

P. PORTO

TECNOLOGIAS WEB

TECNOLOGIAS E SISTEMAS DE INFORMAÇÃO PARA A WEB

POLITÉCNICO
DO PORTO
ESCOLA
SUPERIOR
DE MEDIA ARTES E
DESIGN

2022/2023

M01 - TECNOLOGIAS WEB

MO1 - TECNOLOGIAS WEB

1. World Wide Web
2. Modelo Cliente-Servidor
3. Protocolo HTTP
4. Tecnologias
5. Visual Studio Code
6. GIT

WORLD WIDE WEB

- **WWW = World Wide Web**
- Espaço de informação global e ubíqua onde é feita a gestão de recursos Web
- Criada por Tim Berners-Lee em 1990 e Mantida pelo W3C
- Os recursos Web são:
 - ▶ hipertextos, imagens, sons, vídeos, etc.
 - ▶ armazenados em **servidores**
 - ▶ identificados por **URL** (Uniform Resource Locator)
 - ▶ acedidos por **browsers**
 - ▶ navegáveis por **hipertexto**

WORLD WIDE WEB

Primeira página Web¹

World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#), [Policy](#), November's [W3 news](#), [Frequently Asked Questions](#).

[What's out there?](#)

Pointers to the world's online information, [subjects](#), [W3 servers](#), etc.

[Help](#)

on the browser you are using

[Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#), [X11 Viola](#), [NeXTStep](#), [Servers](#), [Tools](#), [Mail robot](#), [Library](#))

[Technical](#)

Details of protocols, formats, program internals etc

[Bibliography](#)

Paper documentation on W3 and references.

[People](#)

A list of some people involved in the project.

[History](#)

A summary of the history of the project.

[How can I help ?](#)

If you would like to support the web..

[Getting code](#)

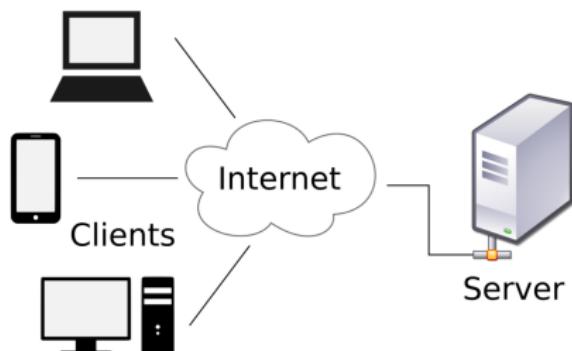
Getting the code by [anonymous FTP](#), etc.

¹<http://info.cern.ch/hypertext/WWW/TheProject.html>

MODELO CLIENTE-SERVIDOR

MODELO CLIENTE-SERVIDOR

- WWW assenta num modelo cliente-servidor que distribui as tarefas/cargas de trabalho entre:
 - ▶ Clientes
 - ▶ Servidores
- Clientes
 - ▶ Parte ativa
 - ▶ Pede um recurso
 - ▶ Processa um recurso
- Servidores
 - ▶ Parte passiva
 - ▶ Recebe um pedido
 - ▶ Fornece o recurso
- Comunicação
 - ▶ Usa um localizador de recursos (URL)
 - ▶ Requer um protocolo (HTTP)



MODELO CLIENTE-SERVIDOR

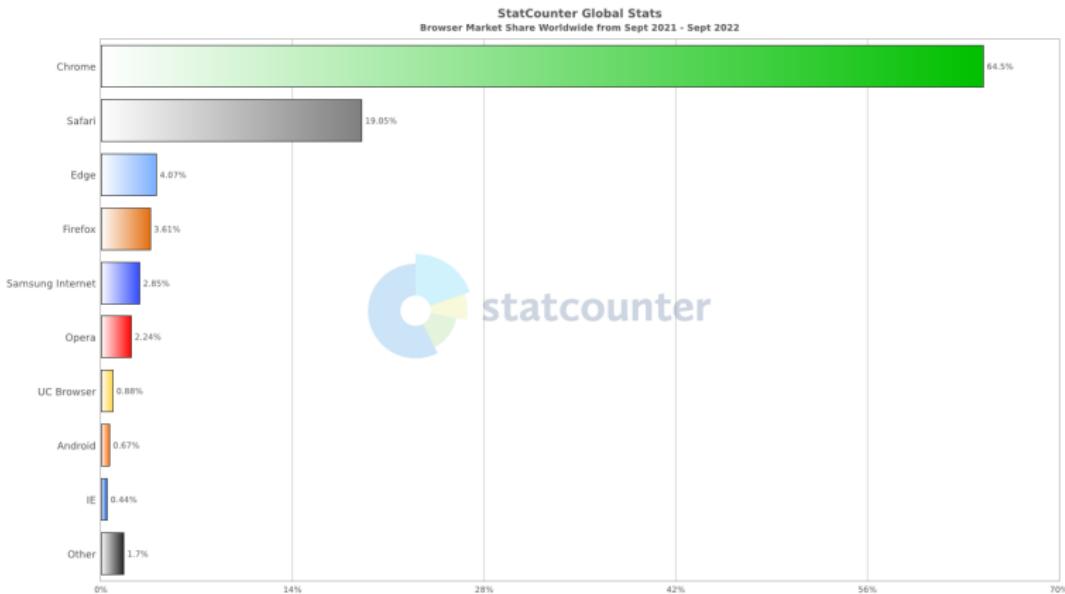
Clientes (Navegadores ou Browsers)

- Software que:
 - ▶ **Pede recursos** ao servidor
 - ▶ Recebe a resposta do servidor e **processa/renderiza** o recurso Web
- Primeiro navegador gráfico: mosaic (1993)
- Principais browsers:
 - ▶ Netscape navigator (1994)
 - ▶ Microsoft Internet explorer (1995)
 - ▶ Opera (1995)
 - ▶ Mozilla Firefox (2002)
 - ▶ Apple safari (2003)
 - ▶ Google chrome (2008)
 - ▶ Microsoft edge (2015)

MODELO CLIENTE-SERVIDOR

Clientes (Navegadores ou Browsers)

- Browser Market Share Worldwide Sept 2021 to Sept 2022²



²<https://gs.statcounter.com/browser-market-share>

MODELO CLIENTE-SERVIDOR

Servidores

- Software que:
 - ▶ **Aceita pedidos** HTTP de clientes
 - ▶ **Fornece recursos** solicitados como páginas Web
- Primeiro servidor Web: CERN httpd (1990)
- Principais servidores:
 - ▶ Apache (1995)
 - ▶ Microsoft IIS (1995)
 - ▶ nginx (2004)
 - ▶ Node.js (2009)

MODELO CLIENTE-SERVIDOR

Servidores

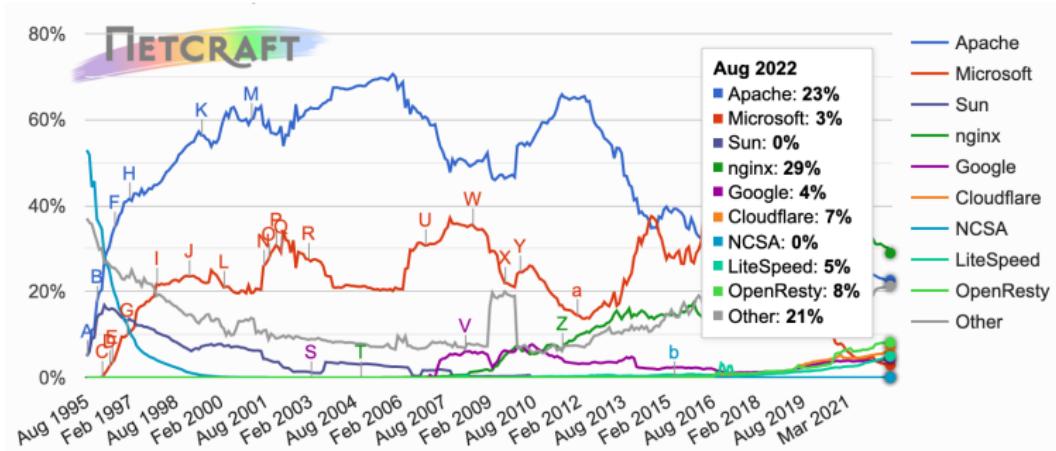
- O primeiro servidor Web (CERN httpd), usado por Tim Berners-Lee no CERN.



MODELO CLIENTE-SERVIDOR

Servidores

- Web server developers: market share of all sites³



³<https://news.netcraft.com/archives/category/web-server-survey/>

MODELO CLIENTE-SERVIDOR

URL (Uniform Resource Locator)

- **Endereço único** que refere a localização de um recurso na Web

Nas redes TCP/IP, um URL possui a seguinte estrutura

esquema://domínio:porta/caminho/recurso?query_string#fragmento

- **Esquema** - protocolo, poderá ser HTTP, HTTPS, FTP, etc.
- **Domínio** - endereço da máquina, designa o servidor que disponibiliza o documento ou recurso solicitado (pode também ser um endereço IP)
- **Porta** - ponto lógico de conexão com o servidor (por omissão, é a porta 80 para HTTP)
- **Caminho** - especifica o local (geralmente num sistema de ficheiros) onde se encontra o recurso, dentro do servidor.
- **Query string** - conjunto de um ou mais pares "nome-valor" enviado ao servidor para filtrar um recurso (opcional)
- **Fragmento** - parte ou posição específica dentro do recurso (opcional)

MODELO CLIENTE-SERVIDOR

URL (Uniform Resource Locator)

Estrutura de um URL

esquema://dominio:porta/caminho/recurso?query_string#fragmento

Exemplo de um URL

<https://www.esmad.ipp.pt/estudantes/horarios/tsiw-2>

■ Caracteres especiais:

- ▶ Não pode conter espaços. Para "emular" o espaço, usa-se o símbolo %XX, sendo XX o valor hexadecimal (20 para espaço).
Exemplo: San%20Jose = San Jose
- ▶ Para tratamento de outros símbolos, veja W3C URI Overview⁴.
- ▶ No protocolo HTTP, quando o ficheiro não é especificado, o servidor retorna o ficheiro index.html (se houver).

⁴http://www.w3.org/Addressing/URL/URI_Overview.html

MODELO CLIENTE-SERVIDOR

Esquema no URL

- Para conexões HTTP o esquema pode ser HTTP ou HTTPS
- Hypertext Transfer Protocol Secure (HTTPS) é apenas HTTP em cima de protocolo SSL/TLS

Esquema

http://

MODELO CLIENTE-SERVIDOR

Hostname

Nome registrado (domínio)

www.google.com

Endereço IP

127.0.0.1

MODELO CLIENTE-SERVIDOR

Porta

- A porta por omissão para um servidor **HTTP** é a porta **80**
- Outras portas normalmente utilizadas: 8080, 8000.
- A porta por omissão para um servidor **HTTPS** é a porta **443**
- O número da porta pode ser omitido do URL se for a porta por omissão

Porta HTTP

:80

MODELO CLIENTE-SERVIDOR

Query String

- A query string contém **dados** a serem passados para o software em execução no servidor
- Ela pode conter pares de **nome/valor** separados por 'e comercial' (&)

Exemplo de query string

```
?first_name=John&last_name=Doe
```

MODELO CLIENTE-SERVIDOR

Fragmento

- O identificador de fragmento, se presente, especifica uma **parte** ou uma **posição** dentro do recurso ou documento geral.
- Se usado com HTML, representa um elemento da página identificado pelo seu id.

Exemplo de fragmento

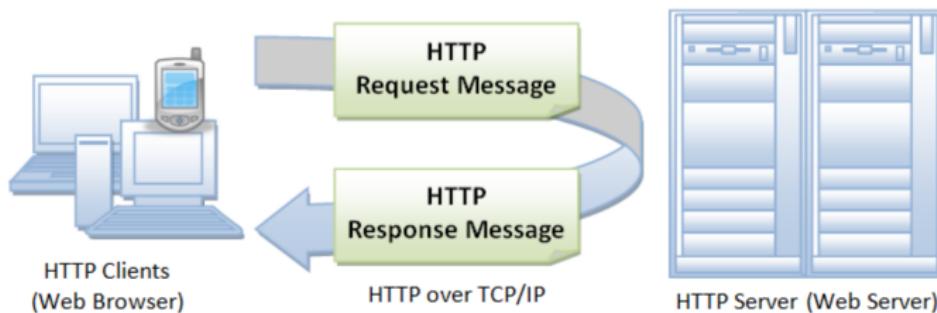
```
#content
```

PROTOCOLO HTTP

PROTOCOLO HTTP

Hypertext Transfer Protocol (HTTP)

- Protocolo de Transferência de Hipertexto na WWW
- Usa TCP/IP
- O protocolo HTTP consiste de duas fases:
 - ▶ Request (pedido)
 - ▶ Response (resposta)



PROTOCOLO HTTP

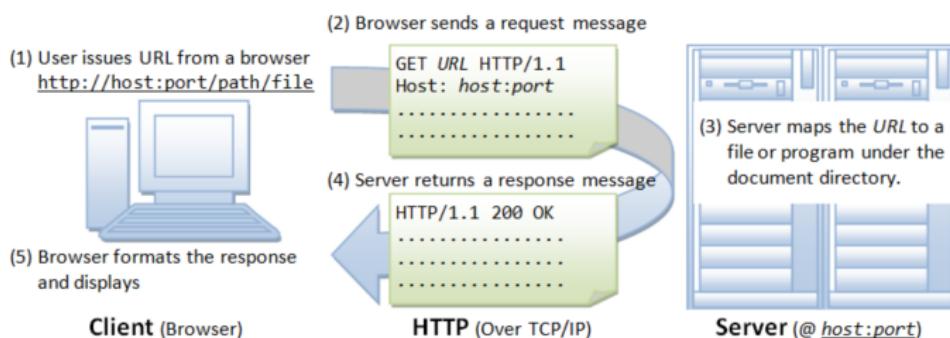
História

- **HTTP/0.9** (1991) Apenas método **GET**
- **HTTP/1.0** (1996) Ficheiros de diferentes tipos. Métodos **HEAD** e **POST**.
- **HTTP/1.1** (1997) Conexões persistentes. **Host** header.
- **HTTP/2.0** (2015) Revisão importante do protocolo de rede HTTP
- **HTTP/3.0** (2018) HTTP sobre QUIC, e não sobre TCP como as restantes versões (um protocolo de rede da camada de transporte projetado por Jim Roskind na Google).

PROTOCOLO HTTP

Funcionamento

- A mensagem entre cliente e servidor consiste de duas partes:
 - ▶ header (info sobre a mensagem)
 - ▶ body (info sobre os dados)



PROTOCOLO HTTP

Pedido HTTP

- Formato geral de um pedido HTTP:
 - ▶ Método HTTP
 - ▶ Campos de cabeçalho
 - ▶ Linha em branco
 - ▶ Corpo da mensagem (opcional)
- Principais métodos HTTP:
 - ▶ **GET** - Retorna o conteúdo do documento especificado
 - ▶ **HEAD** - Retorna o cabeçalho do documento especificado
 - ▶ **POST** - Executa o documento especificado com dados contidos
 - ▶ **PUT** - Substitui documento especificado com os dados contidos
 - ▶ **DELETE** - Remove o documento especificado

PROTOCOLO HTTP

Pedido HTTP

Exemplo de um pedido GET

```
GET /hello.htm HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.tutorialspoint.com
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
```

Exemplo de um pedido POST

```
POST /cgi-bin/process.cgi HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.tutorialspoint.com
Content-Type: application/x-www-form-urlencoded
Content-Length: length
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive

licenseID=string&content=string&/paramsXML=string
```

PROTOCOLO HTTP

Resposta HTTP

- Corresponde à mensagem de resposta do servidor
- Formato geral de uma mensagem de resposta:
 - ▶ Linha de estado (Status Line)
 - ▶ Campos de cabeçalho
 - ▶ Linha em branco
 - ▶ Corpo da resposta

```
HTTP/1.1 200 OK
Date: Sun, 02 Oct 2016 22:18:51 GMT
Server: Apache/2.2.14 (Mandriva Linux /PREFORK-1.4mdv2010.0)
Last Modified: Tue, 07 Jun 2016 10:09:58 GMT
ETag: "4e42b3-42d4-534ad642d0354"
Accept-Ranges : bytes
Content-Length : 17108
Connection: close
Content-Type: text/html

<html>
<head><title>RESEARCH</title></head>
<body>
<p>This is my research page.</p>
<p>...</p>
</body>
</html>
```

PROTOCOLO HTTP

Resposta HTTP

- No exemplo anterior, a linha de estado é HTTP/1.1 200 OK
- Faixa de valores:
 - ▶ 100-199: Informativo (em processamento)
 - ▶ 200-299: Sucesso
 - ▶ 300-399: Redirecionamento
 - ▶ 400-499: Erro do cliente
 - ▶ 500-599: Erro no servidor
- Alguns retornos comuns:
 - ▶ 200: Ok
 - ▶ 404: Not found

PROTOCOLO HTTP

Verificar mensagem HTTP

- **curl** - command line tool and library for transferring data with URLs⁵



⁵<https://curl.se/>

PROTOCOLO HTTP

Exemplos de comandos curl

-v para modo verbose⁶, o comando irá explicar e mostrar de forma mais exaustiva o seu procedimento

```
curl -v https://www.esmad.ipp.pt
```

Descartar o corpo da mensagem com "> /dev/null"

```
curl -v http://www.google.com > /dev/null
```

⁶<https://everything.curl.dev/usingcurl/verbose>

PROTOCOLO HTTP

Verificar mensagem HTTP

■ Chrome Developer Tools

The screenshot shows the homepage of the P.PORTO website. At the top, there's a navigation bar with links for 'esmad', 'cursos', 'estudantes', 'investigação', 'internacional', and 'comunidade'. Below the navigation is a large 'BEM VINDO' banner with the text 'A ESCOLA SUPERIOR DE MÉDIA ARTES E DESIGN' and 'UNIVERSIDADE PÚBLICA'. A search bar at the bottom has the placeholder 'Exemplo: design, multimedia, fotografia ...' and a magnifying glass icon.

The screenshot shows the 'Network' tab in the Chrome Developer Tools. It lists several requests made by the browser to load the page. One request is highlighted in blue, showing the following details:

- Request URL:** `http://www.esmad.pt/`
- Status Code:** `200 OK`
- Response Headers:**
 - Accept-Ranges: bytes
 - Content-Type: text/html; charset=UTF-8
 - Date: Sat, 01 Jun 2014 09:40:15 GMT
 - Expires: Sat, 01 Jun 2014 09:40:16 GMT
 - Last-Modified: Fri, 28 May 2014 10:00:00 GMT
 - Pragma: No-cache
 - Server: Apache/2.2.22 (Ubuntu)
 - Vary: Accept-Encoding
 - Content-Length: 1136
 - Content-Encoding: gzip
 - Content-Language: pt
 - Content-Type: text/html; charset=UTF-8
- Response Body:** (Large binary file, 1136 bytes)

PROTOCOLO HTTP

Verificar mensagem HTTP

■ Chrome Developer Tools

1. Abrir developer tools no browser (Three dots or ellipsis, more tools, developer tools)
2. Ver network
3. Ver headers e response do documento

The screenshot shows the Network tab in the Chrome DevTools. At the top, there's a timeline with various requests listed by name. Below the timeline, a specific request for 'esmad.ipp.pt' is expanded. The 'Response' tab is selected, showing the raw HTTP response content. The response starts with the XML declaration and includes meta tags, a base href, and a large block of text at the bottom.

Name	Headers	Preview	Response	Initiator	Timing	Cookies
esmad.ipp.pt			1 <!DOCTYPE html>			
www.esmad.ipp.pt			4 <html xmlns="http://www.w3.org/1999/xhtml" lang="pt">			
reset-cachekey-70f64846e5cae1414276b561efcdab.css			5 <head>			
base-cachekey-5a6563307c0db28a477bdc30223a21.css			6 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />			
resourcecollective.customfooterstylesheetmain-cachekey-3			7 <title>Esmad - Escola Superior de Media Artes e Design</title>			
phoneCustom-cachekey-9a19d3f5ff816bf243d99fb5ecabbf78			8 <script>function() {</script>			
quantcast-cachekey-a13153e731ab73c0509688f252d2da8.js			9 <script>var quantcast = quantcast {};			
resourceonline.app.jquery-cachekey-781361d7d75755636a2.js			10 <script>\$(document).ready(function() {</script>			
resourceapp.theme.jmain-cachekey-0e2590b60fa8a179af.js			11 \$(document).on('click', '.dropdown-item', function(e) {</script>			
resourcecollective.sliders.bosilder.jquery.owl.carousel-cachekey-3			12 \$(document).on('click', '.owl-item', function(e) {</script>			
collective.jquery.custom.min-cachekey-218c0ed496e91.js			13 \$(document).on('click', '.owl-item', function(e) {</script>			
resourceapp.theme.jquery.autocomplete.html-cachekey-1.js			14 \$(document).on('click', '.owl-item', function(e) {</script>			
resourcejquery.prettyPhoto-cachekey-a665aa9e58db03076.js			15 \$(document).on('click', '.owl-item', function(e) {</script>			
resourceapp.theme.jquery.multiple.select-cachekey-40228fe.js			16 \$(document).on('click', '.owl-item', function(e) {</script>			
resourceonline.formwidget.recaptcha.ajax-cachekey-3			17 \$(document).on('click', '.owl-item', function(e) {</script>			
resourceapp.theme.jquery.freewall-cachekey-a05837e18fb44a30d.js			18 \$(document).on('click', '.owl-item', function(e) {</script>			
resourcedrop_down-cachekey-915ab748f0b449bb58ef643a.js			19 \$(document).on('click', '.owl-item', function(e) {</script>			

TECNOLOGIAS

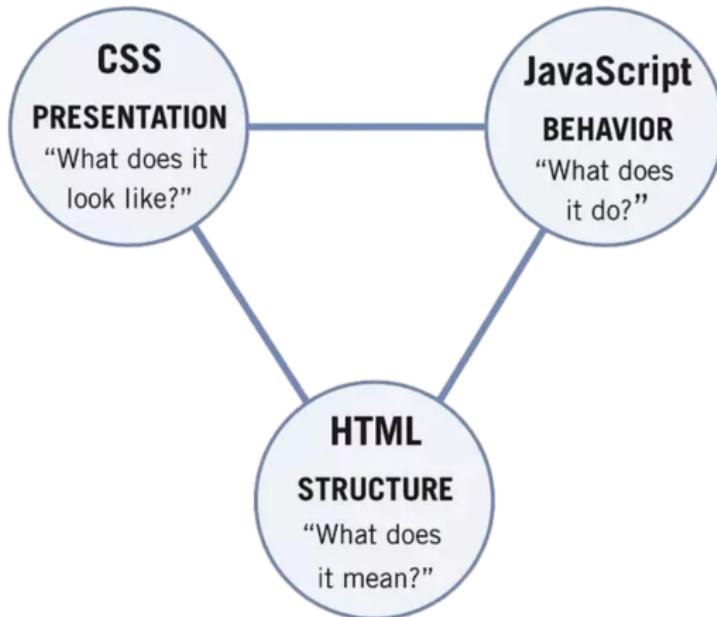
TECNOLOGIAS

Tecnologias Web

- **HTML** (Hypertext Markup Language) - Estruturação/Apresentação de dados
- **CSS** (Cascading Style Sheets) - Estilização de dados
- **JavaScript** - Interação de dados



TECNOLOGIAS



TECNOLOGIAS



JavaScript

```
<script type="text/javascript">
```

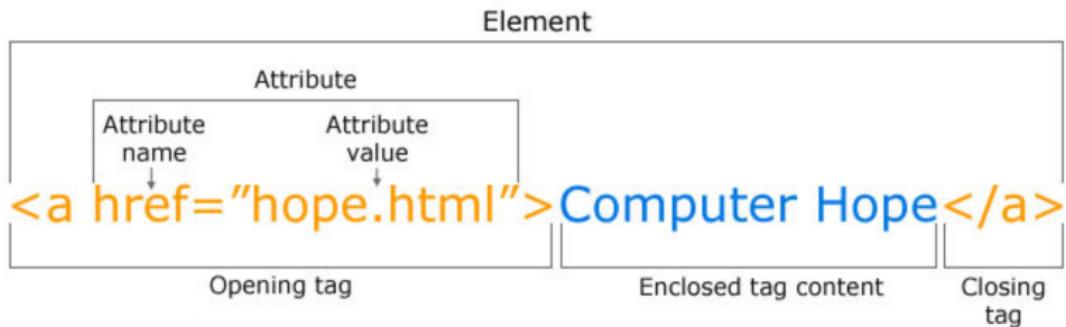
CSS

```
<style>
Body {
  color:...
ul#mylist {
  font-family
</style>
```

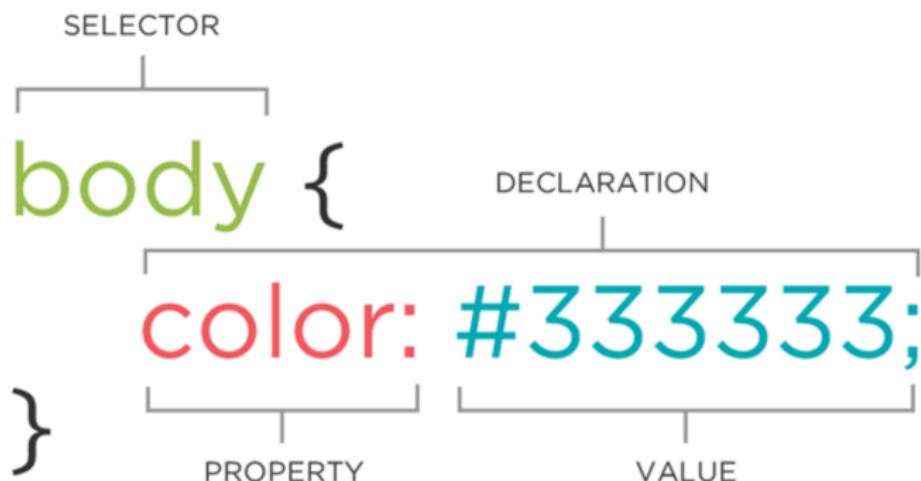
HTML

```
<html>
<head>...
<body>
<h1>
<h2>
<p>...</p>
<ol><li><li>
```

HTML Tag



Seletor CSS

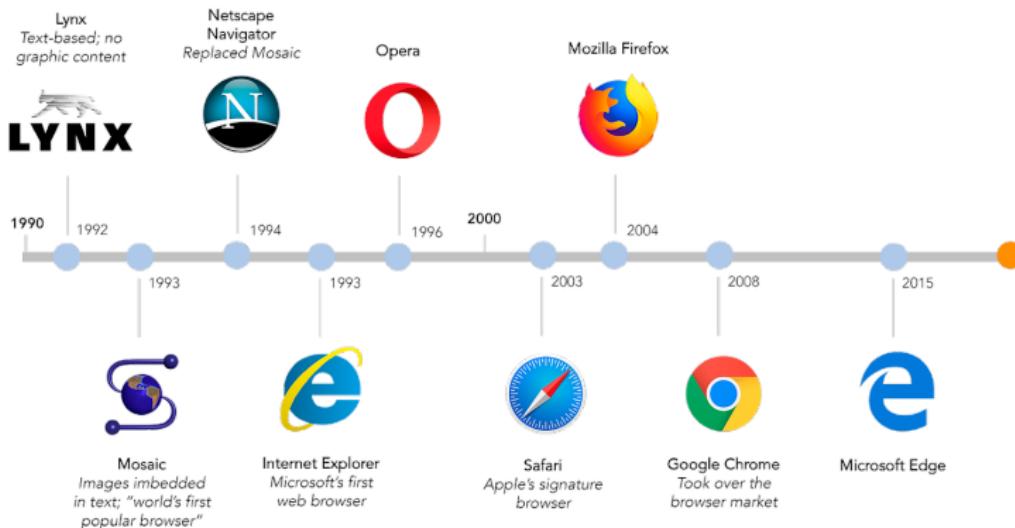


Código JavaScript

```
<script type="text/javascript">
    function getInspiration() {
        if (morningDay == "depressed")
            start.Coding & get.Awesome;
    }
</script>
```

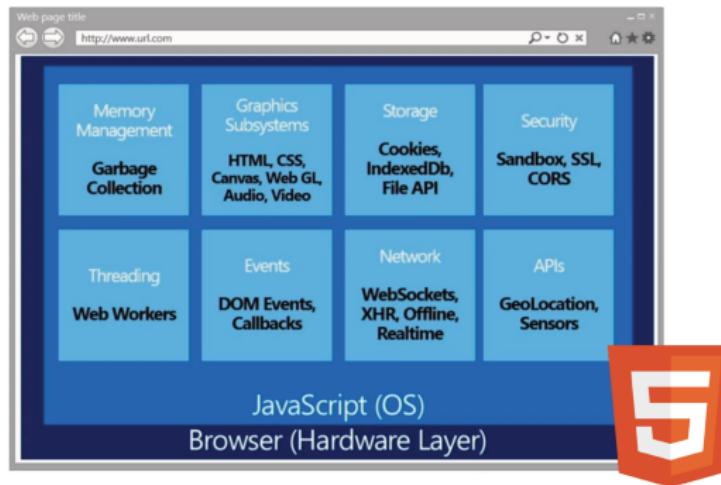
TECNOLOGIAS

Evolução dos Browsers Web



HTML5: Um sistema operativo?

Scott Hanselman Fluent 2014 keynote "Virtual Machines, JavaScript and Assembler" ⁷.



⁷<https://www.youtube.com/watch?v=UzyoT4DziQ4>

"Browsers will become a new OSs, JS will become a new C language, the software will migrate into the browser."⁸

⁸Gary Bernhardt, The Birth & Death of Javascript
(<https://www.destroyallsoftware.com/talks/the-birth-and-death-of-javascript>)

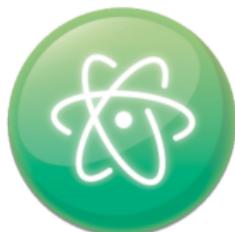
TECNOLOGIAS



Beyond the Browser



Aplicações Desktop baseadas em Electron



Atom



Visual Studio Code



Slack



WhatsApp



Brave



Nylas Mail

Aplicações React Native⁹



Facebook
iOS · Android



Facebook Ads Manager
iOS · Android



Instagram
iOS · Android



F8
iOS · Android

[Using React Native in the Facebook App](#)

How We Built the First Cross-Platform React Native App

React Native at Instagram

Tutorial: Building the F8 conference app



Airbnb
iOS · Android



Walmart
iOS · Android



Tesla
iOS · Android



Tencent QQ
Android

[Hybrid React Native Apps at Airbnb](#)

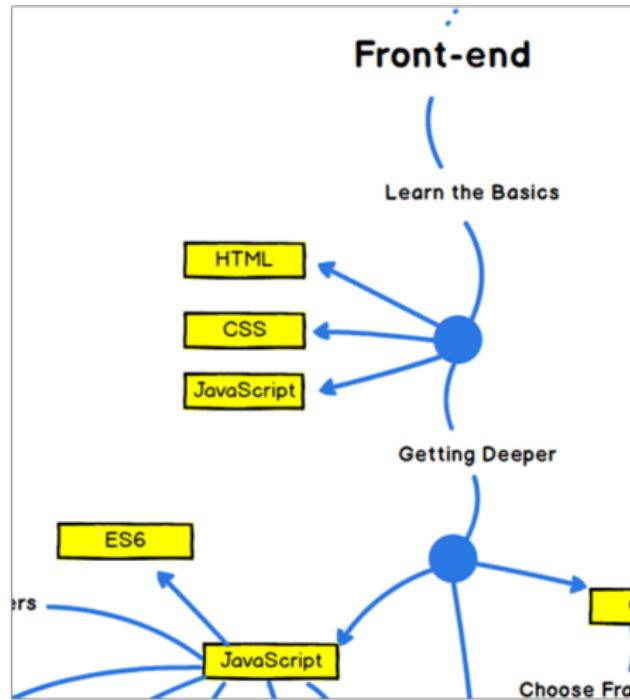
React Native at Walmart Labs

QQ is China's largest messaging platform, with over 829 million active accounts

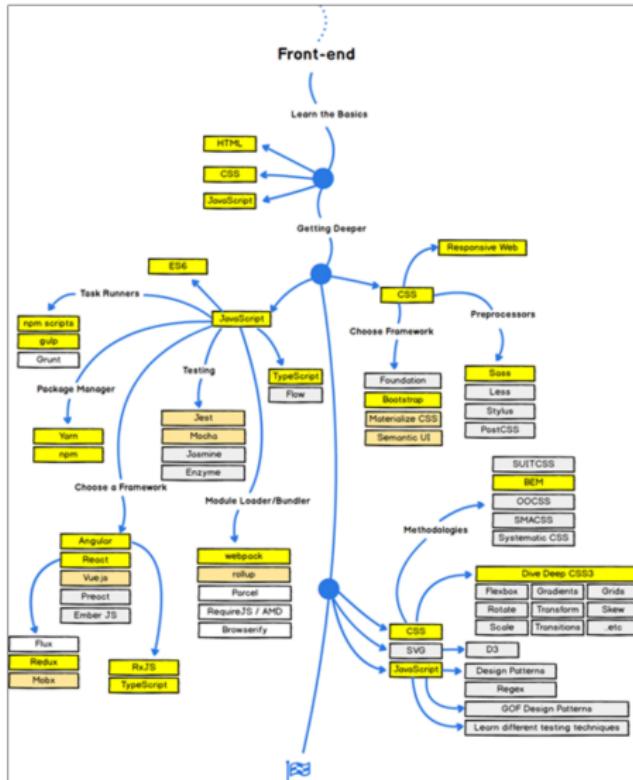
⁹<https://reactnative.dev/showcase>



TECNOLOGIAS



TECNOLOGIAS



VISUAL STUDIO CODE

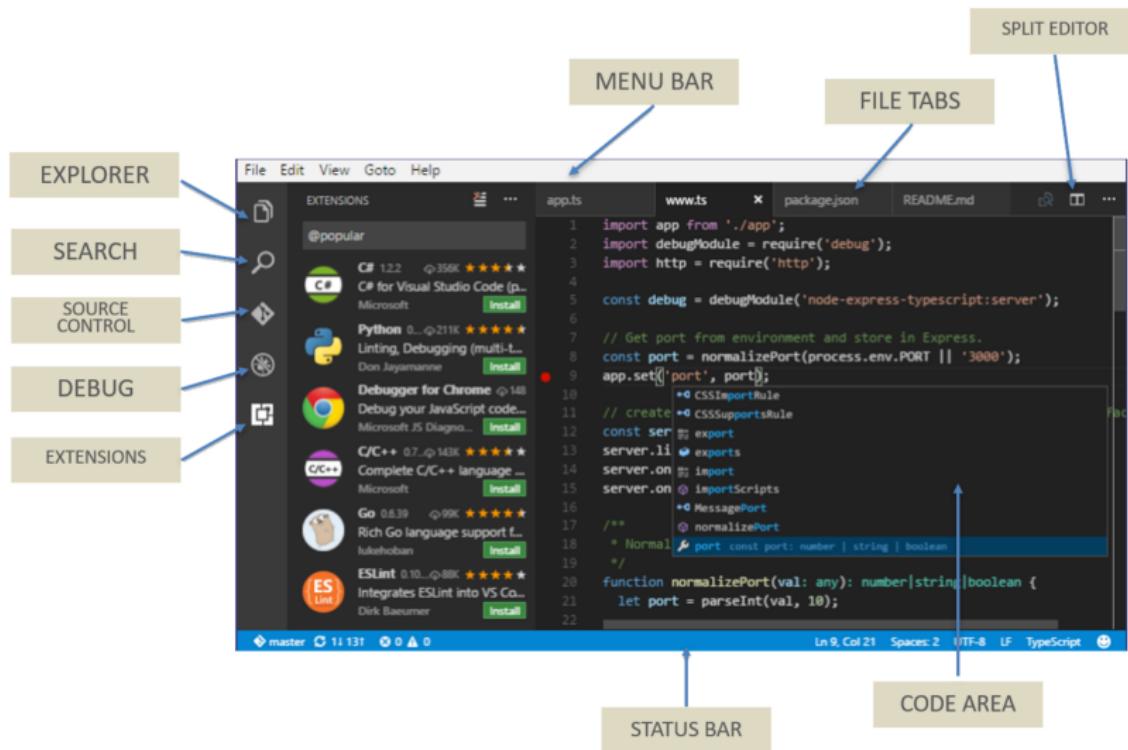
VISUAL STUDIO CODE

Download page for Visual Studio Code¹⁰

The screenshot shows the official Visual Studio Code website at the top, featuring a dark header with navigation links like 'Visual Studio Code', 'Docs', 'Updates', 'Blog', 'API', 'Extensions', 'FAQ', 'Learn', 'Search Docs', and a 'Download' button. Below the header, there's a large promotional image with the text 'Code editing. Redefined.' and 'Free. Built on open source. Runs everywhere.' A prominent blue 'Download Mac Universal Stable Build' button is visible. To the right, a smaller window displays the Visual Studio Code interface. It shows the 'EXTENSIONS MARKETPLACE' tab in the sidebar, listing extensions such as Python, GitLens, C/C++, ESLint, Debugger for Chrome, Language Support for Java, and vscode-linters. The main editor area contains a snippet of JavaScript code related to blog posts. The bottom status bar shows the file path 'blog-post.js ~ index.js ~ util.js', the terminal output 'Compiling...', and other system information.

¹⁰<https://code.visualstudio.com>

VISUAL STUDIO CODE



VISUAL STUDIO CODE

Extensões recomendadas

- Live Server¹¹
- ESLint¹²
- JavaScript (ES6) code snippets¹³

¹¹[https://marketplace\(...\)?itemName=ritwickdey.LiveServer](https://marketplace(...)?itemName=ritwickdey.LiveServer)

¹²[https://marketplace\(...\)?itemName=dbaeumer.vscode-eslint](https://marketplace(...)?itemName=dbaeumer.vscode-eslint)

¹³[https://marketplace\(...\)?itemName=xabikos.JavaScriptSnippets](https://marketplace(...)?itemName=xabikos.JavaScriptSnippets)

GIT

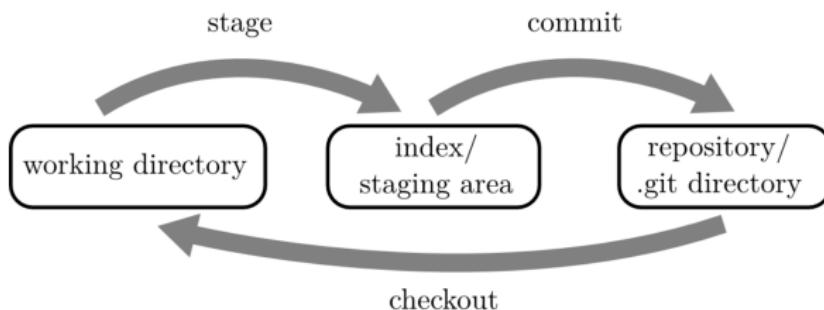
- Sistema de controle de versões distribuído de código fonte
- Gratuito e de código aberto¹⁴
- Inicialmente projetado por Linus Torvalds para o desenvolvimento do kernel Linux
- Posteriormente adotado por muitos outros projetos



¹⁴<https://git-scm.com>

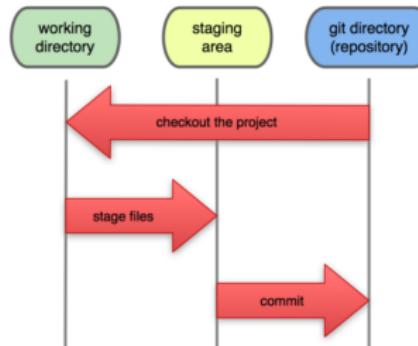
■ Estados locais de ficheiros

- ▶ **Diretório de trabalho:** único checkout de uma versão do projeto, os ficheiros são obtidos a partir da base de dados comprimida no diretório do Git e colocados em disco para uso.
- ▶ **Área de preparação:** simples ficheiro com dados sobre o que irá no próximo commit
- ▶ **Diretório do Git:** local onde o Git armazena os metadados e objectos do projeto



Workflow básico

1. **Modificar ficheiros** no diretório de trabalho (**working directory**)
2. **Selecionar ficheiros**, adicionando snapshots para a área de preparação (**staging area**).
3. **Fazer um commit**, que leva os ficheiros como tal como eles estão na área de preparação e os armazena permanentemente no diretório **Git**.



Obter um projeto Git (Linha de comandos)

Uso de um projeto ou diretório existente: inicializar um repositório Git

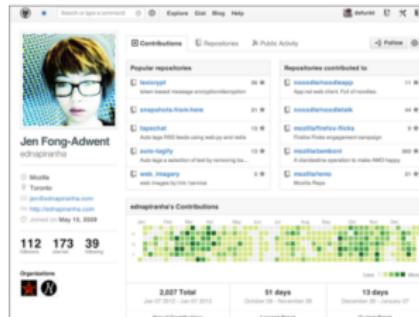
```
// iniciar a monitorização de um projeto existente com Git  
git init  
  
// controlar o versionamento dos ficheiros existentes  
git add *.c  
git add README  
git commit -m 'initial project version'
```

Clonar um repositório Git existente a partir de outro servidor

```
// copiar um repositório Git já existente para uma pasta específica (myProject)  
git clone git://github.com/twbs/bootstrap.git myProject
```

GitHub¹⁵

- Plataforma de hospedagem de código para controle de versões e colaboração
- Usa o sistema de versionamento de projetos Git



¹⁵<https://github.com>

Criação de um projeto VSC e integração via Git

Owner Repository name

PUBLIC  hubot / hello-world 

Great repository names are short and memorable. Need inspiration? How about [petulant-shame](#).

Description (optional)

Just another repository

 Public
Anyone can see this repository. You choose who can commit.

 Private
You choose who can see and commit to this repository.

Initialize this repository with a README
This will allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.

Add .gitignore: **None** | Add a license: **None** ⓘ

Create repository

Criação de um projeto VSC e integração via Git

(1) Instale o Git¹⁶

(2) Crie uma conta no GitHub¹⁷

(3) Leia o tutorial (opcional, 10 minutos)¹⁸

(4) Crie um repositório no **GitHub**

1. Clique em "**New repository**"
2. Dê um nome (ex.: MyRep)
3. Escreva uma breve descrição
4. Selecione "**Initialize this repository with a README**"
5. Clique em "**Create repository**"

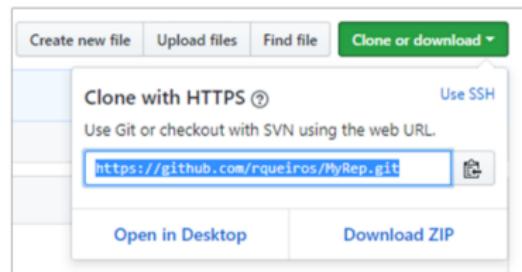
¹⁶<https://git-scm.com/downloads>

¹⁷<https://github.com/join>

¹⁸<https://guides.github.com/activities/hello-world>

Criação de um projeto VSC e integração via Git

(5) Para clonar o repo para o seu computador, clique em "clone or download" e copie o URL



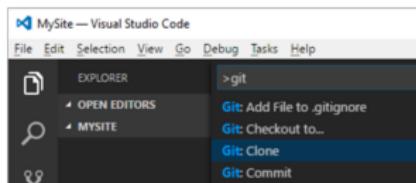
(6) Abra o visual studio code

(7) Selecione **ctrl+shift+p** para abrir o editor de comando do VSC

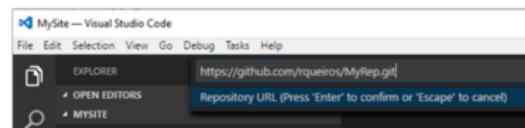
GIT

Criação de um projeto VSC e integração via Git

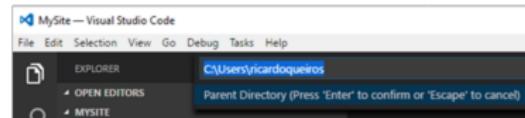
(8) Escreva/selecione git clone



(9) Coloque o url copiado

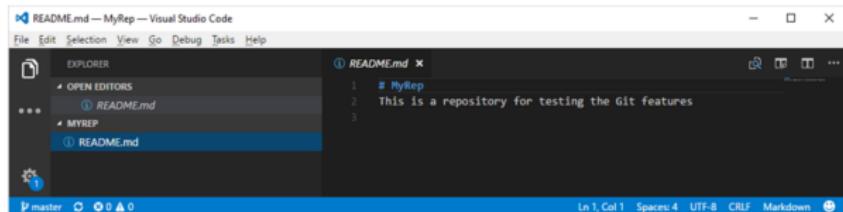


(10) Introduza a localização do projeto e clique em "Open repository"



Criação de um projeto VSC e integração via Git

(11) Criação de repositório em C:\Users\<username>\MyRep



The screenshot shows the Visual Studio Code interface with the following details:

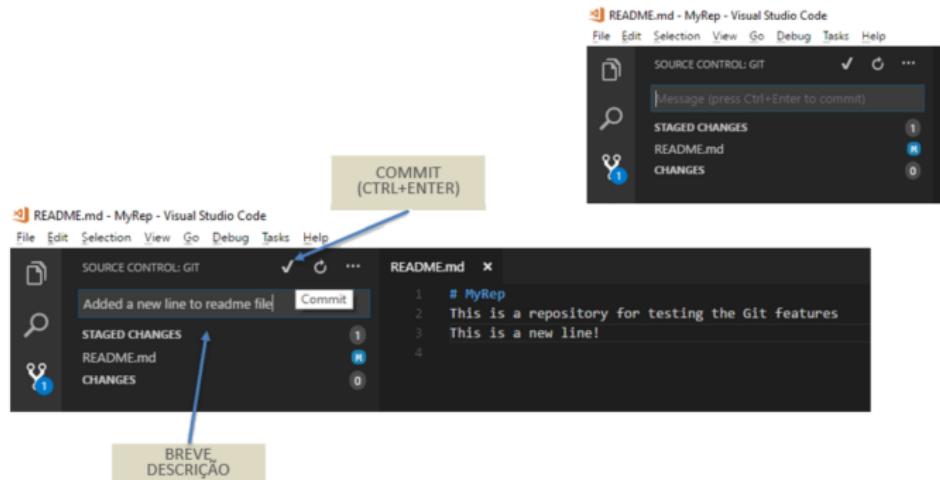
- Title Bar:** README.md — MyRep — Visual Studio Code
- File Menu:** File, Edit, Selection, View, Go, Debug, Tasks, Help
- Explorer Panel:** Shows a tree view with 'OPEN EDITORS' containing 'README.md', and a folder 'MYREP' containing 'README.md'. The 'README.md' under 'MYREP' is currently selected.
- Editor Panel:** Displays the contents of 'README.md':

```
1 # MyRep
2 This is a repository for testing the Git features
3
```
- Bottom Status Bar:** master, 0, 0, 0, Ln 1, Col 1, Spaces: 4, UTF-8, CRLF, Markdown, 56 / 70

(12) Está concluída a integração!

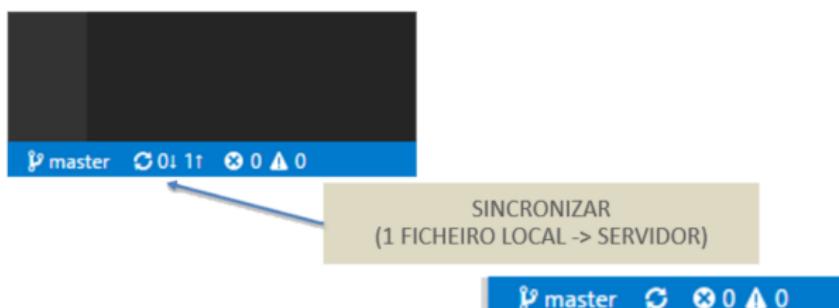
GIT no Visual Studio Code

- Para passar o ficheiro para a área de preparação (staging area) clique em "+"
- Depois pode fazer commit para o diretório git (local)
- Escreva uma breve descrição sobre a alteração em curso



GIT no Visual Studio Code

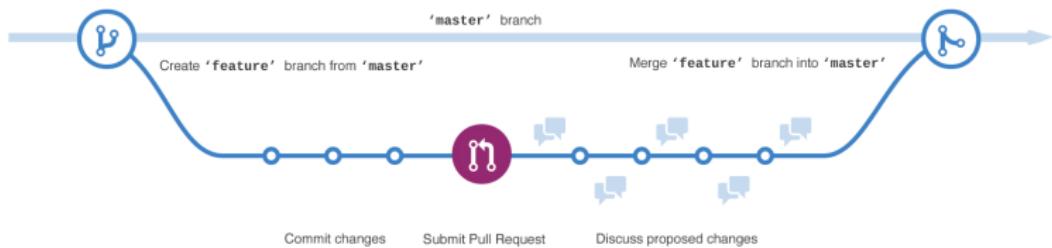
- Após commit, deve sincronizar as alterações com o servidor (GitHub).



- **Master¹⁹ & Branches** são conceitos importantes
 - ▶ Trabalhar em projectos colaborativos
 - ▶ Várias versões em paralelo (DEVELOPMENT VS. PRODUCTION)

¹⁹Poderá também encontrar o termo **Main**

- **Branching** permite trabalhar em diferentes versões de um repositório em simultâneo.
- **O diagrama mostra:**
 - ▶ A branch master
 - ▶ Uma nova branch chamada **feature**
 - ▶ Todo o percurso da **branch feature** até ser **merged** com a **branch master**



Branching básico

Para criar uma nova branch e alternar imediatamente para ela, pode executar o comando git checkout com a parâmetro **-b**

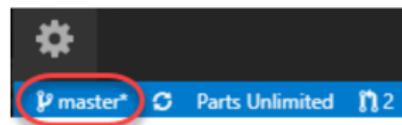
```
git checkout -b iss53
// Switched to a new branch "iss53"
```

Isto é um shorthand para

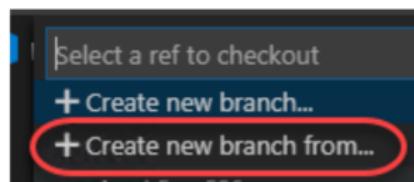
```
git branch iss53
git checkout iss53
// Switched to a new branch "iss53"
```

Criar uma **nova branch** no **Visual Studio Code**

(1) Pressione **master** branch no canto inferior esquerdo



(2) Seleccione **Create new branch from...**

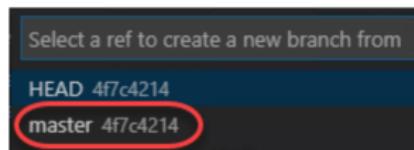


(3) Introduza o nome da nova branch, por exemplo: 'dev'

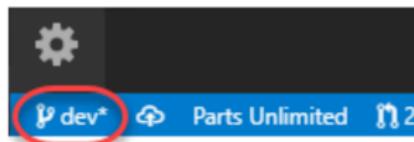


Criar uma **nova branch** no **Visual Studio Code**

(4) Seleccione **master** branch como referência

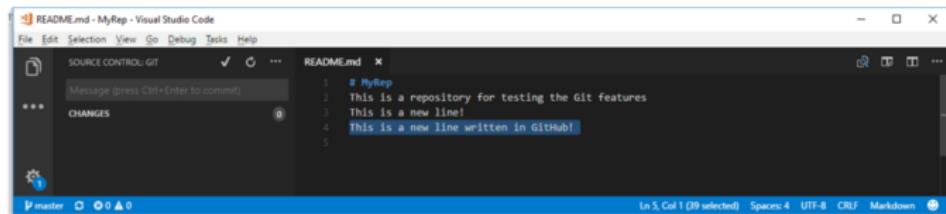


(5) Já se encontra a trabalhar na branch 'dev'



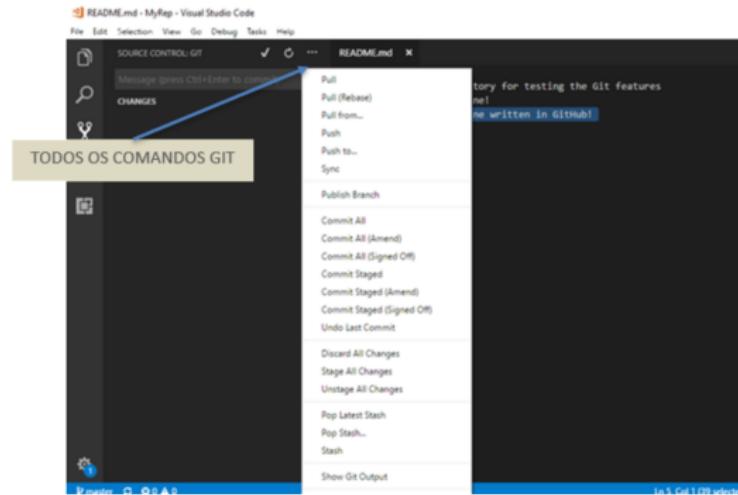
GIT

Se fizer alterações noutro computador ou mesmo no GitHub



GIT

Todos os comandos Git podem ser acedidos clicando em "..."



GIT Bundle²⁰

- Criar, descompactar e manipular ficheiros arquivo.
- Os ficheiros são usados para a transferência offline de objetos Git sem um servidor ativo
- Podem ser usados para criar backups incrementais e completos de um repositório

²⁰<https://git-scm.com/docs/git-bundle>

Exemplo de comando git-bundle

Criar um ficheiro "myrepo.bundle" que tem todas as alterações da branch main até ao último commit

```
git bundle create myrepo.bundle HEAD main
```

INFORMAÇÃO ADICIONAL

INFORMAÇÃO ADICIONAL

Vídeos e Tutoriais

- Google Chrome Developer Tools Crash Course²¹
- Introduction To Visual Studio Code²²
- How to use Git inside of VSCode²³

²¹<https://www.youtube.com/watch?v=x4q86ljFag>

²²https://www.youtube.com/watch?v=4q1tD39Mk_A

²³<https://www.youtube.com/watch?v=VOwyH2-VCVY>

INFORMAÇÃO ADICIONAL

Visual Studio Code

- Visual Studio Code Cheat Sheet²⁴
- Emmet Plugin - Abbreviations²⁵
- Emmet in Visual Studio Code²⁶

²⁴<https://www.git-tower.com/learn/cheat-sheets/vscode>

²⁵<https://docs.emmet.io/abbreviations>

²⁶<https://code.visualstudio.com/docs/editor/emmet>

INFORMAÇÃO ADICIONAL

GIT

- Workflow of Version Control²⁷
- GIT Cheat Sheet²⁸

²⁷<https://www.git-tower.com/learn/cheat-sheets/vcs-workflow>

²⁸<https://www.git-tower.com/learn/cheat-sheets/git>

INFORMAÇÃO ADICIONAL

Windows - Mostrar Ficheiros Escondidos

- Microsoft Support - Show hidden files²⁹

Na linha de comandos (CMD)

dir /ah

²⁹[https://support.microsoft.com/en-us/windows/show-hidden-files-\(...\)db2](https://support.microsoft.com/en-us/windows/show-hidden-files-(...)db2)