

# Git - Controle de versões, Github e Github Pages



### Conteúdo da aula

Para este primeiro momento da aula falaremos sobre **Git** e **Github** e discutiremos um pouco sobre os **sistemas de controle de versões** de código e os **repositórios** de softwares.





Breve pesquisa:

Quem sabe o que é Git e Github?





#### Breve pesquisa:

Git = Github?





# Git e Github não são a mesma coisa!



O Git é um sistema de controle de versão de arquivos.

É um <u>software livre</u> e muito utilizado no desenvolvimento de software onde diversas <u>pessoas estão contribuindo</u> <u>simultaneamente</u>, podendo criar e editar arquivos.



Um sistema de controle de versão ajuda a acompanhar as mudanças feitas no código base. E mais, ele também registra quem efetuou a mudança e permite a restauração do código removido ou modificado.



Sempre quando alguém disponibiliza sua parte do projeto no Git, ele gerencia as alterações feitas e guarda um histórico. Isso é importante pois se houver algum problema você pode desfazer as alterações e voltar para a versão que estava estável.



Projetado e desenvolvido pelo <u>Linus Torvalds</u> para o desenvolvimento do Linux e foi adotado por muitos outros projetos.

Existem sites que provêm <u>hospedagem</u> gratuita de <u>código</u> fonte para repositório Git, <u>um deles é o GitHub</u>.



O <u>Github é um repositório remoto</u>, criado como um serviço online de hospedagem de repositórios do Git.







Funciona como um <u>servidor</u> que agrega todas as <u>modificações realizadas por cada uma das pessoas</u> <u>envolvidas em um projeto</u>, unificando as diferentes versões de código e seus históricos, permitindo compartilhamento entre as equipes.



Além disso, o GitHub também conta com várias <u>integrações</u> com outros serviços online, permitindo o <u>deploy</u> automático de aplicação. Assim, garante a integração contínua e facilita o desenvolvimento.







Como já dito o GIT é um <u>sistema de controle de versões</u>

<u>distribuído</u>, usado principalmente no desenvolvimento de software, mas pode ser usado para registrar o histórico de edições de qualquer tipo de arquivo.



Lançado em 2005 o GIT tem hoje 17 anos de existência e encontra-se na versão 2.42 (lançada em 21/08/2023).





Cada <u>diretório</u> de trabalho do Git é um <u>repositório</u> com um histórico completo e habilidade total de acompanhamento das revisões, <u>não dependente de acesso a uma rede ou a um servidor central</u>.



Um fator importante do git é a possibilidade de criar, a qualquer momento, **snapshots** de um projeto, conhecidos como **branchs**.



Suponha que o seu projeto seja um site html, e você deseja criar uma nova seção no seu código HTML, mas naquele momento você não deseja que estas alterações estejam disponíveis para mais ninguém, só para você.



Isso é, você quer alterar o projeto (incluindo vários arquivos nele), mas ainda não quer que isso seja tratado como "oficial" para outras pessoas, então você cria um <u>branch</u>.



Este branch é como se fosse uma cópia, um espelho do projeto original. Agora você trabalha apenas nesse branch, até acertar todos os detalhes dele. Após isso, você pode fazer um merge de volta do seu branch até o projeto original.

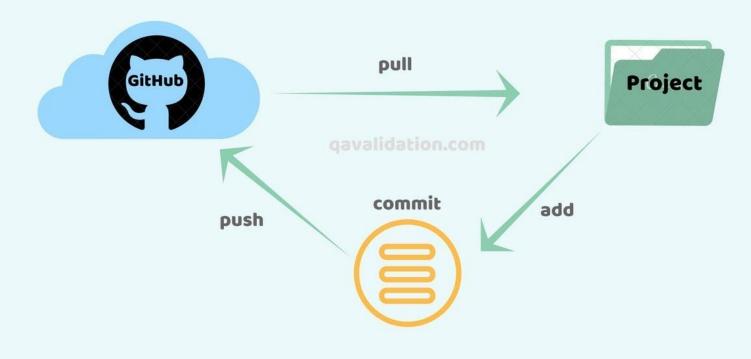


O git possui algumas diretivas (comandos) muito utilizados (e falados) que são o *add*, o *commit*, o *pull* e *push*.

Estes são os comandos mais básicos (e importantes) de se saber.



### Git Push Pull Commands





O GIT possui vários <u>conceitos</u> relacionados à manipulação do código-fonte e suas versões. A seguir abordaremos alguns dos principais.



#### **Repositório**

É o <u>local físico</u> onde os arquivos e suas cópias são armazenados. Pode ser local ou remoto, podendo salvar não apenas arquivos de texto, mas também imagens, áudios e outros elementos relacionados ao projeto.



#### **Branch**

Branches são os <u>ramos</u>, <u>cópias</u> do código original que podem ser manipuladas, sem afetar o código-fonte do projeto original (ou as funcionalidades do ambiente de produção).



#### Merge

Após a finalização de uma alteração em um *Branch*, é necessário realizar o *Merge*, ou seja, fundir a cópia dos arquivos modificados com o ramo principal do projeto. Isso acontece no repositório local para que ajustes possam ser feitos.



#### **Push Request**

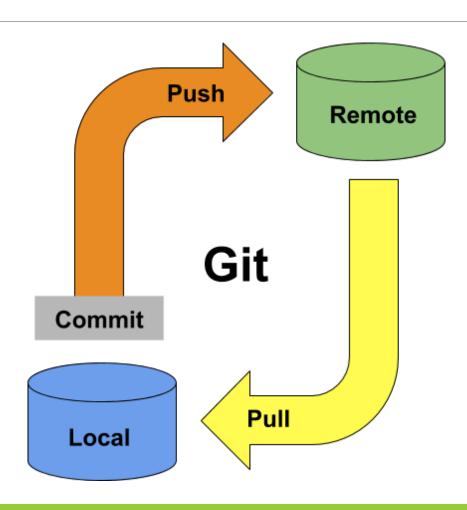
O *Push Request* é o **envio** das modificações após o *Merge* **para o repositório central**, para que todas as outras pessoas que atuam no desenvolvimento possam atualizar suas cópias e revisar o código criado, verificando conflitos com seus próprios trabalhos.



#### **Pull Request**

O *Pull Request* é utilizado quando outra pessoa que atua no desenvolvimento muda o ramo principal no repositório central, **puxando** as modificações realizadas p**ara a sua máquina**, fundindo a nova versão com o seu código local.







#### **Fork**

Comando que copia um repositório remoto para uma máquina local, realizado sempre que vamos começar a trabalhar em um projeto que já existe. Também é usado para pegar um código público para modificação e utilização em um novo projeto.



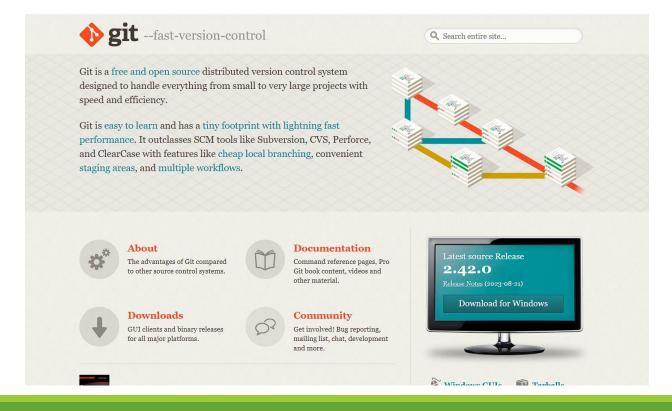
# Instalação



## Instalação

#### Site oficial da ferramenta:

https://git-scm.com/



#### Documentation

#### Downloads

GUI Clients Logos

#### Community

The entire **Pro Git book**written by Scott Chacon and
Ben Straub is available to read
online for free. Dead tree
versions are available on
Amazon.com.

#### Downloads



Older releases are available and the Git source repository is on GitHub.



#### **GUI Clients**

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

View GUI Clients →

#### Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

View Logos →

#### Git via Git

If you already have Git installed, you can get the latest development version via Git itself:

git clone https://github.com/git/git

You can also always browse the current contents of the git repository using the web interface.





## Instalação

Para verificar se o git está devidamente instalado:

Abrir o **Prompt de comandos** ou **terminal linux** 

E digitar o comando: git --version



```
C:\Windows\system32\cmd.exe
Microsoft Windows [versão 10.0.19044.1645]
(c) Microsoft Corporation. Todos os direitos reservados.
C:\Users\epans>git --version
git version 2.35.1.windows.2
C:\Users\epans>
```



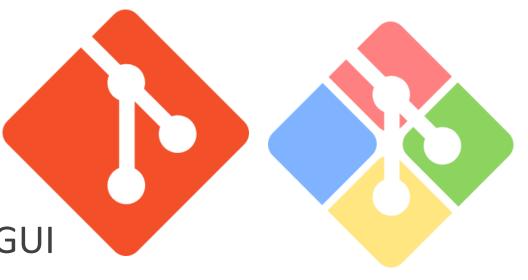
# Instalação

Git for Windows:

https://gitforwindows.org/

Conjunto de ferramentas para o windows que traz o Git BASH, Git GUI

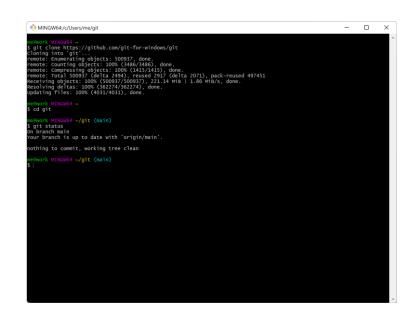
e integração com o Win Explorer (via botão direito).

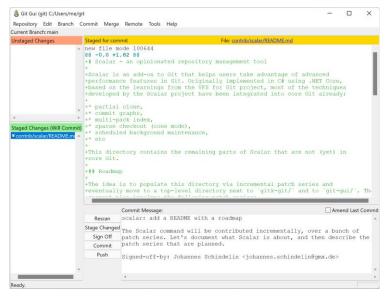


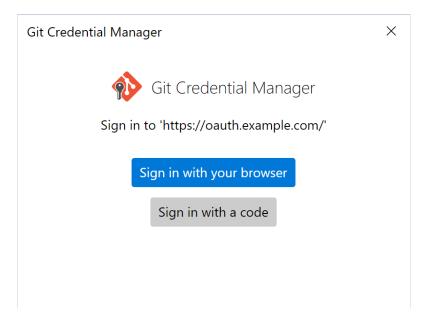


# Instalação

#### Git for Windows:











GitHub



GitHub é uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git. Ele permite que qualquer usuário cadastrado contribua em projetos privados e/ou *Open Source* de qualquer lugar do mundo.



Mascote, Octocat





### E suas várias versões...









GitHub também é <u>amplamente utilizado por</u> <u>programadores para divulgação de seus trabalhos</u> ou para que outros programadores contribuam com o projeto, além de promover fácil comunicação através de recursos que relatam problemas ou mesclam repositórios remotos.



Em outras palavras, o GitHub é uma rede social de desenvolvedores.



Alguns *README's Profiles* legais...

https://github.com/M0nica/M0nica

### Hi, I'm Monica 👏 🧮





Gi

Alg

## **MONICA POWELL**

software engineer, content creator & community organizer



I'm a software engineer who is passionate about making contributing to open-source more approachable, creating technology to elevate people, and building community. Some technologies I enjoy working with include ReactJS, Jamstack (JavaScript, APIs + Markup) and GraphQL. In 2020, I was selected to be an inaugural GitHub Star 💥 based on my involvement in the tech community. My interest in the React ecosystem led me to launch React Robins, a community for women and non-binary ReactJS developers.

#### Find me around the web ①:



- Tinkering with interactions on Codepen
- Sharing updates on LinkedIn







Alguns *README's Profiles* legais...

https://github.com/anuraghazra/anuraghazra

### Hello! I'm Anurag

I do open source:)

I'm a self-taught passionate FrontEnd developer from India IN

#### About me

- FrontEnd Engineer at Razorpay
- Built github-readme-stats, verlyjs and more, 50m+ hits 50K stars on GitHub
- VI love writing TypeScript, and building fun experiments on type-level
- 💬 Ask me about anything here











#### **Anurag Hazra's GitHub Stats**

**☆** Total Stars Earned: 62.3k

Total Commits: 0

Total PRs: 660

Total Issues: 154

Contributed to (last year):



### **Most Used Languages** CSS 2.65%

JavaScript 76.92%

HTML 10.01%

TypeScript 9.91%

Astro 0.06%

GLSL 0.45%

#### Releases

No releases published

#### **Packages**

No packages published

#### Contributors 3



anuraghazra Anurag Hazra



eddiejaoude Eddie Jaoude



github-actions[bot]



Alguns *README's Profiles* legais...

https://github.com/sagar-viradiya/sagar-viradiya

#### README.md

### Hey there 👏



Visitor count

#### About me

I'm Sagar Viradiya. A Google Developer Expert on Android. An Android Engineer at DeliveryHero and also Coorganizer of GDG MAD (Mumbai). Community is 💜. Code blooded animal, 1xEngineer 😛 and 🕭 lover.

I ocassionally blog and tweet about android. Also open sourced few android related stuff.

**양 311** forks

Report repository

Releases

No releases published

**Packages** 

No packages published



Alguns *README's Profiles* legais...

https://github.com/sagar-viradiya/sagar-viradiya

#### **Tim's Community Chess Tournament**

Game is in progress. This is open to ANYONE to play the next move. That's the point. 👏 It's your turn! Move a white (hollow) piece.

	Α	В	С	D	E	F	G	Н
8			<u> </u>					
7	<u>*</u>	<u>*</u>	<u>*</u>		<u>*</u>	<u>*</u>		<u>*</u>
6							<u> </u>	
5				<u> </u>				
4				<u>a</u>		<u>a</u>		
3					<u>a</u>			
2	<u>a</u>	<u>a</u>	<u>a</u>				<u>a</u>	<u>a</u>
		_	А	<mark>የአ</mark> ያ	ھ	A	_	լսո

#### Releases

No releases published

#### **Packages**

No packages published

#### Contributors 7















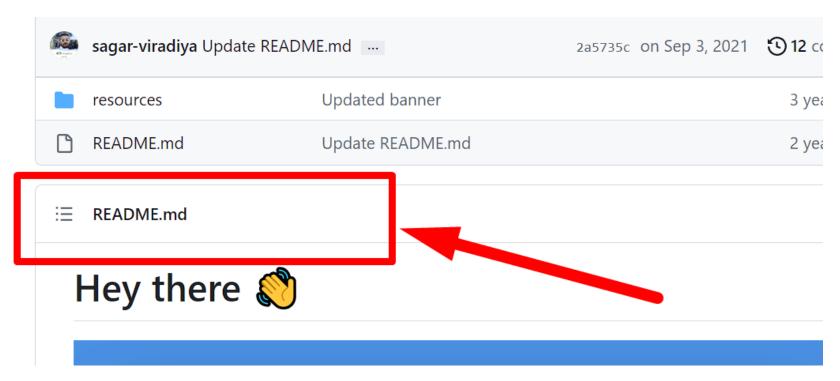
Fonte:

https://dev.to/github/10-standout-github-profile-readmes-

<u>h2o</u>



### O que é README?





### O que é README?

O README é um arquivo com extensão .md, ou seja, ele é escrito em *Markdown* que é uma <u>linguagem de marcação</u> <u>utilizada para converter o texto em um HTML válido</u>.



Nas plataformas de repositórios remotos, como o GitHub, a

função desse arquivo é apresentar informações do projeto,

como:

- Descrição do seu projeto;
- Funcionalidades;



- Como os usuários podem utilizá-lo;
- Onde os usuários podem encontrar ajuda sobre o projeto;
- Autores do projeto.



Por que fazer um README?

O que é bacana de se ter em um README?

Como colocar títulos, imagens, etc...

https://www.alura.com.br/artigos/escrever-bom-readme



Leitura recomendada:

Markdown: como trabalhar com essa linguagem de

markup?



Existem ainda editores que ajudam nessa tarefa, como o

https://stackedit.io/







Welcome file







#### # Welcome to StackEdit!

Hi! I'm your first Markdown file in \*\*StackEdit\*\*. If you want to learn about StackEdit, you can read me. If you want to play with Markdown, you can edit me. Once you have finished with me, you can create new files by opening the \*\*file explorer\*\* on the left corner of the navigation bar.

I π S ≔ ≔ 5 9 0 D

#### # Files

StackEdit stores your files in your browser, which means all your files are automatically saved locally and are accessible \*\*offline!\*\*

#### ## Create files and folders

The file explorer is accessible using the button in left corner of the navigation bar. You can create a new file by clicking the \*\* New file \*\* button in the file explorer. You can also create folders by clicking the \*\* New folder \*\* button.

#### ## Switch to another file

All your files and folders are presented as a tree in the file explorer. You can switch from one to another by clicking a file in the tree.

#### ## Rename a file

You can rename the current file by clicking the file name in the navigation bar or by clicking the \*Rename\*\* button in the file explorer.

#### Welcome to StackEdit!

Hi! I'm your first Markdown file in StackEdit. If you want to learn about StackEdit, you can read me. If you want to play with Markdown, you can edit me. Once you have finished with me, you can create new files by opening the file explorer on the left corner of the navigation bar.

#### **Files**

StackEdit stores your files in your browser, which means all your files are automatically saved locally and are accessible offline!

#### Create files and folders

The file explorer is accessible using the button in left corner of the navigation bar. You can create a new file by clicking the **New file** button in the file explorer. You can also create folders by clicking the New folder button.

#### Switch to another file

All your files and folders are presented as a tree in the file explorer. You can switch from one to another by clicking a file in the tree.



### Ou o <a href="https://dillinger.io/">https://dillinger.io/</a>

















PREVIEW AS -

</>

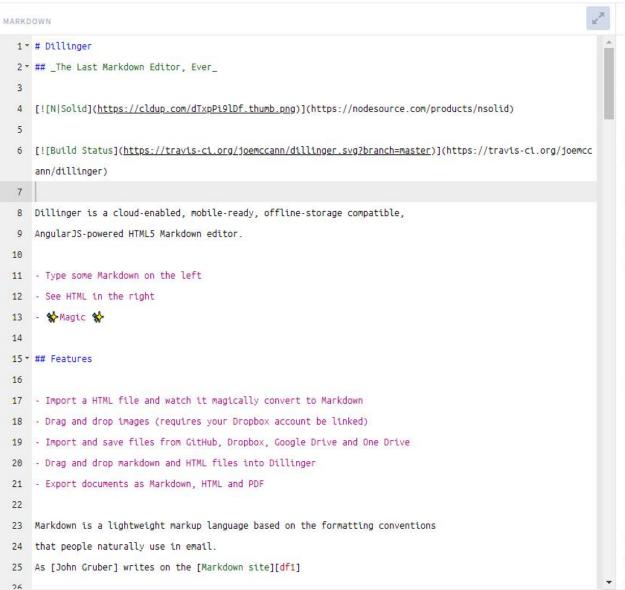
DOCUMENT NAME

READING TIME: 3 MIN READ WORDS: 549

CHARACTERS: 3734

Untitled Document.md

DILLINGER



#### Dillinger

PREVIEW

#### The Last Markdown Editor, Ever



build unknow

Dillinger is a cloud-enabled, mobile-ready, offline-storage compatible, AngularJS-powered HTML5 Markdown editor.

- · Type some Markdown on the left
- · See HTML in the right
- Magic %

#### **Features**

- · Import a HTML file and watch it magically convert to Markdown
- Drag and drop images (requires your Dropbox account be linked)
- · Import and save files from GitHub, Dropbox, Google Drive and One Drive
- · Drag and drop markdown and HTML files into Dillinger
- · Export documents as Markdown, HTML and PDF

Markdown is a lightweight markup language based on the formatting conventions that people naturally use in email.

As John Gruber writes on the Markdown site



Voltando ao assunto principal...



A primeira parte do nome, "Git", é por causa da utilização do sistema de controle de versão e a segunda parte, "Hub", tem a ver com a conexão entre profissionais de programação de qualquer lugar do mundo.



Contando com mais de 73 milhões de desenvolvedores (usuários) e mais de 200 milhões\* de repositórios o Github é sem dúvida uma plataforma importante no contexto do desenvolvimento como um todo.

\*2022 Fonte: <a href="https://en.wikipedia.org/wiki/GitHub">https://en.wikipedia.org/wiki/GitHub</a>



Contando com mais de 100 milhões de desenvolvedores (usuários) e mais de 372 milhões\* de repositórios o Github é sem dúvida uma plataforma importante no contexto do desenvolvimento como um todo.

\*Jan/2023 Fonte: <a href="https://en.wikipedia.org/wiki/GitHub">https://en.wikipedia.org/wiki/GitHub</a>



Em junho de 2018 a Microsoft anunciou a compra (GitHub) da plataforma por US\$ 7,5 bilhões.





Satya Nadella, diretor executivo da Microsoft, reafirmou a postura da empresa frente ao código aberto, destacando **liberdade, a abertura e a inovação**.



### Maiores informações sobre este fato:

https://meiobit.com/385462/microsoft-comprou-github-e-agora

https://g1.globo.com/economia/tecnologia/noticia/microsoft-compra-github-

por-us-75-bilhoes.ghtml

https://diolinux.com.br/noticias/sobre-compra-do-github-pela-microsoft.html



É possível criar uma conta gratuita no Github (apesar de existirem planos pagos) isso é simples e rápido.

https://github.com/

# Where the world **builds software**

Millions of developers and companies build, ship, and maintain their software on GitHub—the largest and most advanced development platform in the world.

Email address

Sign up for GitHub



73+ million

Developers

4+ million **Organizations** 

200+ million

Repositories

84%



Clique em Sign Up e siga os passos...

Ao concluir e se autenticar você terá acesso a sua área e verá todos os seus **repositórios**.



Um repositório contém todos os arquivos do seu projeto e o histórico de revisão de cada arquivo. É possível pode discutir e gerenciar o trabalho de um projeto dentro do repositório.



É possível trabalhar nos repositórios de forma individual ou com outras pessoas, compartilhando a propriedade do repositório.



É possível também restringir quem tem acesso a um repositório, escolhendo a <u>visibilidade</u> de um repositório: público ou privado.



Repositórios públicos podem ser acessados por todos na internet.

Repositórios **privados** só podem ser acessados pelo <u>proprietário</u>, pelas <u>pessoas</u> com as <u>quais você compartilha</u> explicitamente o acesso e, para repositórios de uma <u>organização</u>, por seus integrantes.



#### **Ícones**:



#### **Public**

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



#### Internal

GitHub account members can see this repository. You choose who can commit.





#### **Private**

You choose who can see and commit to this repository.



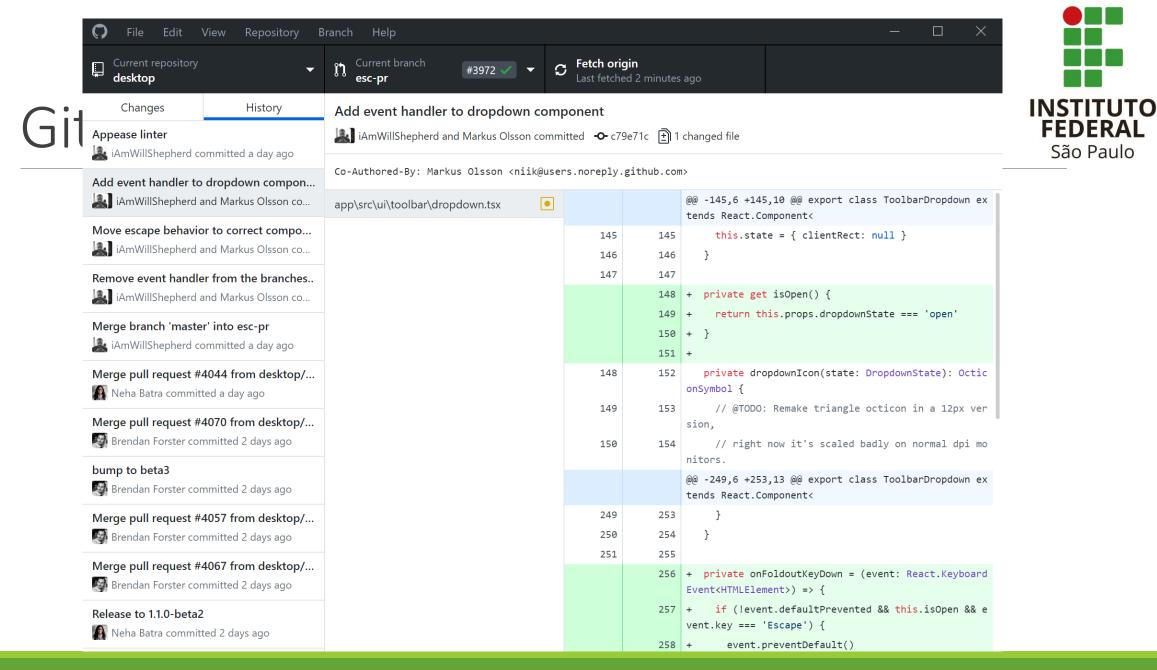




O GitHub Desktop é um aplicativo que permite que você interaja com o GitHub usando uma GUI em vez da linha de comando ou de um navegador web.



Ele é como uma casca para os comandos do Git, escondendo alguns termos que podem ser assustadores para quem está começando. Suas principais vantagens são facilitar a visualização, o envio e recebimento das modificações e a resolução de conflitos (merge).



**FEDERAL** 

São Paulo



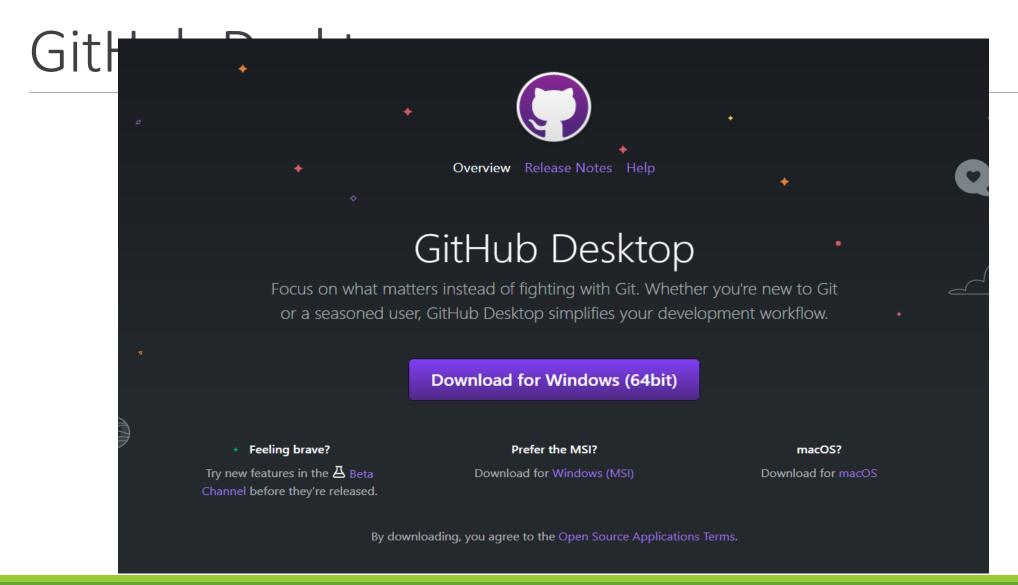
# Instalação



Para instalar o GitHub Desktop deve-se baixar o instalador no site oficial da ferramenta:

https://desktop.github.com/







E em seguida executar o arquivo e aguardar a instalação.

Um passo a passo está disponível na documentação do software:

https://docs.github.com/pt/desktop/installing-and-configuring-github-desktop/installing-and-authenticating-to-github-desktop/installing-github-desktop



Depois de <u>instalado</u> é preciso entrar no GitHub Desktop e se autenticar usando uma conta no Github.

Há várias maneiras de fazer isso, e assim como para instalação da ferramenta há um passo a passo no site oficial.



Autenticar com o GitHub

https://docs.github.com/pt/desktop/installing-and-

configuring-github-desktop/installing-and-authenticating-

to-github-desktop/authenticating-to-github



Há ainda algumas configurações interessantes,

https://docs.github.com/pt/desktop/installing-and-

configuring-github-desktop/configuring-and-customizing-

github-desktop/configuring-a-default-editor



Como por exemplo a escolha do Editor externo padrão.

O Github Desktop é compatível com vários editores, incluindo o Visual Studio Code.



Mais informações sobre o Github Desktop:

https://docs.github.com/pt/desktop

https://www.dio.me/articles/github-desktop-uma-

excelente-aplicacao



# Git no VSCode



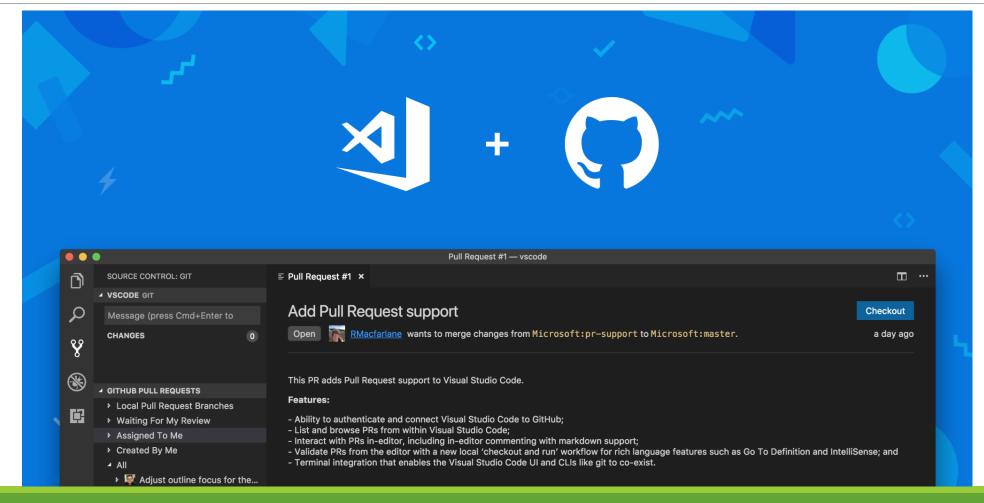
O Visual Studio Code (VS Code) tornou-se <u>um dos editores</u> <u>mais populares</u> disponíveis para o desenvolvimento Web.

Ele é minha escolha pessoal para o desenvolvimento dos exemplos e exercícios de aula.



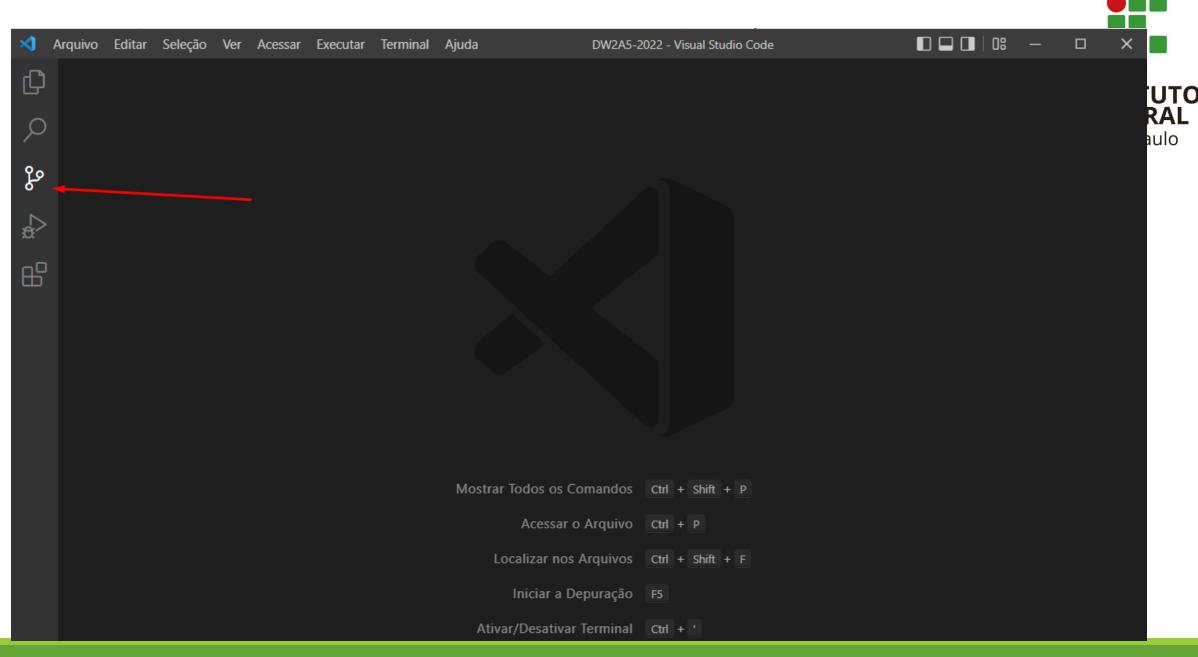
Além de ser possível utilizar o Github Desktop em conjunto o VS Code é possível utilizar somente o VS Code, pois ele possui uma série de funcionalidades integradas para controle de código via Git.



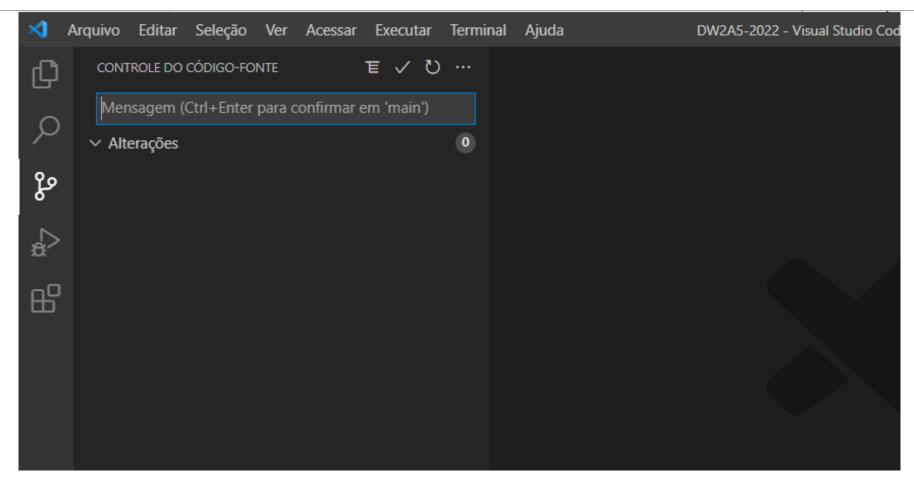




No VS Code há uma área (com um ícone que remete aos ramos, ramificações, *branchs*) que serve para **controle do código fonte**. Nela é possível criar um repositório, dar *push*, *pull* e tudo mais que o Git/Github permitem.

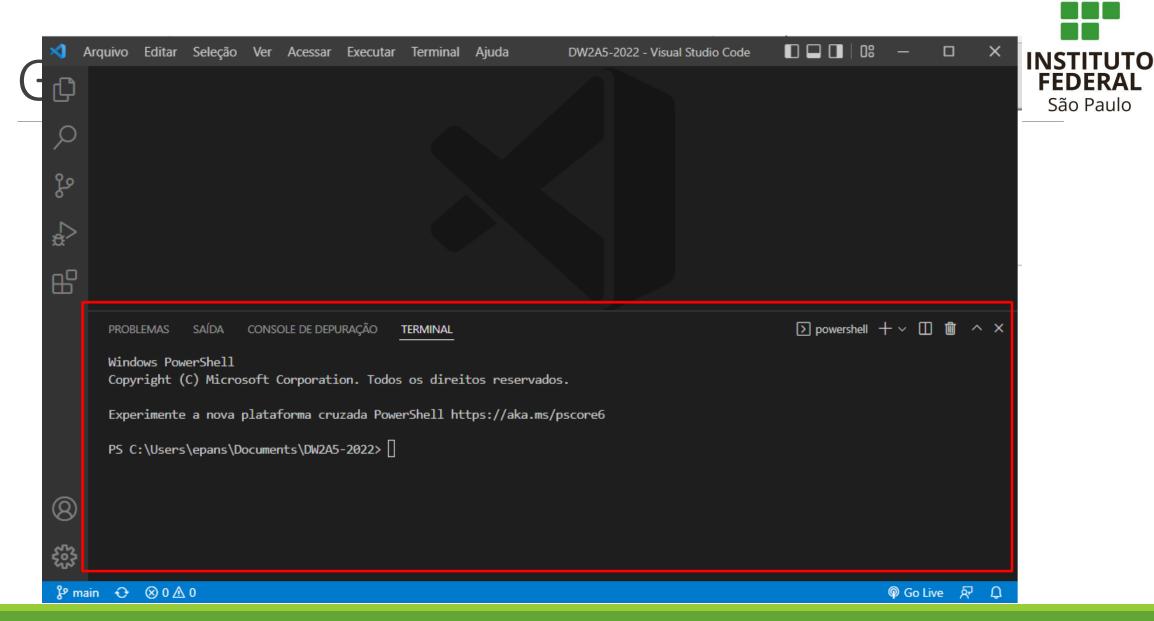


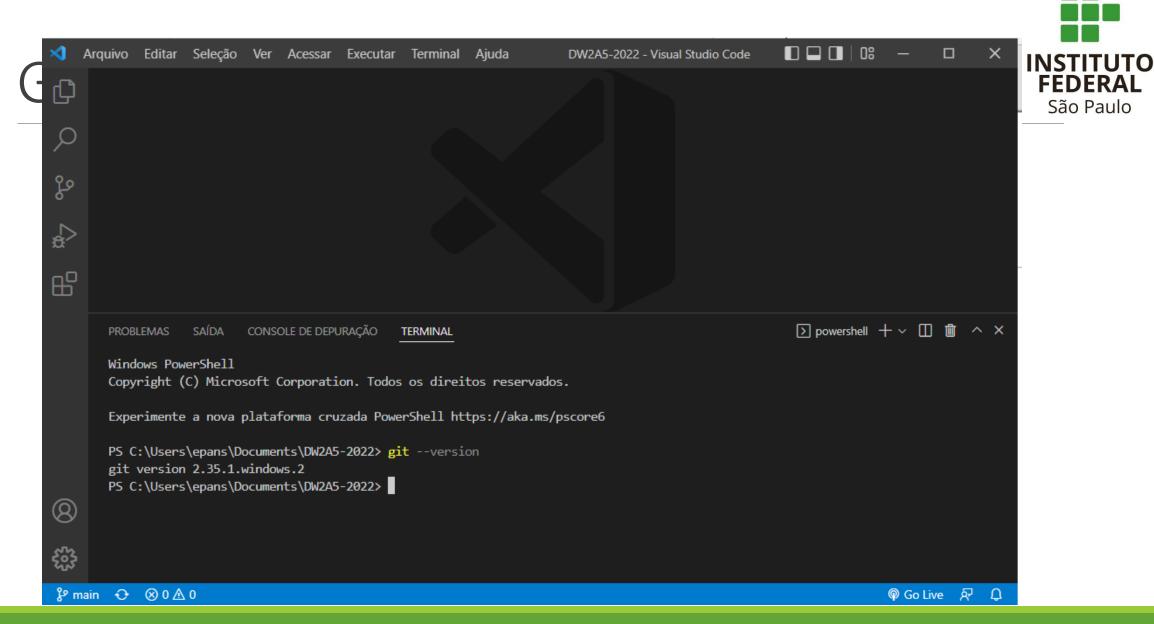






Além disso é possível usando o terminal embutido no VS Code através do atalho Ctrl + \'.







Mais informações sobre o uso do Git no VS Code:

https://docs.microsoft.com/pt-br/learn/modules/use-git-

from-vs-code/



Mais informações sobre o uso do Git no VS Code:

https://www.digitalocean.com/community/tutorials/how-

to-use-git-integration-in-visual-studio-code-pt







O *Github Pages* é um serviço disponibilizado pelo *GitHub* para hospedar sites estáticos com HTML, CSS e JavaScript.

https://pages.github.com/



Com ele é <u>possível</u> <u>hospedar um site</u> gratuitamente sobre

você, sua organização, ou seu projeto diretamente a partir

de um repositório em GitHub.com.



O domínio utilizado é github.io do próprio GitHub, porém é

possível utilizar um domínio personalizado.



Para **publicar um site**, você precisa criar um repositório em sua conta do github e depois ativar e configurar o serviço para um determinado branch (ramo, fonte).



# GitHub Pages

Este processo é bastante simples, um tutorial completo pode ser visto nesse link:

https://docs.github.com/pt/pages/getting-started-with-githubpages/configuring-a-publishing-source-for-your-github-pages-site



# GitHub Pages

É importante lembrar sobre o arquivo inicial (index.html).

Todo website deve conter um arquivo inicial chamado index.html (ou index.php, index.jsp, etc).



# GitHub Pages

O arquivo <u>"index.html" é a página padrão dentro dos diretórios</u> <u>nos servidores de websites</u> que é carregada sempre que uma pasta seja solicitada e não seja especificado o nome de um arquivo específico, neste caso o próprio servidor se encarrega de procurar pelo arquivo "index.html".



https://www.treinaweb.com.br/blog/git-e-github-quais-as-

diferencas

https://blog.betrybe.com/tecnologia/git-e-github/

https://www.alura.com.br/artigos/o-que-e-git-github

https://kenzie.com.br/blog/o-que-e-git/



https://tableless.com.br/tudo-que-voce-queria-saber-sobre-git-e-

github-mas-tinha-vergonha-de-perguntar/

https://pt.wikipedia.org/wiki/Git



Playlist do Curso em Vídeo (módulo 3 capítulo 18)

O que é Git e GitHub? - @Curso em Vídeo HTML5 e CSS3:

https://www.youtube.com/watch?v=GDGMf2bnHlE&list=PL

Hz AreHm4dmcAviDwiGgHbeEJToxbOpZ&index=2



#### Vídeos essenciais:

- 1 O que é Git e GitHub?
- 2 Instalando Git e GitHub no PC
- 3 Criando conta no GitHub
- 4 Primeiro repositório git e GitHub
- 5 Gerenciando Repositórios GitHub



Vamos praticar um pouco com alguns exercícios.





#### Cada aluno deverá criar:

- 1) uma conta no GitHub (caso ainda não tenha)
- 2) um repositório público para a disciplina (eu sugiro algo como a sigla "DW1S4-2023-SeuNome")



3) Preencher o formulário disponível no Google Forms com seu prontuário, nome e link do repositório recém criado.

https://forms.gle/tUg85ZoY19t6NAZYA

■ Atividade GitHub DW1S4 (2023/2 × +



#### Atividade GitHub DW1S4 (2023/2)

Faça login no Google para salvar o que você já preencheu. Saiba mais

\* Indica uma pergunta obrigatória

E-mail \*

Seu e-mail

Prontuário \*

Prontuário ou matrícula, número que começa com "VP".

Sua resposta

Nome \*

Sua resposta

Link do seu repositório da disciplina \*

Algo como https://github.com/epansani/Webl-IFSP



Neste <u>repositório</u> você deverá <u>colocar as atividades feitas</u> <u>em aula</u>, isso ajuda na organização dos arquivos e também serve como uma <u>cópia de segurança das atividades</u> <u>desenvolvidas ao longo do semestre</u>.



E por fim você deverá <u>ativar o GitHub Pages</u> para seu repositório e criar seu Website da disciplina.



#### How to:

https://docs.github.com/pt/pages/getting-started-with-github-pages/configuring-a-publishing-source-for-your-github-pages-site#publishing-from-a-branch

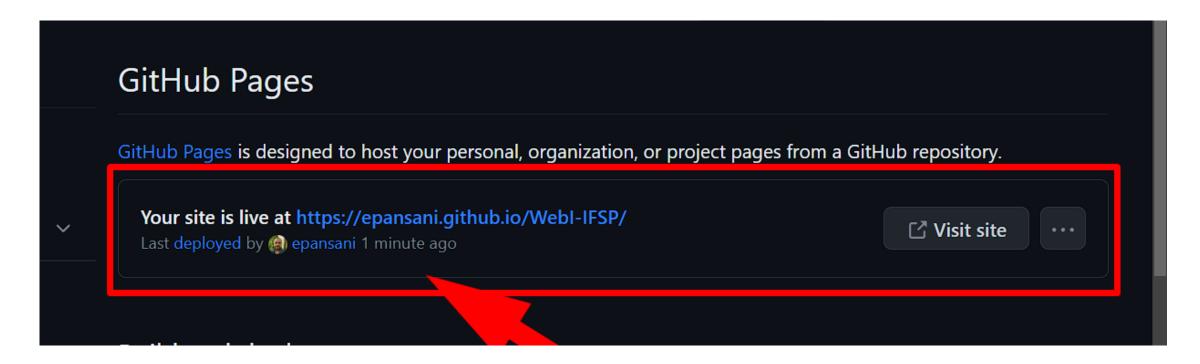


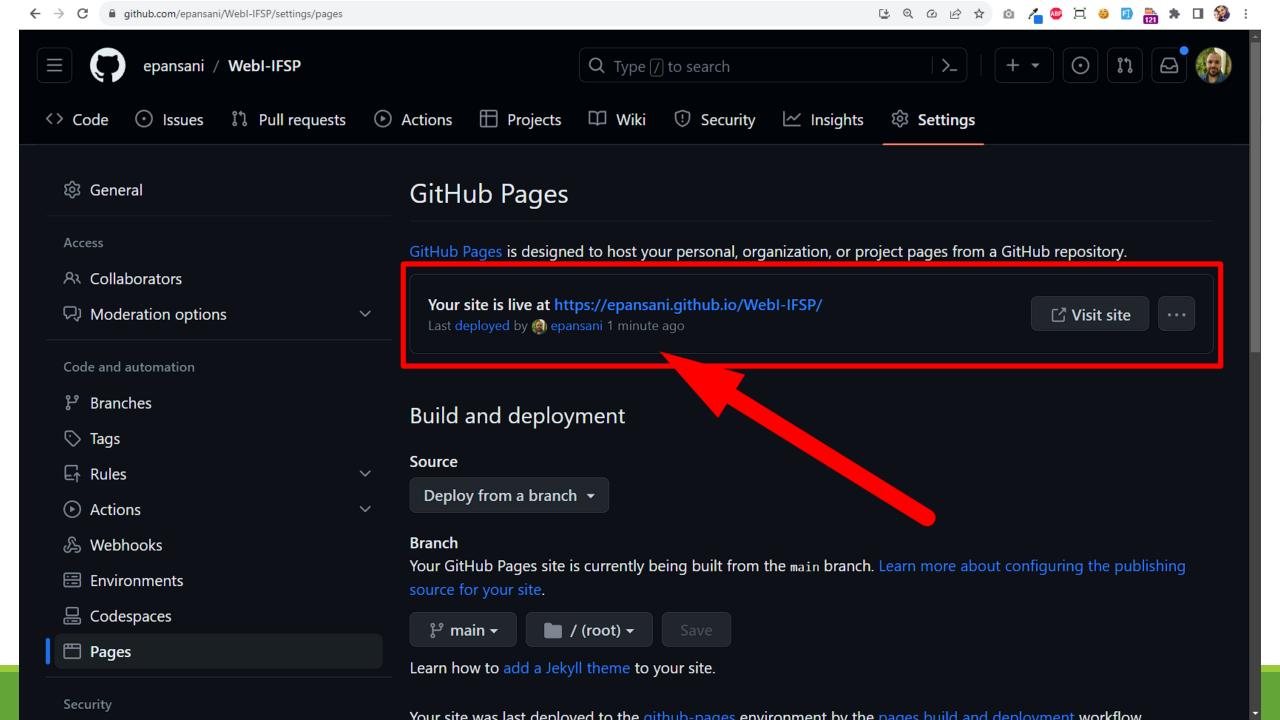
#### Passo a passo:

- 1. Crie o repositório (público!)
- 2. Publique seus arquivos, faça o commit
- 3. Quando estiver online vá para o site (github.com)
- 4. Vá em *Settings => Pages* 
  - 1. Escolha o branch (principal por exemplo) e salve
  - 2. Aguarde! Demora uns minutos.



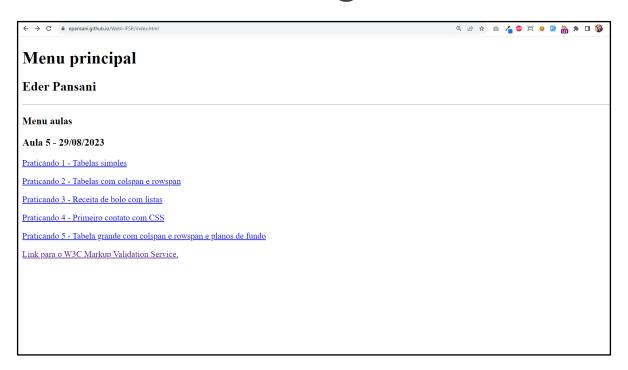
### Você deverá observar a seguinte mensagem:







### E no seu repositório deve ter algo como:



#### Menu principal

#### **Eder Pansani**

#### Menu aulas

Aula 5 - 29/08/2023

<u>Praticando 1 - Tabelas simples</u>

<u>Praticando 2 - Tabelas com colspan e rowspan</u>

Praticando 3 - Receita de bolo com listas

Praticando 4 - Primeiro contato com CSS

<u>Praticando 5 - Tabela grande com colspan e rowspan e planos de fundo</u>

Link para o W3C Markup Validation Service.