/topics/iot>



"... have to understand a heterogeneous technology landscape consisting of diverse IoT systems and services from different vendors and manufacturers. **This diversity** includes variations in **communication protocols**, **data models for payload data exchange**, and **security requirements**."

"The Web of Things (WoT) provides a set of **standardized technology building blocks** that help to simplify IoT application development by following the well-known and successful Web paradigm."

<https://www.w3.org/WoT/documentation/>

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>

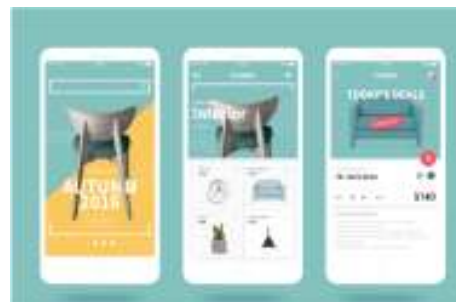
Principais Desafios

■ *web mobile*

- Atualmente o acesso por plataformas móveis é dominante.

*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/mobilweb>

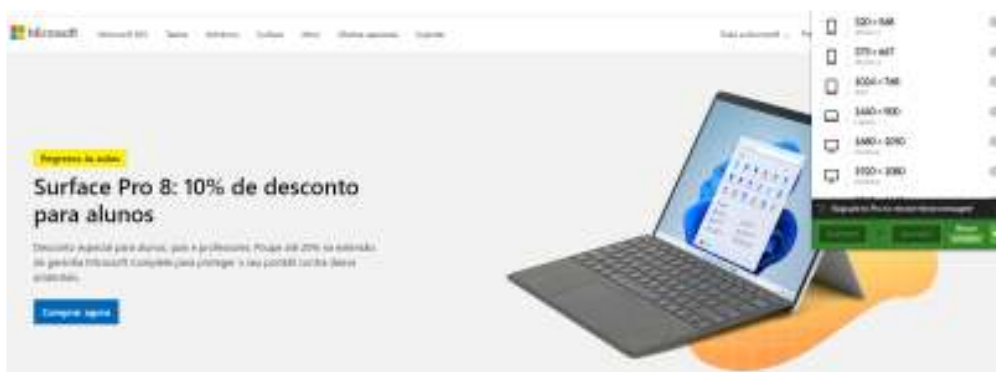


<https://explodingtopics.com/blog/mobile-internet-traffic>

Principais Desafios

■ *web mobile (responsive)*

- Testar a aplicação para as diversas plataformas é fundamental
- Existe um conjunto de ferramentas disponíveis que podem ser muito úteis



window resizer
(extensão Chrome)

Permite a visualização do conteúdo considerando vários breakpoints vulgarmente utilizados.

Principais Desafios

■ web mobile (responsive)

Chrome Dev Tools - Device Mode (Browser)

Permite a escolha de vários dispositivos para efetuar uma pré visualização dos conteúdos



Visualização em dispositivos móveis



Principais Desafios

■ web mobile (responsive)



Responsinator

Acesso a visualização em classes de dispositivos

<https://www.responsinator.com/?url=www.observador.pt>



BrowserStack

Testar a componente visual da aplicação num conjunto alargado de equipamentos

<https://www.browserstack.com/user/choose-product>

Principais Desafios

■ Segurança

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

- The yearly global cost of cybercrime is estimated to exceed **\$20 trillion by 2026**. (Cybersecurity Ventures)
- **2,244 cyberattacks** are happening every single day. (University of Maryland)
- There were **236.1 million ransomware attacks** in the first half of 2022. (Statista)
- **71% of organizations worldwide** have been victims from ransomware attacks in 2022. (Cybersecurity Ventures)
- **Organized crime** is responsible for 80% of all security and data breaches. (Verizon)
- Ransomware attacks happen every **10 seconds**. (InfoSecurity Group)
- **71% of all cyberattacks** are financially motivated (followed by intellectual property theft, and then espionage). (Verizon)

■ 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

■ Segurança

■ Alguns exemplos recentes:

Ataques de ransomware nas notícias

Infelizmente, as menções a [ameaças de ransomware](#) nas notícias são agora uma ocorrência comum. Ataques recentes de ransomware de grande repercussão têm afetado infraestruturas essenciais, a indústria da saúde e fornecedores de serviços de TI. Os efeitos destes ataques têm-se tornado mais imprevisíveis à medida que expandem o alcance. Vejamos alguns ataques de ransomware e como afetaram as organizações:

- Em março de 2022, o **sistema de correios da Grécia** sofreu um ataque de ransomware. O ataque interrompeu temporariamente a entrega de correio e afetou o processamento de transações.
- Uma das **maiores companhias aéreas da Índia** sofreu um ataque de ransomware em maio de 2022. O incidente provocou atrasos e cancelamentos de voo e centenas de passageiros ficaram retidos no aeroporto.
- Uma grande **empresa de recursos humanos** foi vítima de um ataque de ransomware em dezembro de 2021, e a respetiva folha de pagamentos e o sistema de folgas dos clientes que utilizam o respetivo serviço da nuvem foram afetados.
- Em maio de 2021, um **oleoduto americano** interrompeu os serviços para impedir mais falhas de segurança depois de um ataque de ransomware ter comprometido milhares de informações pessoais dos respetivos colaboradores. Este evento fez disparar os preços dos combustíveis em toda a costa leste.
- Uma **empresa de distribuição de produtos químicos alemã** sofreu um ataque de ransomware em abril de 2021. Foram roubadas datas de nascimento, números da Segurança Social, números da carta de condução e dados médicos de mais de 6000 pessoas.
- O **maior fornecedor de carne a nível mundial** foi alvo de um ataque de ransomware em maio de 2021. Depois de suspender temporariamente o respetivo site e interromper a produção, a empresa acabou por pagar um resgate de 11 milhões USD em Bitcoin.

<https://www.microsoft.com/pt-pt/security/business/security-101/what-is-ransomware>

Principais Desafios

■ Segurança



TIME IT TAKES A HACKER TO BRUTE FORCE YOUR PASSWORD IN 2022					
Number of Characters	Numbers Only	Lowercase Letters	Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters, Symbols
4	Instantly	Instantly	Instantly	Instantly	Instantly
5	Instantly	Instantly	Instantly	Instantly	Instantly
6	Instantly	Instantly	Instantly	Instantly	Instantly
7	Instantly	Instantly	2 secs	7 secs	31 secs
8	Instantly	Instantly	2 mins	7 mins	39 mins
9	Instantly	10 secs	1 hour	7 hours	2 days
10	Instantly	4 mins	3 days	3 weeks	5 months
11	Instantly	2 hours	5 months	3 years	34 years
12	2 secs	2 days	24 years	200 years	3k years
13	19 secs	2 months	1k years	12k years	202k years
14	3 mins	4 years	64k years	750k years	16m years
15	32 mins	100 years	3m years	46m years	1bn years
16	5 hours	3k years	173m years	3bn years	92bn years
17	2 days	69k years	9bn years	179bn years	7tn years
18	3 weeks	2m years	467bn years	11tn years	438tn years

 > Learn about our methodology at [hivesystems.io/password](https://www.hivesystems.io/password)

<https://www.hivesystems.io/>

Principais Desafios

■ Desempenho

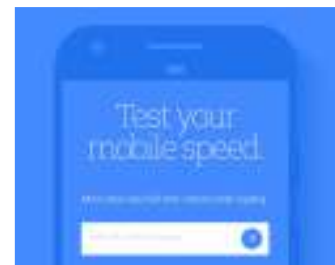
■ Single Page Application (SPA)

- Otimizar a transferência de dados

■ Progressive Web Apps

■ Guidelines para melhorar o desempenho

- Reduzir o número de pedidos (*requests*) ao servidor
- Evitar atributos src e href vazios
- Remover JS e CSS duplicados
- Minificar JS/CSS
- Limitar a dimensão dos ficheiros (imagens, vídeos), ...



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

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

■ Testar

- Várias opções disponíveis para teste de performance

- <https://dynamap.com/blog/21-sitemaps-and-seo/457-top-15-tools-for-measuring-website-or-application-speed>

Principais Desafios

- Desempenho: testar
 - Google tool <https://testmysite.withgoogle.com/intl/en-gb>

Your speed results for isec.pt ①

Your mobile site speed is 1.7 seconds in Portugal - on a 4G - connection.

RATING

Fast

[Learn more](#)

MONTHLY TREND

No Change

Your site speed has not changed since last month.
[Learn more](#)



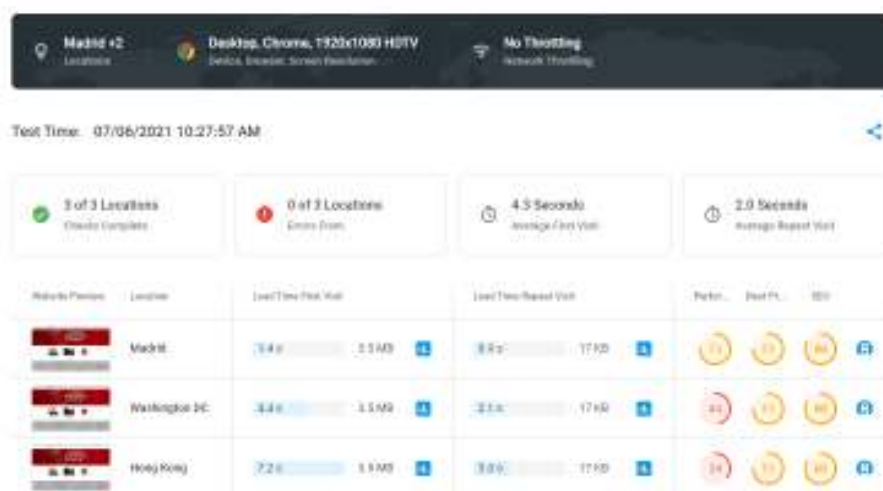
Learn how to optimise your mobile site



Principais Desafios

- Desempenho: testar
 - Dados completos sobre o desempenho das aplicações
 - velocidade de download a partir de diferentes localizações (e.g. dot-com-tools, ...)

<http://www.isec.pt>
Website Speed Test Statistics Report



<https://www.dotcom-tools.com/website-speed-test?type=summary-report&id=9d9e5d9adb814f40a309027a402853b5>

Principais Desafios

■ Search Engine Optimization (SEO)

■ Visibilidade

*“In simple terms, it means the process **of improving the site to increase its visibility** when people search for products or services related to your business in Google, Bing, and other search engines. The better visibility your pages have in search results, the more likely you are to gather attention and attract prospective and existing customers to your business”*

<https://searchengineland.com/guide/what-is-seo>

■ Guidelines

- Criação de títulos únicos e precisos em cada página
- Descrição precisa do conteúdo
- Indicação de *keywords*
- Descrição do documento
- <meta> tag ...



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

■ https://developers.google.com/search/docs/beginner/seo-starter-guide?hl=en&visit_id=637611750414470147-3671772374&rd=1

Principais Desafios

■ Search Engine Optimization (SEO)

■ Testar

- Existem várias ferramentas de teste disponíveis no mercado



<https://www.seoptimer.com/>

www: Principais Desafios

■ Cross browser Testing

- 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*
 - Assegurar que o conteúdo nuclear é disponibilizado de forma uniforme entre browsers
 - Avaliar a utilização de determinadas soluções (ex: advanced CSS features) que possam diferir entre browsers

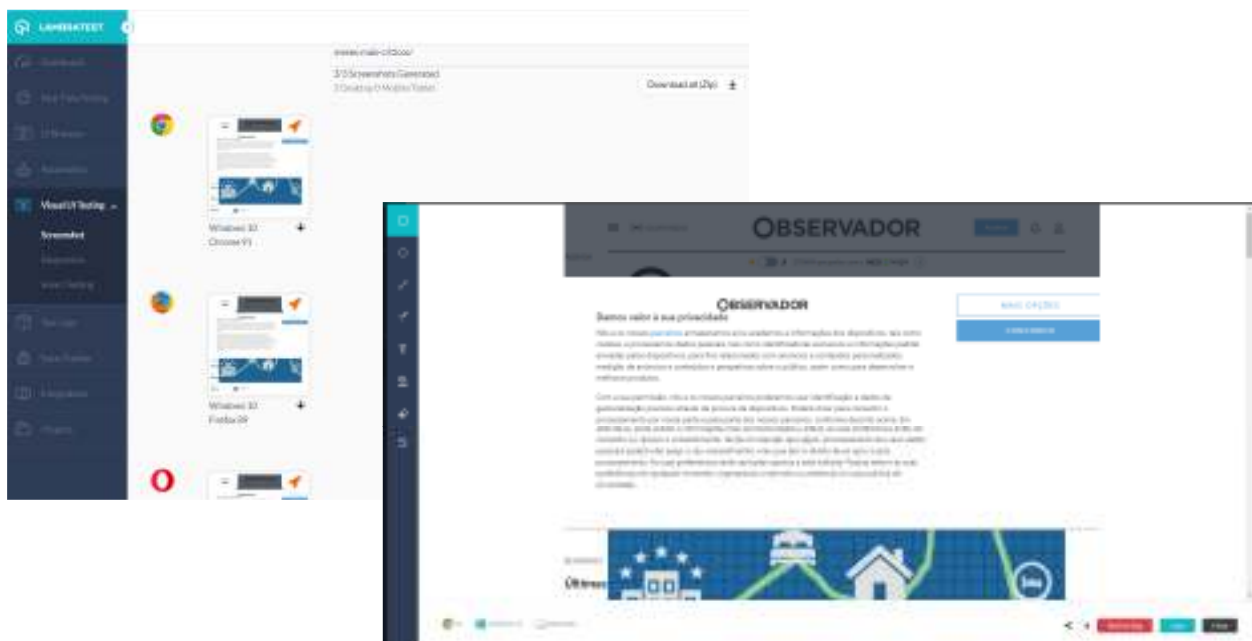
• <https://www.softwaretestinghelp.com/best-cross-browser-testing-tools-to-ease-your-browser-compatibility-testing-efforts/>



www: Principais Desafios

■ Cross browser Testing

- exemplo de verificação de UI tendo por base 3 browsers diferentes:



■ Acessibilidade

- Reduzir as barreiras ao acesso à informação por parte dos utilizadores com necessidades especiais (invisuais, ...)

- *Web Content Accessibility Guidelines (WCAG)* <http://www.w3.org/WAI>

- Ferramentas de teste: <https://wave.webaim.org/>

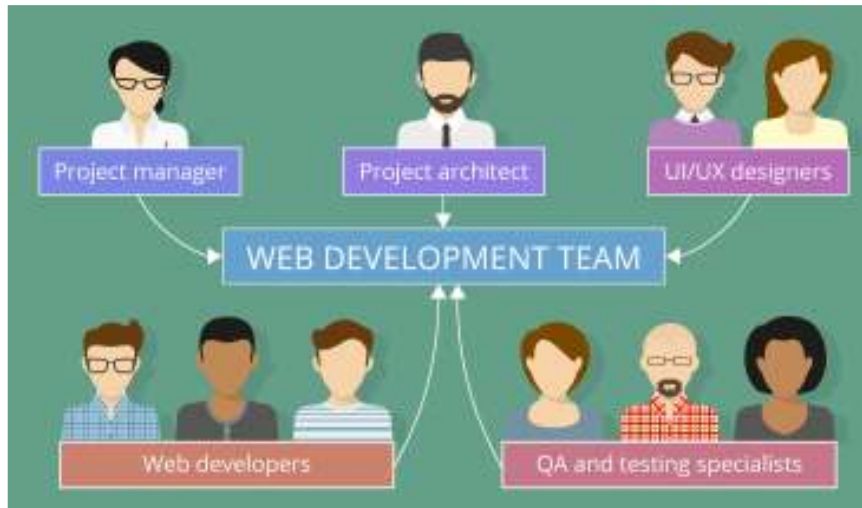


Web Development

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>



web development

- Existem 3 perfis associados ao web developer:

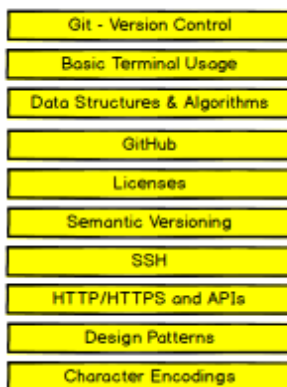
Front-end developer

Back-end developer

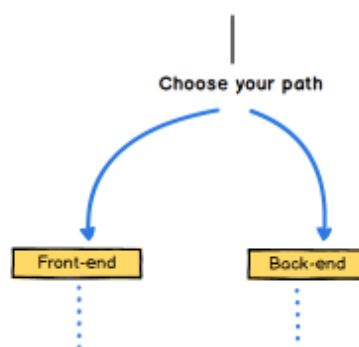
Full-stack developer

- Qualquer destes perfis exige o conhecimento de um conjunto de conceitos e tecnologias que se encontram em permanente evolução

Required for any path



Web Developer



Find the detailed version of this roadmap along with resources and other roadmaps

<http://roadmap.sh>

<https://github.com/kamranahmedse/developer-roadmap>

Front-end Development

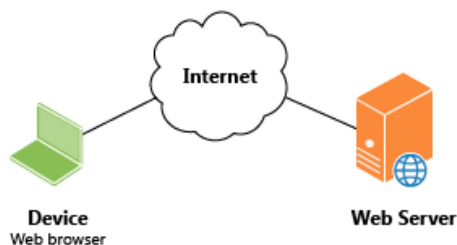
world wide web

Front End Development: Tecnologias

- Diretamente interpretadas pelo browser:

- Tecnologias base:

- HTML
- CSS
- JavaScript



Client Side Web Technologies

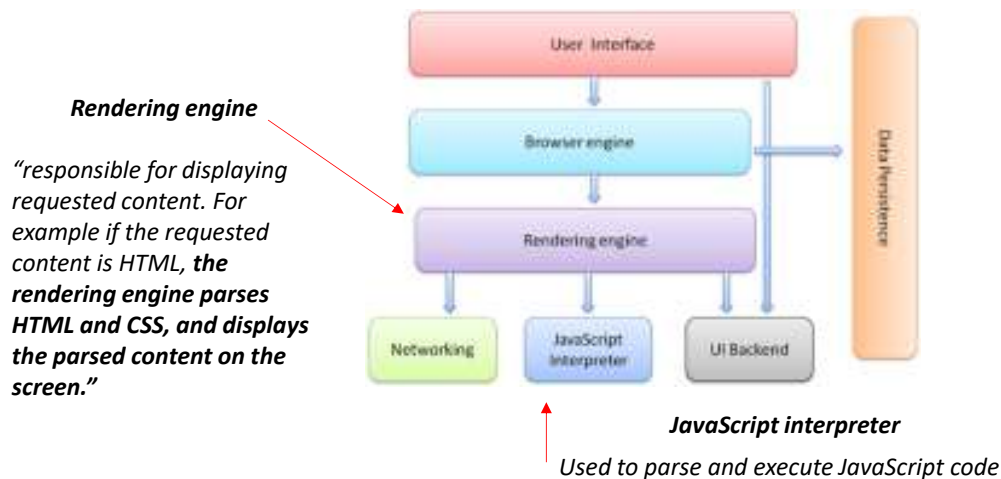


Front End Development: Tecnologias

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



- Browser Components <https://www.html5rocks.com/en/tutorials/internals/howbrowserswork/>



55

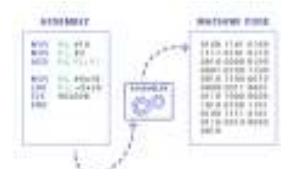
Front End Development: Tecnologias

- Na realidade, atualmente os browsers também podem executar WebAssembly (WASM)



<https://blog.logrocket.com/webassembly-how-and-why-559b7f96cd71/>

- O Javascript é particularmente bem adaptado na definição da interação com o utilizador, mas é pouco eficiente em contextos que requerem processamento intensivo
 - exemplos: incorporação numa aplicação web de jogos, edição vídeo, edição de audio, etc.
- WebAssembly (WASM) é um código de baixo nível, concebido para ser leve e rápido, que resulta da compilação de uma linguagem de alto nível (C, C++) e é incorporado via JavaScript.

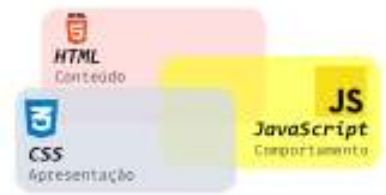


56

Front End Development

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

- Estrutura / Conteúdo
- Apresentação
- Comportamento



Front End Development: Tecnologias

- Tecnologias diretamente interpretadas pelo browser:



- Atualmente o conhecimento destas tecnologias é suficiente no *front-end development*?

- **Não!**

- Existe um conjunto de *frameworks/tecnologias* associadas utilizados para:
 - Potenciar/optimizar o processo de desenvolvimento
 - Melhorar a capacidade de apresentação
 - Melhorar funcionalidades
 - Implementar “novos” paradigmas na implementação de aplicações web (ex: SPA)

Front End Development: Tecnologias

■ Frameworks CSS

- *Bootstrap*
- *Foundation, ...* <https://geekflare.com/best-css-frameworks/>



<https://getbootstrap.com/>

■ Pré-processadores CSS

- conjunto adicional de funcionalidades de forma a agilizar a construção das CSS
 - SASS
 - LESS



<https://sass-lang.com/>

Front End Development: Tecnologias

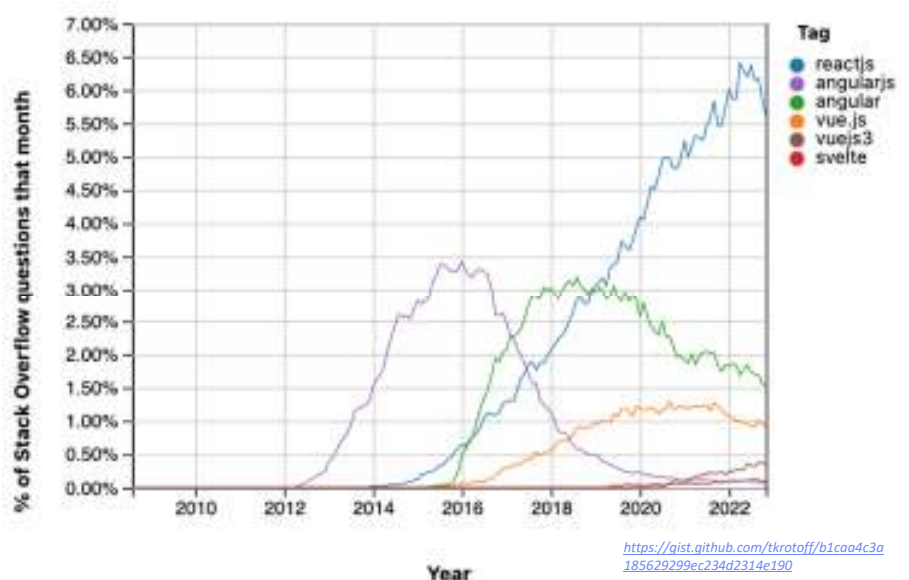
■ Otimização comunicação Cliente- Servidor

- AJAX
 - Objeto JS: XMLHttpRequest

■ front end frameworks

- *React*
- *Angular*
- *Vue,*
- ...

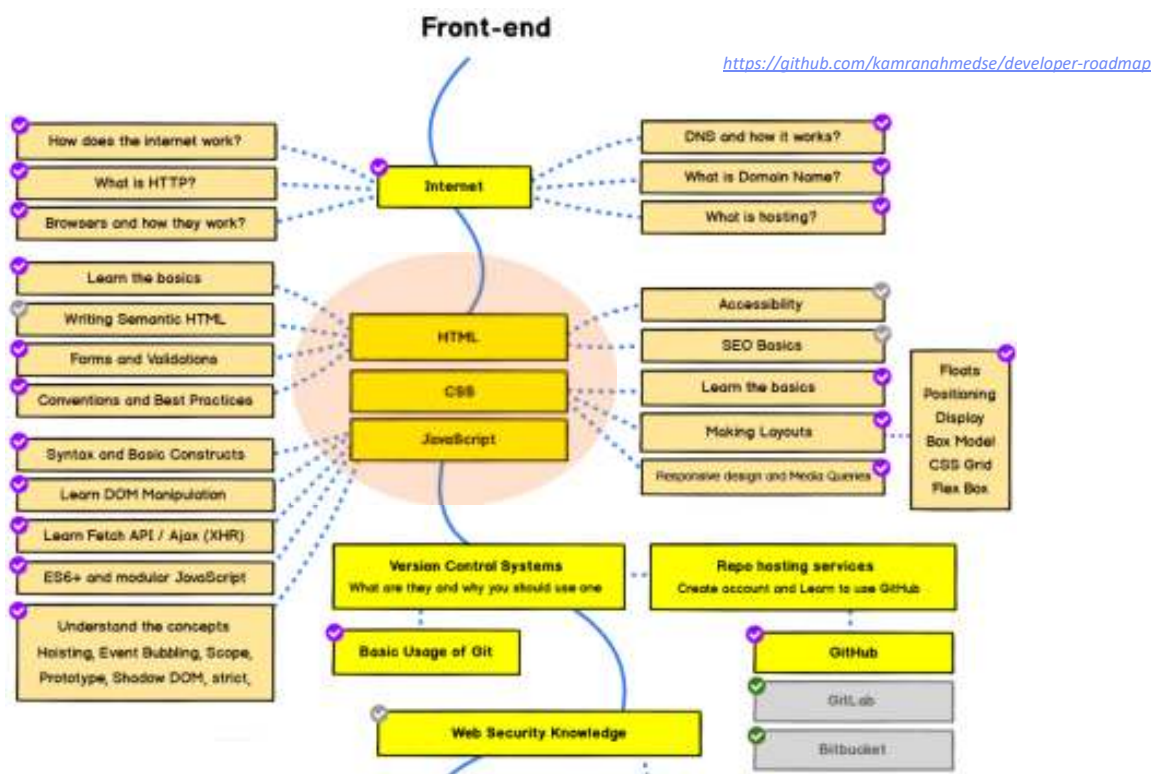
<https://hackr.io/blog/best-javascript-frameworks>



<https://gist.github.com/tkrotoff/b1ca04c3a185629299ec234d2314e190>

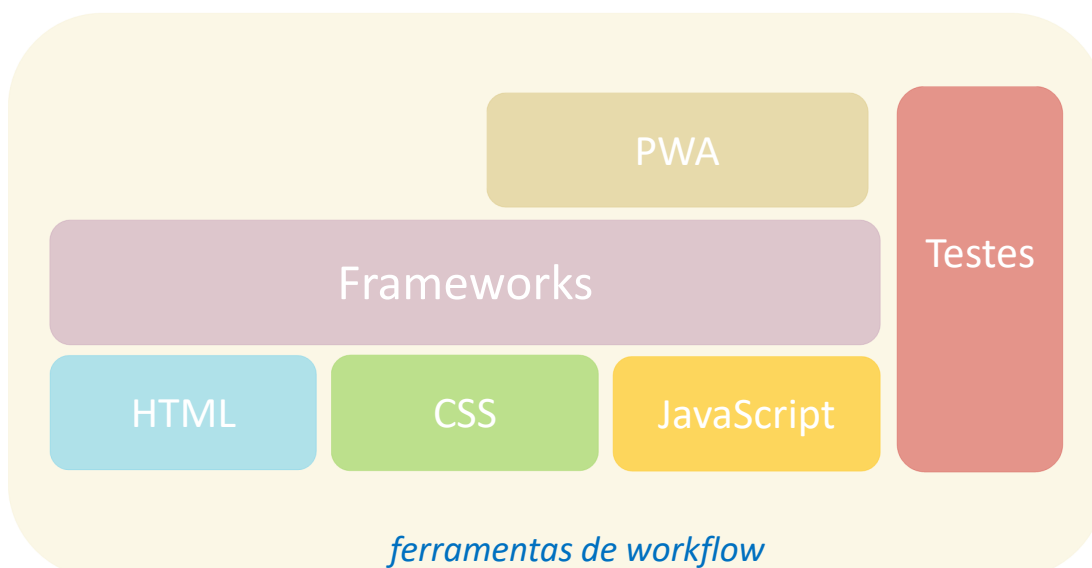
Front End Development: Tecnologias

- O roadmap do front-end developer envolve múltiplas tecnologias:



Front End Development: Tecnologias

- Front End Web Development
 - Roadmap



- *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



<https://www.javatpoint.com/cms-full-form>

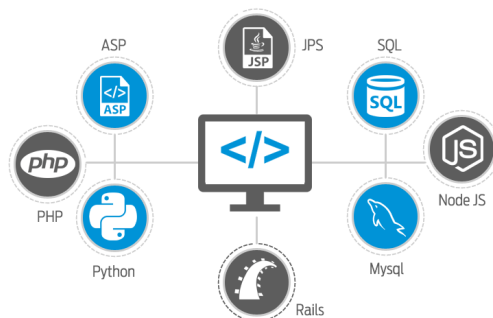
- Comparativo entre as diferentes plataformas

- <https://www.wpbeginner.com/showcase/best-cms-platforms-compared/>

Natureza do Conteúdo

Natureza do Conteúdo

- Estático
 - O conteúdo não pode ser actualizado/alterado sem que haja uma alteração no código-fonte do site (atualmente apenas implementado em situações muito específicas).
- Dinâmico
 - Os conteúdos são alterados sem necessidade de alterações no código-fonte.
 - Implementação generalizada.

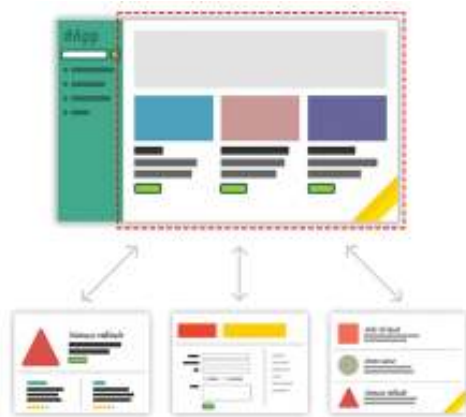


<https://www.addonsolutions.com/blog/selection-dilemma-static-vs-dynamic-web-development.html>

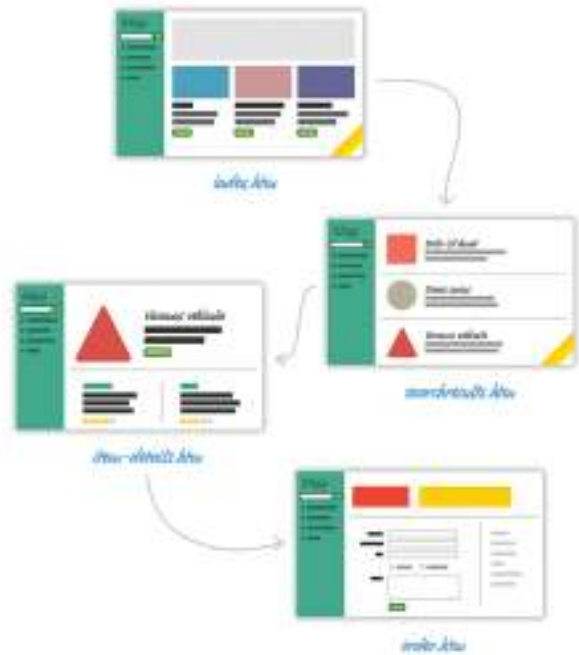
- *Server side: Linguagens Script + Base de Dados*
- *Client side: Javascript*

Single Page Application vs. Multi Page Application

Single Page Application



Multi Page Application



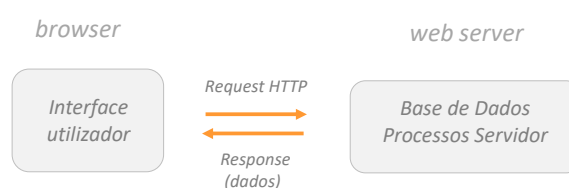
Chinnathambi, K., *Learning React*, ISBN: 970-0-13-484355-1, Addison-Wesley, 2018

Multi Page web Applications (MPA)

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



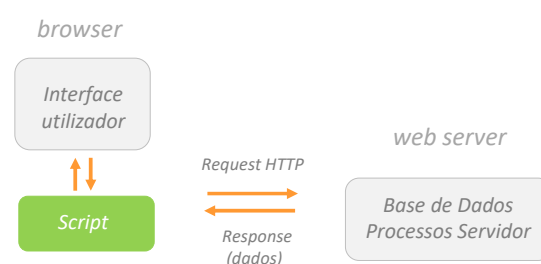
Multi Page Application (MPA)

- 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. Pedido HTTP (início da sessão)
 2. Servidor envia um script 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 atualização 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 (Angular, 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

Progressive Web Application (PWA)

Progressive Web Application (PWA)



- Aplicações web :
 - Não estão diretamente dependentes de dispositivos e sistemas operativos (nessa perspetiva são ubíquas)
 - Não são centradas nem controladas por nenhuma empresa comercial
 - Assente na capacidade de estabelecer ligações, o que permite a pesquisa e partilha imediata de informação

*Web applications can **reach anyone, anywhere, on any device** with a single codebase.*

Platform-Specific Apps



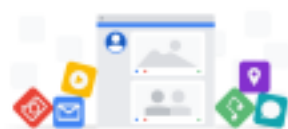
- Aplicações nativas (mobile/desktop):
 - Específicas de cada plataforma/sistema Operativo
 - Fiáveis
 - Múltiplas funcionalidades
 - Funcionam independentemente da ligação à rede (offline)
 - A aplicação é iniciada de forma totalmente programável/diferenciada (splash screen)
 - Permitem a interação com ficheiros armazenados localmente, hardware
 - Notificações push

*Native applications are **rich and reliable**, feel like part of the device they run on.*

<https://developers.google.com/codelabs/pwa-training/pwa03-going-offline#0>

Progressive Web Application (PWA)

Platform-Specific Apps



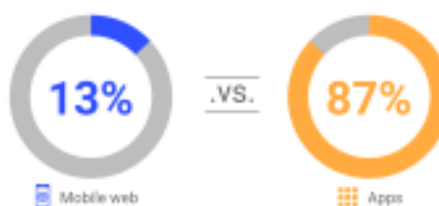
*Elevada capacidade
Muitas funcionalidades*



Elevada Disponibilidade

<https://web.dev/what-are-pwas/>

Tempo dispendido
pelos utilizadores



<https://www.comscore.com/Insights/Presentations-and-Whitepapers/2020/Global-State-of-Mobile>

Progressive Web Application (PWA)

■ PWA

- Pretendem tirar partido das vantagens dos dois universos
 - Ter uma capacidade semelhante a uma *native app* sem necessidade de *download* a partir da Google Play ou da Apple App Store.
 - Fluidez
 - Rapidez no acesso aos conteúdos / sem tempo de espera
 - Disponibilidade/possibilidade de funcionamento offline
 - Simultaneamente
 - Ter a capacidade de estar disponível independentemente da plataforma
 - Interligação de conteúdos



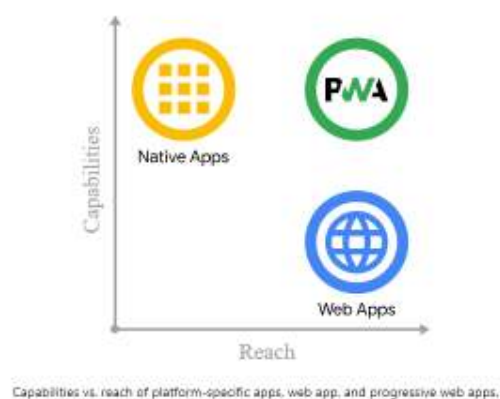
Excerpted as part of platform-specific apps, web apps, and progressive web apps.

<https://web.dev/what-are-pwas/>

Progressive Web Application (PWA)

Progressive Web Apps (PWAs) are web apps built and enhanced with modern APIs to deliver enhanced **capabilities, reliability, and installability** while reaching **anyone, anywhere, on any device**, all with a single codebase.

<https://www.comscore.com/Insights/Presentations-and-Whitepapers/2020/Global-State-of-Mobile>



Progressive Web Application (PWA)

- PWA
 - Características Obrigatórias (core)
 - Velocidade (starts fast / stays fast)
 - Tendo por base o serviço de *cache* assegurado pelos *service workers* permite que uma resposta a pedidos semelhantes seja entregue de forma muito mais rápida



<https://web.dev/progressive-web-apps/>

- Cross Platform
 - Executada em várias plataformas, independente do sistema operativo, uma vez que é executada diretamente no browser

Progressive Web Application (PWA)

- Independente da conectividade
 - Permite o funcionamento offline baseado no service worker
- Responsive
 - Adapta-se ao dispositivo no qual está a ser executada
- Permite instalação
 - Na 1ª interação é proposto ao utilizador a sua instalação e adição ao ecrã inicial
- Atualizada
 - Conteúdo sempre atualizado, permite notificações push
- Funcionalidades
 - Tendo por base o desenvolvimento de novas API's o grau de interação e de funcionalidades disponíveis podem aproximar-se ao de uma aplicação nativa

<https://developers.google.com/codelabs/pwa-training/pwa03--going-offline#0>

Progressive Web Application (PWA)

- PWA
 - Vantagens
 - Mais baratas de desenvolver
 - Experiência de utilização satisfatória
 - Multiplataforma
 - Evita o desenvolvimento android, ios
 - A instalação não é obrigatória, pelo que permite uma utilização pontual sem necessidade de instalação
 - Desvantagens
 - O design não explora todas as potencialidades de cada uma das plataformas
 - Grafismo genérico
 - Ainda que aproximado, não consegue obter o nível de interação proporcionado por uma aplicação nativa
 - *trends*
 - <https://www.neoito.com/blog/the-future-of-pwa-and-web-apps/>
 - <https://mobidev.biz/blog/why-when-use-progressive-web-app-pwa-development>

<https://developers.google.com/codelabs/pwa-training/pwa03--going-offline#0>

Comunicação entre Aplicações/Plataformas (web services)

81

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/>



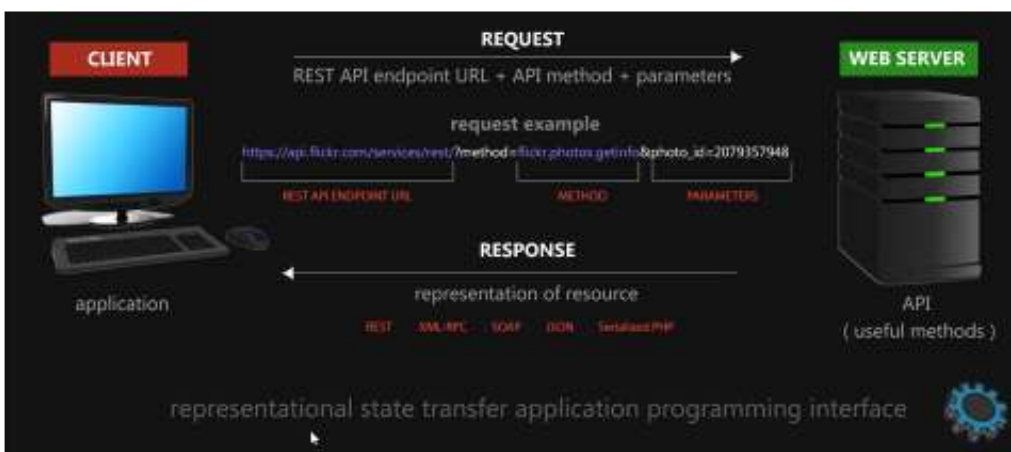
- Baseados na arquitetura REST (**Representational State Transfer**)
 - Nem todos os web services são baseados em REST (ex:SOAP)
 - Muito utilizados na *www* devido à simplicidade na sua implementação
 - Cada comunicação é totalmente independente da anterior e da seguinte
 - Todos os dados são enviados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



<https://www.datamounts.com/>



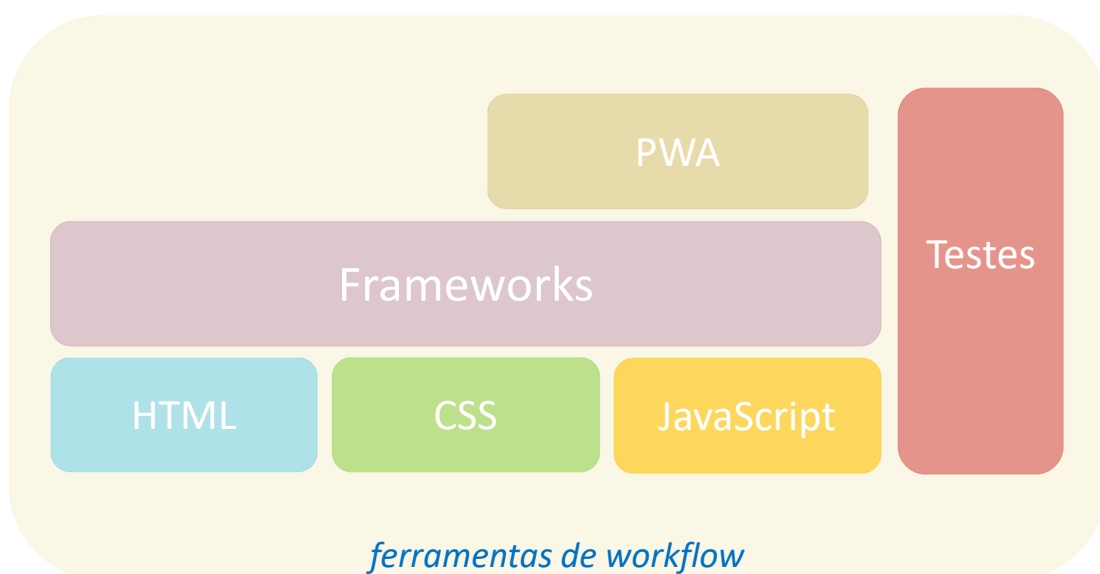
Numa interação cliente - servidor é solicitado ao servidor um determinado recurso (*.jpg; *.html)



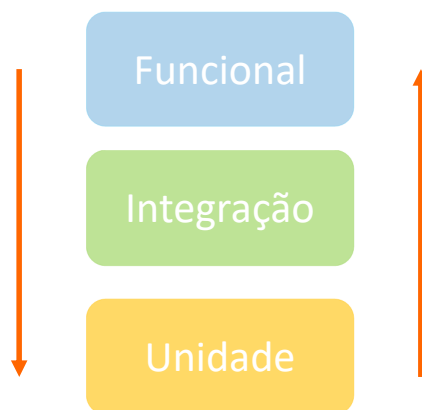
Num *REST Request* podem ser invocados um conjunto de métodos/funções disponibilizados pela REST API aos quais são passados parâmetros de forma a obter os dados desejados (*REST Response*).

Testes

Testes



- Os testes são absolutamente essenciais ao longo do ciclo de vida do desenvolvimento de uma aplicação web
 - Processo contínuo que acompanha o desenvolvimento de funcionalidades
 - Efetuada por elementos exteriores ao desenvolvimento (ideal)
 - Podem ser organizados em 3 grandes grupos:



- Testes de Unidade
 - Teste de módulos individuais sem nenhuma interação com dependências
 - Teste funcional do código (funções, métodos, classes,...)
- Testes de Integração
 - Teste da interligação entre diferentes módulos
 - Incidem essencialmente na transmissão de dados entre os vários componentes de um sistema
- Testes Funcionais
 - Teste das funcionalidades do sistema
 - Teste feito na perspetiva do utilizador, idealmente por não programadores exteriores à implementação da aplicação
 - O objetivo é percorrer todos os requisitos funcionais da aplicação e garantir dessa forma que toda a aplicação funciona como planeado.

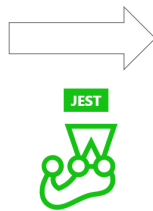
■ Automação de testes

- Existem várias ferramentas de teste vocacionadas para aplicações web:

- Jest <https://jestjs.io/>
- Enzyme
- Cypress
- Mocha

```
const functions = {  
  sum: (a, b) => {  
    return a + b  
  },  
  subtract: (a, b) => {  
    return a - b  
  }  
}  
module.exports = functions
```

functions.js



```
const functions = require('./functions.js')  
  
test('adds properly', () => {  
  expect(functions.sum(1, 5)).toBe(6)  
})  
  
test('subtracts properly', () => {  
  expect(functions.subtract(1, 2)).toBe(-1)  
})
```

functions.test.js

Human Computer Interaction (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)



<https://www.glasbergen.com/>

- Dois aspetos centrais que caracterizam o comportamento dos utilizadores:

- O utilizador não lê em pormenor, faz um **scan** do espaço visual
- Escolhe a primeira alternativa que lhe parece razoável para obter a informação desejada



<http://www.useit.com/>
<http://www.usabilityfirst.com/>

- A chave do sucesso é a **simplicidade**:

“Não Fazer Pensar”

- Intuitivo
- Auto-explicativo
- Deve ser utilizado sem exigir um esforço significativo para a compreensão do seu funcionamento



Intuitivo ?

1. Consistência Gráfica

Consistência Gráfica

- *Single Page Application (SPA)*
 - Se bem implementada a consistência gráfica está naturalmente assegurada!
- Multi page Application (MPA)
 - Deve ser garantida tanto ao nível da estrutura como de formatação
 - Explorar convenientemente tecnologias para formatação dos conteúdos
 - Estrutura/conteúdo
 - Elementos comuns
 - Mesma estratégia para disposição do conteúdo
 - ...
 - Formatação
 - Divisão das páginas em áreas bem definidas, ...
 - Hierarquia visual
 - Mesmo conjunto de cores
 - ...



Consistência gráfica

Elementos comuns

Mesmas cores

Disposição dos conteúdos

...



<https://www.apple.com/ipad/>

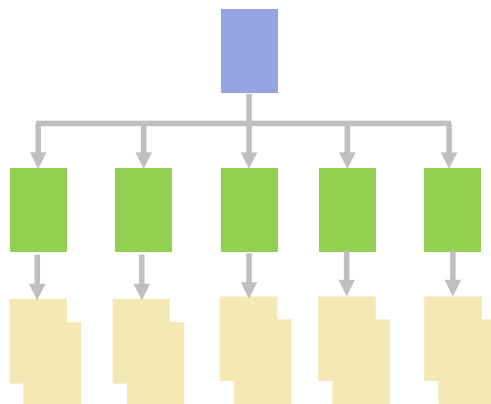
Intuitivo ?

2. Estrutura de Informação

Estrutura de Informação

■ Estrutura de uma aplicação

- O utilizador deve compreender de forma intuitiva a estrutura de informação.



- A profundidade da estrutura de informação de uma aplicação é função da quantidade de informação disponibilizada.

Estrutura de Informação



homepage



iPhone



Watch

Intuitivo ?

3. Homepage

Homepage

- **Atrativa**
 - Capta a atenção dos utilizadores e inspira credibilidade e confiança.
- **Confusa/mal estruturada**
 - Afasta do site os seus potenciais utilizadores.
- **Os elementos a incluir são diversos e dependem da natureza do site**
 - Os sites de natureza Institucional / Comercial são os mais exigentes do ponto de vista do equilíbrio na disponibilização da informação.
 - Promoção de produtos;
 - Promoção de serviços;
 - Dinâmica das organizações,
 - ...



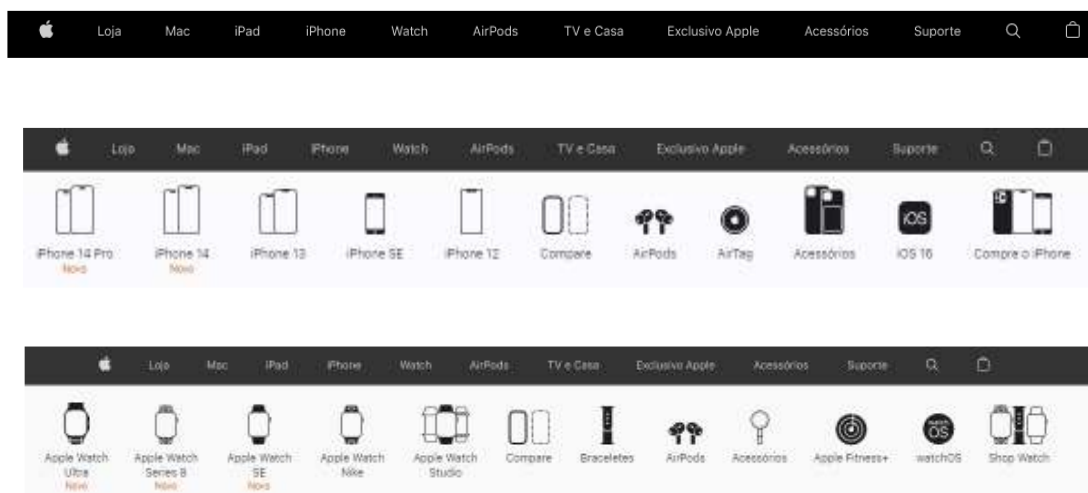
Intuitivo ?

4. Navegação

Navegação

- Deve ser implementado o conceito de Navegação Persistente ou Global

Algumas partes do site podem sofrer alterações, dependendo da zona a que se está a aceder, mas a estrutura é sempre a mesma e funciona sempre da mesma forma



Navegação

- Âncora de Navegação
 - A estrutura de navegação deve estar sempre acessível ao utilizador (nunca fica perdido)



- Metáforas
 - Reforço à navegação (ex: carrinho/cesto de compras)



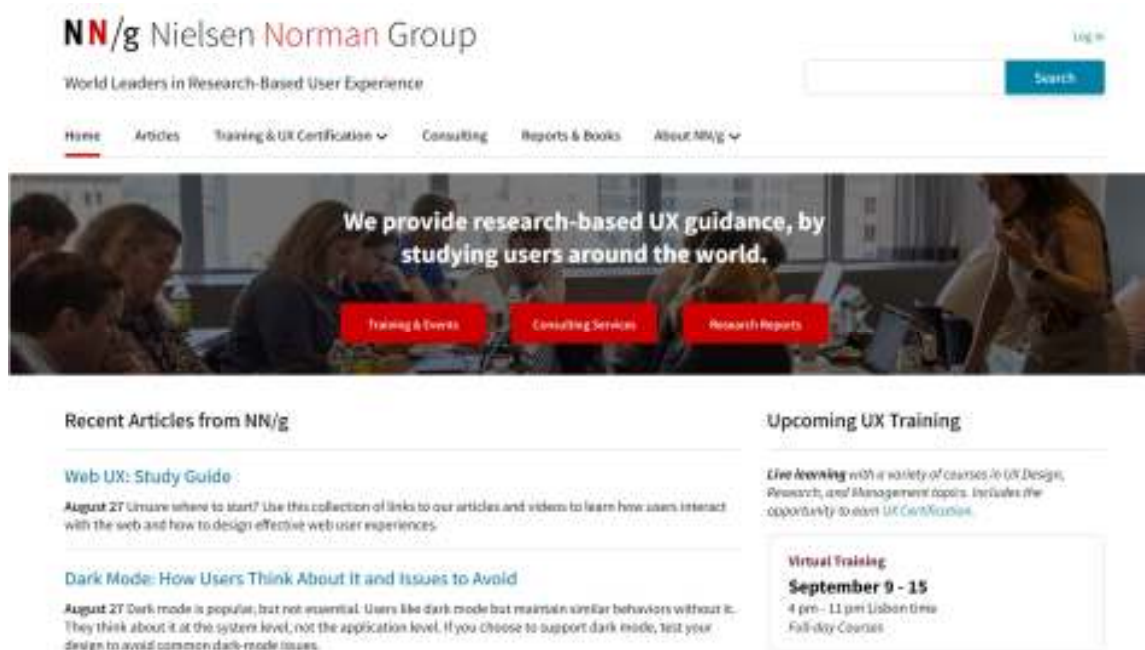
- Identificação de links
 - Formatação diferente / Sublinhado
 - Menus (verticais / horizontais)
 - Marcadores (tabs)
 - Botões, ...

Intuitivo ?

5. Conteúdos

Disposição dos conteúdos

- Espaçoamento
 - Divisão da Informação em áreas bem definidas
 - Perceção imediata da organização de conteúdos



Disposição dos conteúdos

■ Espaçamento

■ Exemplo: Identificação

- O logótipo não necessita de ser muito grande (com o espaçamento pode ser facilmente destacado)

NN/g Nielsen Norman Group

World Leaders in Research-Based User Experience

[Home](#) [Articles](#) [Training & Events](#) [Consulting](#) [Reports & Books](#) [About NN/g](#)

■ Deve ser sempre evitada uma grande concentração de informação

- O espaço livre é fundamental para captar a atenção do utilizador!
- Com conteúdos textuais, o espaço vazio deve ser cerca de **50%** do espaço disponível

Conteúdos

■ Formulários

- O mais simples e objetivos possível
- Obter apenas a informação necessária

- Text box
- Menus (opção / acção),
- Push Button (acção);
- Radio Button (opção),
- Check box (opção);
- ...

Microsoft account

Already have a Microsoft account? If you use **Hotmail**, **SkyDrive**, **Xbox LIVE** and **Outlook**, you can sign in with the same account.

Not ready to sign up yet? - [Learn more](#)

Who are you?

Name

First name Surname

Birth date

Day Month Year

Gender

Select one

How would you like to sign in?

Microsoft account name

@ outlook.com

Create a password

8-character minimum; case-sensitive

Re-enter password

Conteúdos

■ Menus

- Evitar uso de linhas separadoras
 - Espaçamento é muito mais eficaz!
- Limitar o número de opções
 - Em alternativa deve ser equacionado o aumento da profundidade dos menus



Conteúdos

■ Texto

- Deve ser o mais objetivo possível
- Não deve ser contínuo
 - A divisão do texto em parágrafos é essencial
 - Facilita a leitura e compreensão da mensagem por parte do utilizador.
 - Uma ideia por parágrafo
- Garantir o espaçamento entre linhas
 - Evitar sobreposições ou espaço exagerado entre linhas
- Nunca utilizar alinhamento centrado para corpo de texto
- Revisão Ortográfica e Gramatical
 - Obrigatória!!!

Intuitivo ?

6. Cor

Cor

- Gestão da Cor
 - É um aspeto central na implementação de uma aplicação web
 - Se bem utilizada orienta a visão para a informação
 - Se mal utilizada reduz significativamente a eficiência da comunicação
 - Combinação de cores
 - Devem ser testadas várias combinação de cores



<https://designschool.canva.com/blog/website-color-schemes/>



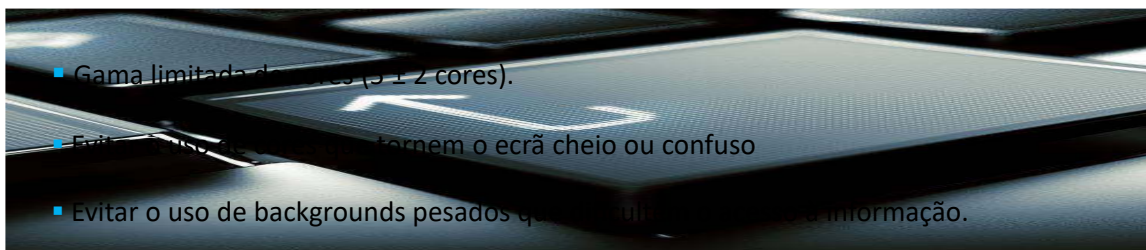
<https://www.designwizard.com/blog/design-trends/colour-combination>
<https://en.99designs.pt/blog/creative-inspiration/color-combinations/>

■ Conceitos Principais

■ Consistência

- Adotar o mesmo conjunto de cores ao longo de toda a *web application*

■ Simplicidade & Legibilidade



- A suavidade da cor é um bom caminho para a simplicidade pretendida;
 - Regra geral, um bom fundo passa pela escolha do branco.

■ Cores saturadas

■ Nunca devem ser utilizadas para o texto:

- Fadiga ; desconforto visual; ilegibilidade; ...
- Só devem ser utilizadas nos casos em que se pretende atribuir um ênfase especial a algum elemento.
- Pelo mesmo motivo também não devem ser usadas como cor de *background*:



<https://webdesign.tutsplus.com/articles/why-you-should-avoid-vibrating-color-combinations--cms-25621>