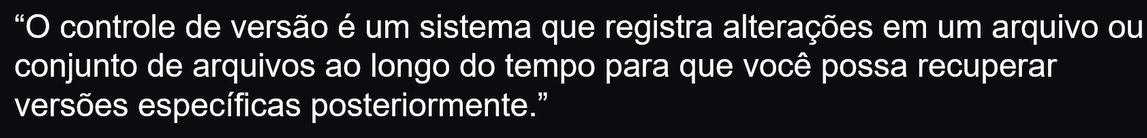
Guit Hub 4 Women

Version Control

Sistema = software



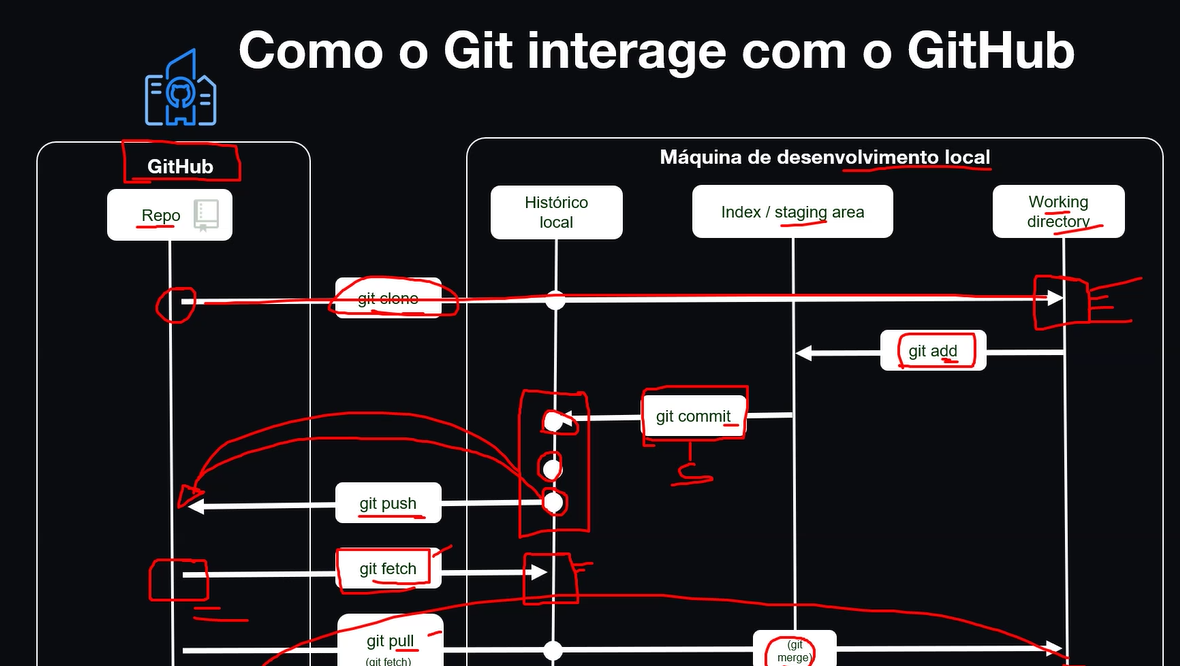
Git é uma ferramenta que faz controle de versão (versionamento) do tipo distribuído (dvcs) não usa um servidor centralizado.

Controle de versão mais antigos guardava as diferenças de versão por meio dos deltas (guarda a diferença entre um versão e outra) e o Git trabalha com snapshots (como se fosse um print do status de cada alteração - versão)

Versionamento centralizado x descentralizados

|  |  |
| --- | --- |
|  |  |

Versionamento descentralizado não depende necessariamente do computador servidor estar online. Ele só precisa estar online qdo cada computador baixa o cód, neste momento é feito um clone, cada computador fará sua BRANCHING (alterações) e são salvas, todas alterando o mesmo arquivo só que em computadores diferentes, isto td pode ser feito com o servidor offline, só que vão considerar voltar (push) para o computador servidor qdo ele estiver online. Qdo há conflito de “marshing” como qdo fizer o push e a alterção na mesma linha de cod estiver em conflito aí as partes terão que conversar para ajustes necessários.



Branch qdo trabalho em paralelo com o código

Commit é qdo é feito a alteração e qdo é feito o print das alterações

Cada commit gera uma descrição e tem por trás um rash (código)

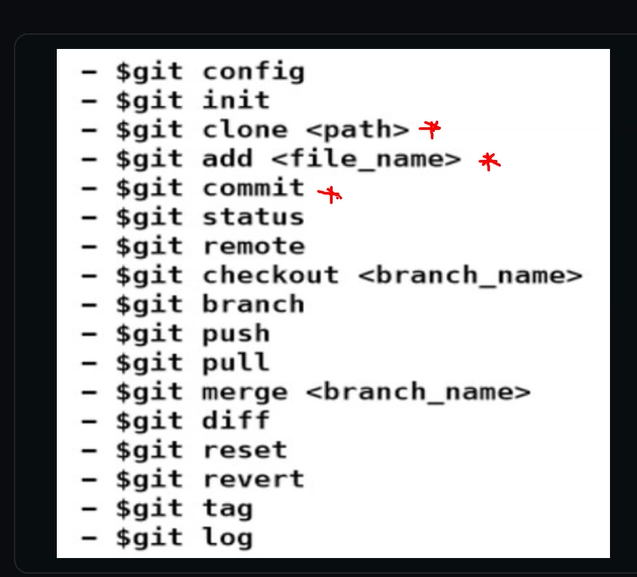
Git commit-m – todo commit tem que vir acompanhado de uma msg “”

Push para enviar ao repositório remoto do git

Pull para receber

Git fetch só vai mostrar a diferença entre uma Branch e a outra , não faz a merge pq depende do git pull, qdo fizer fará a merge.

Comandos básicos

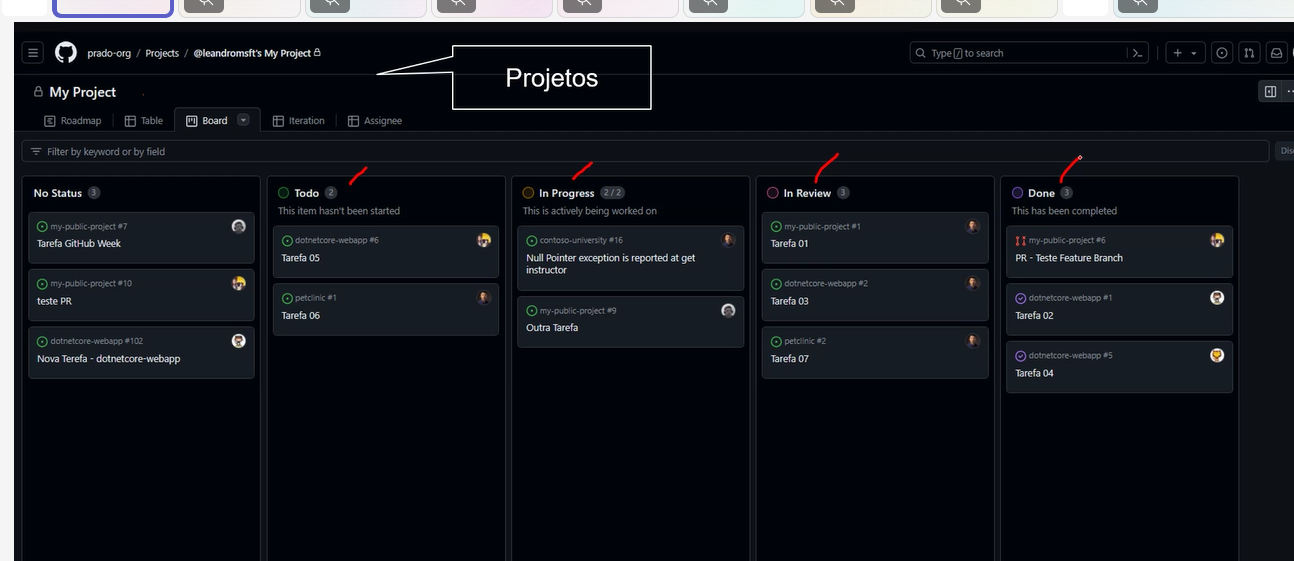


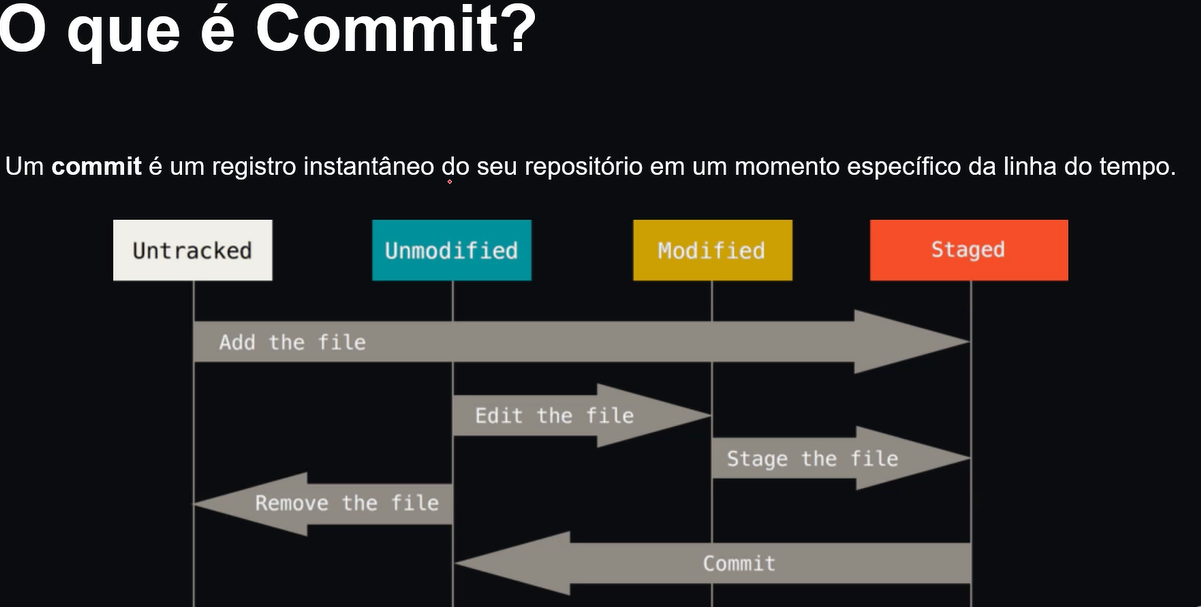
Git x Git Hub

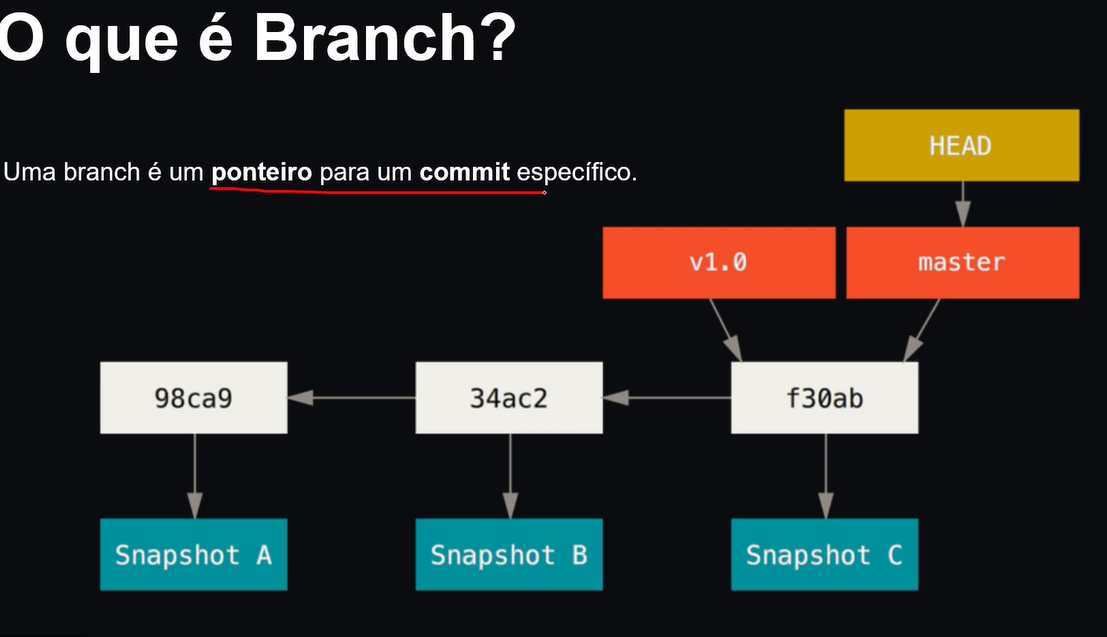
Git – sistemas (software) de controle de versão distribuído

GitHub é uma plataforma hospedagem de código-fonte e arquivos com controle de versão usando git

Quadro Kanban com issue







Tarefa

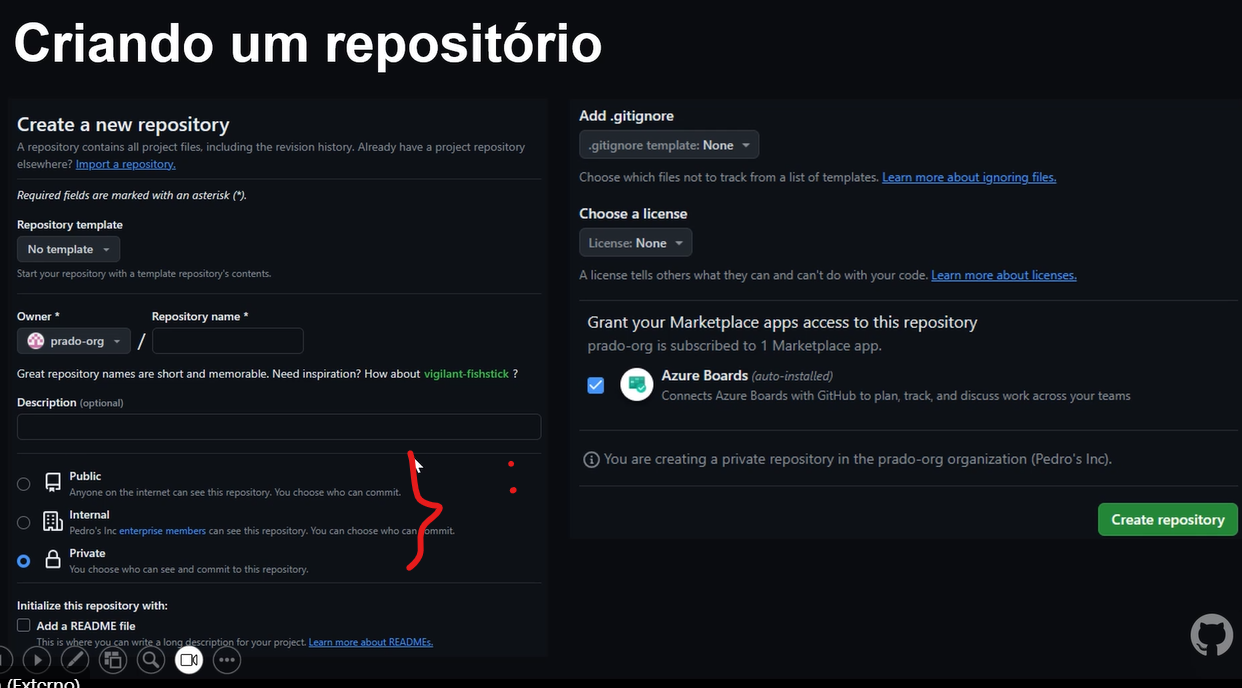
Criar uma conta free

<https://github.com/>

use e-mail pessoal

[Danielle1978-tech/Github4women](https://github.com/Danielle1978-tech/Github4women)

Repositório



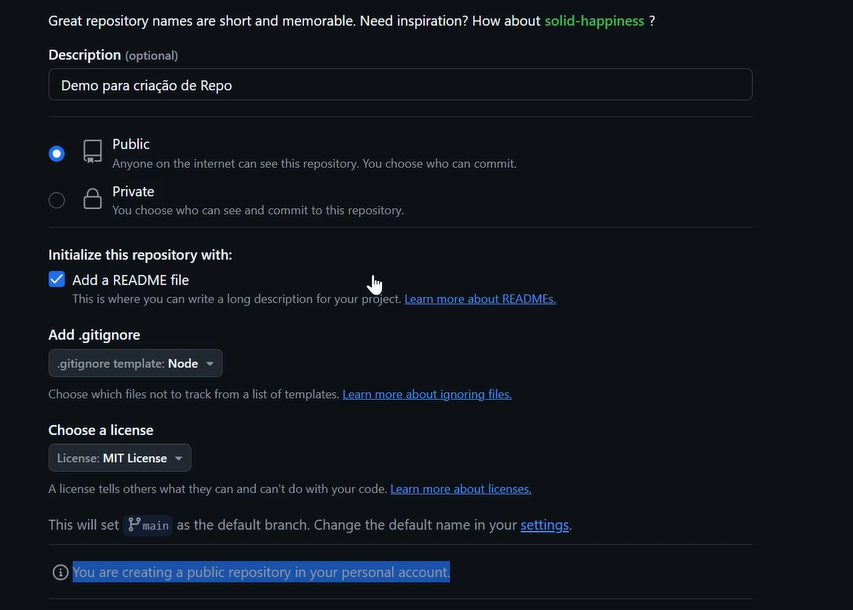
Visibilidade – importante para certificação

Obs na caixa só vai permitir o privado, não vai admitir o internal(interno)



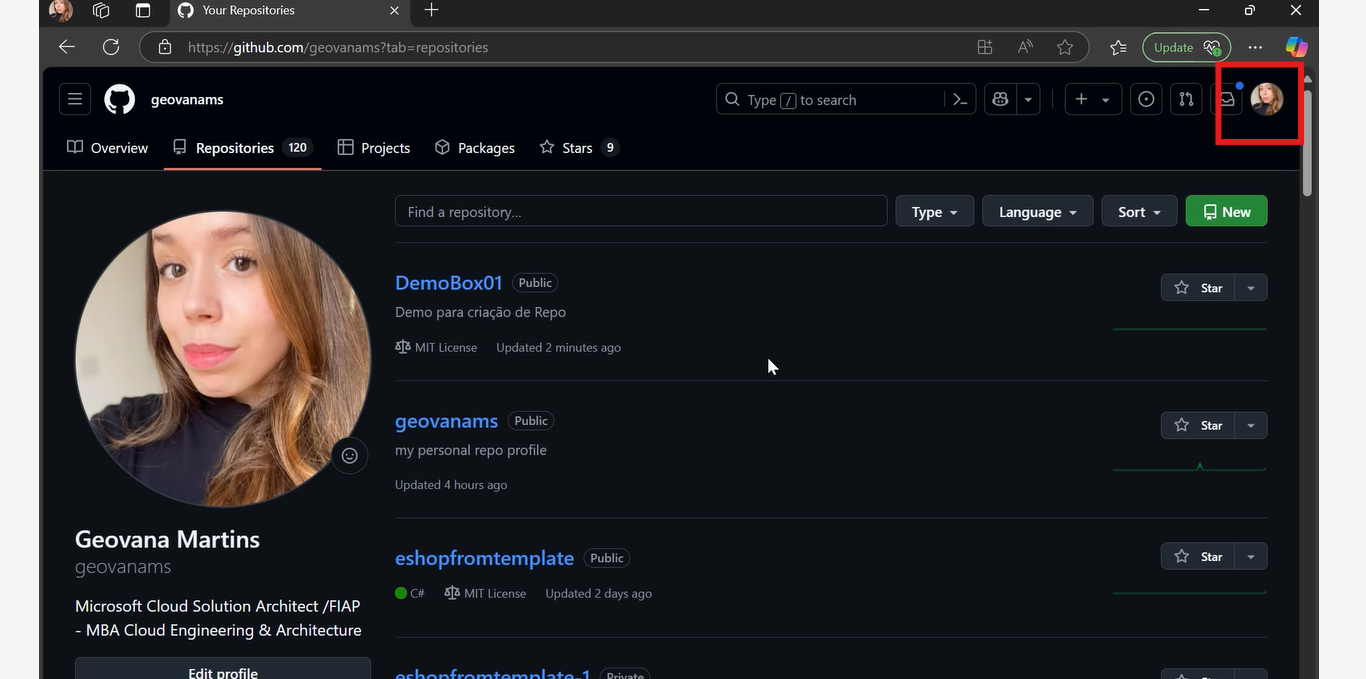
Git ignore – é qdo vc deseja ignorar algum tipo de arquivo, aí qdo tiver marcado não vai ser feito o versonamento.

Criar um repositório

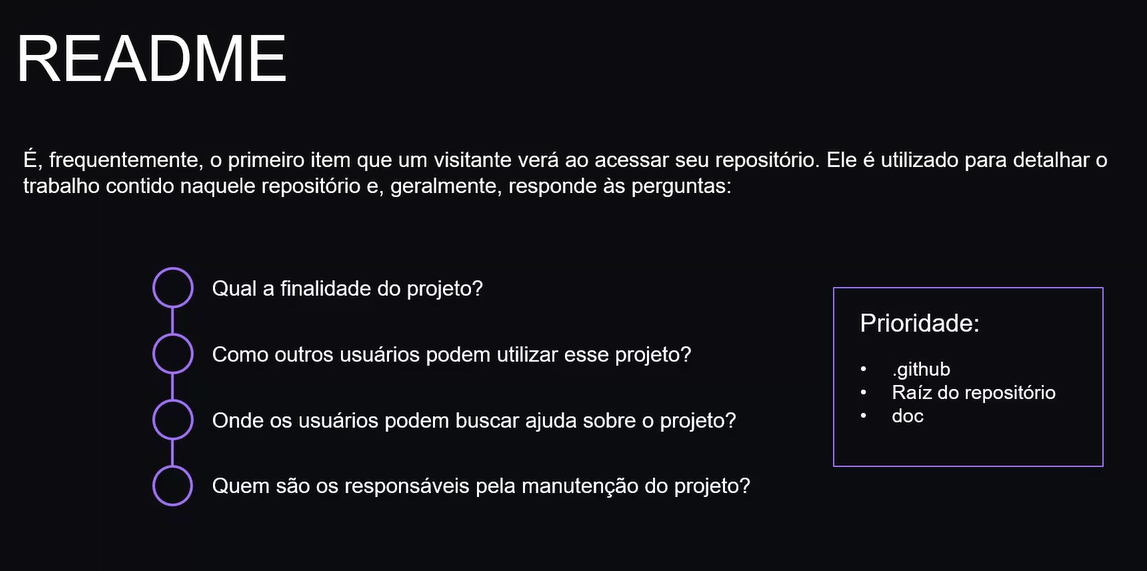


Qdo cria já sobe os arquivos que marcamos como ignorar node, licença, readme

Se clicar na foto vc pode ver seus repositórios (qdo vc cria como owner)



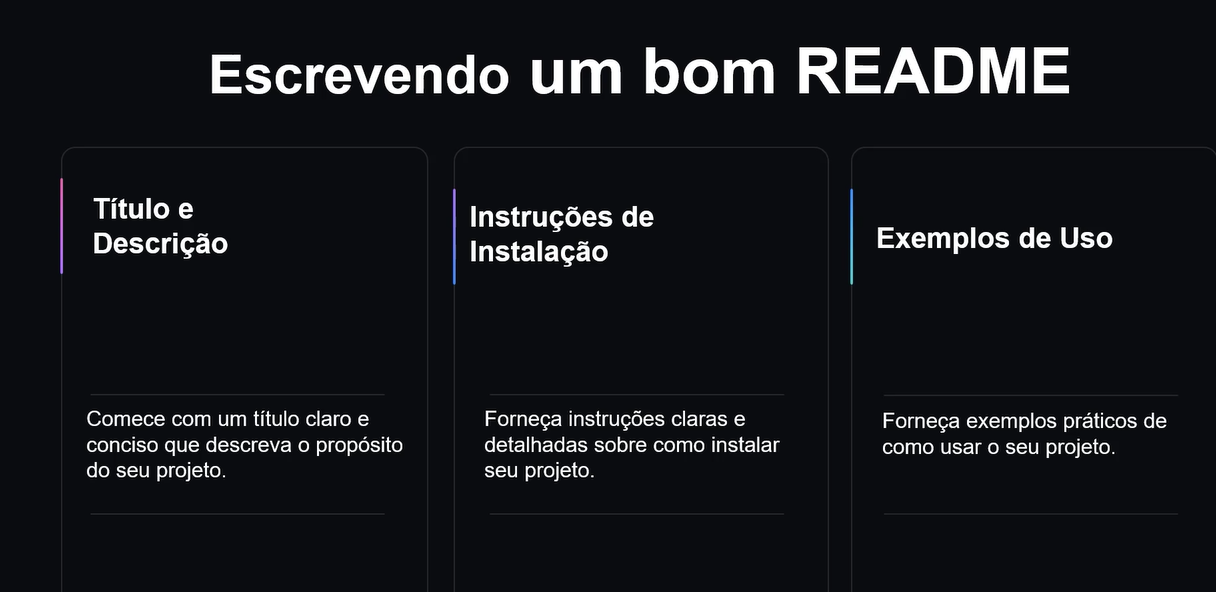




Readme tem que ser escrito deste jeito e com algumas destas extenção.

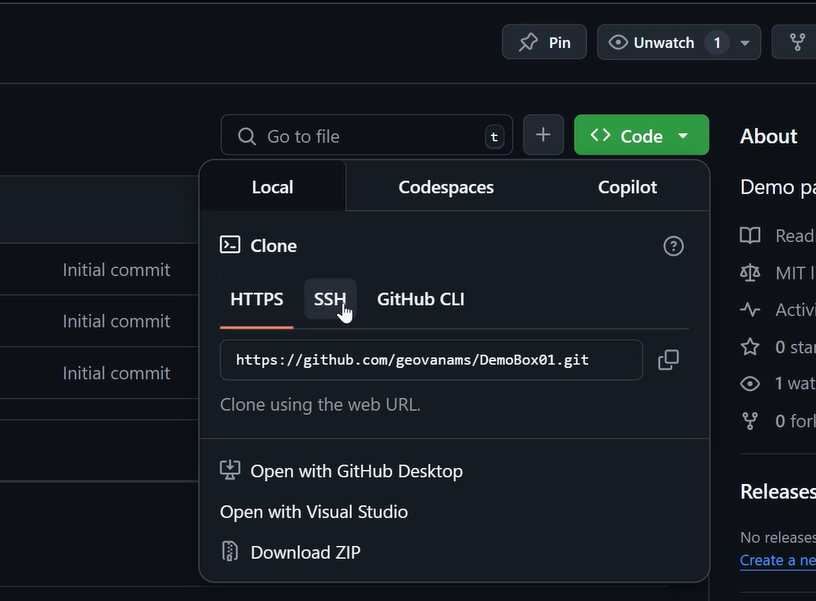
O ideal é que seja readme.md e aí já fica marcado no local correto de readme

Readme pode ser criado para um projeto como tb para pag inicial

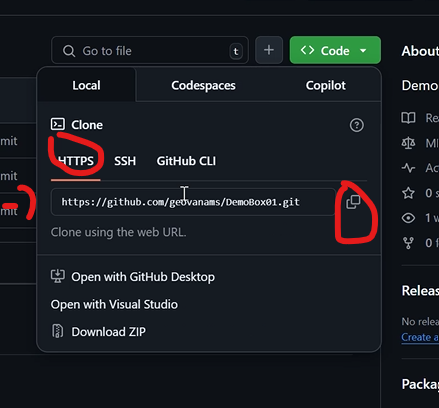


Clonar repositório colaborando com os códigos

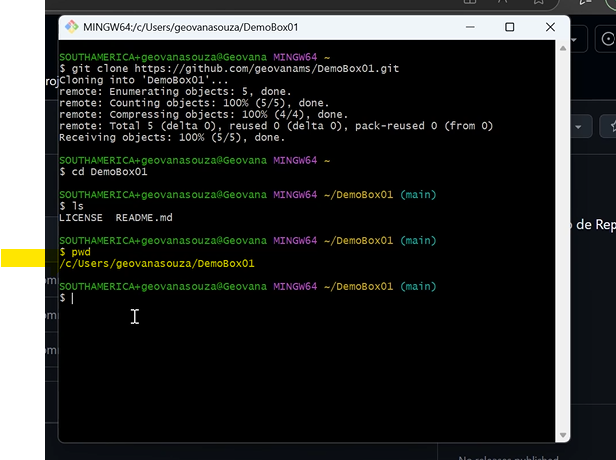




Abra um terminal e escreva git clone e copie a url

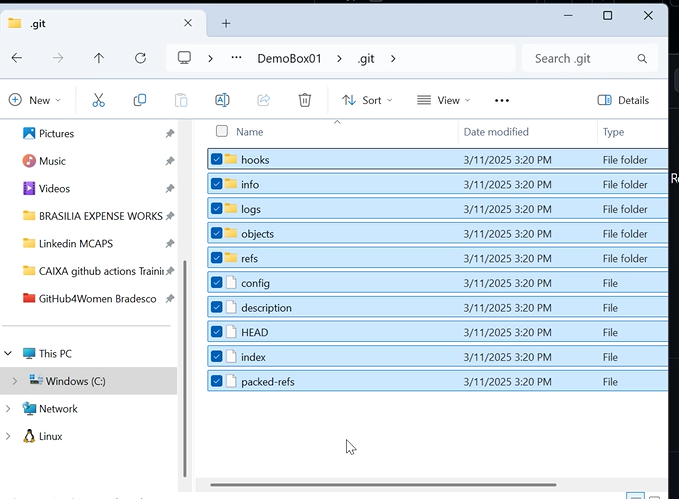


Onde vamos colar? Qual esta tela?

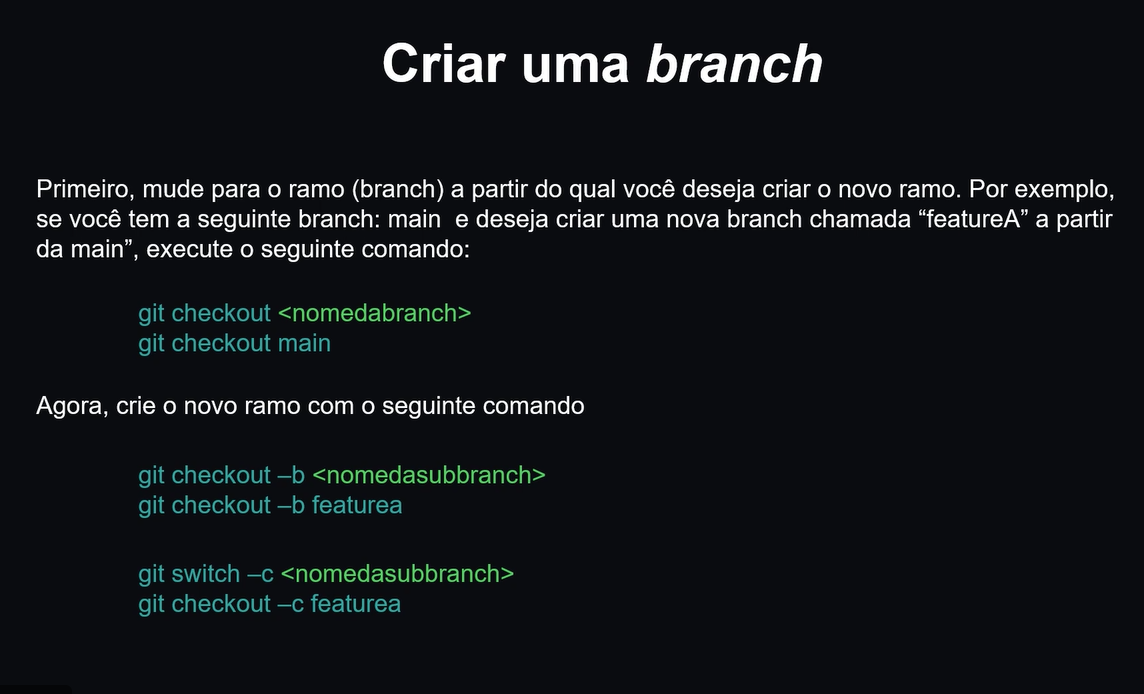


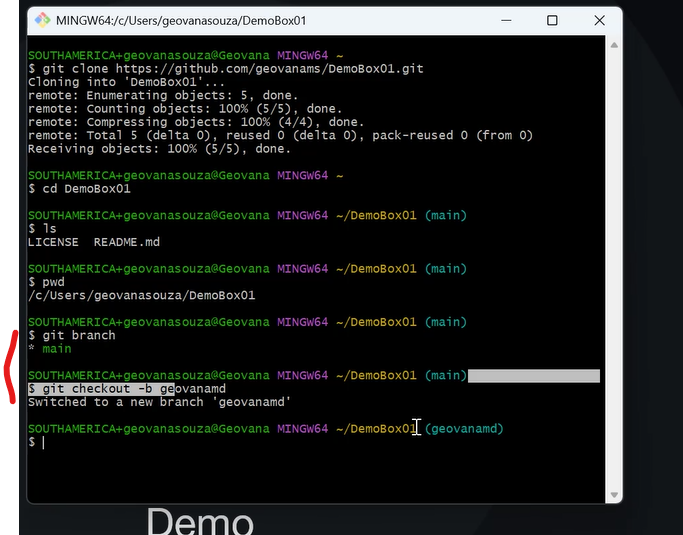
Qdo faz o git clone já cria como git em uma pasta.

A pasta .git é o que determina que ele seja repositório

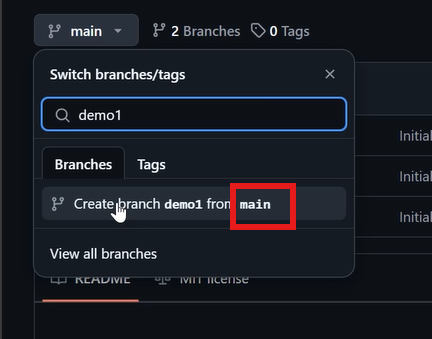
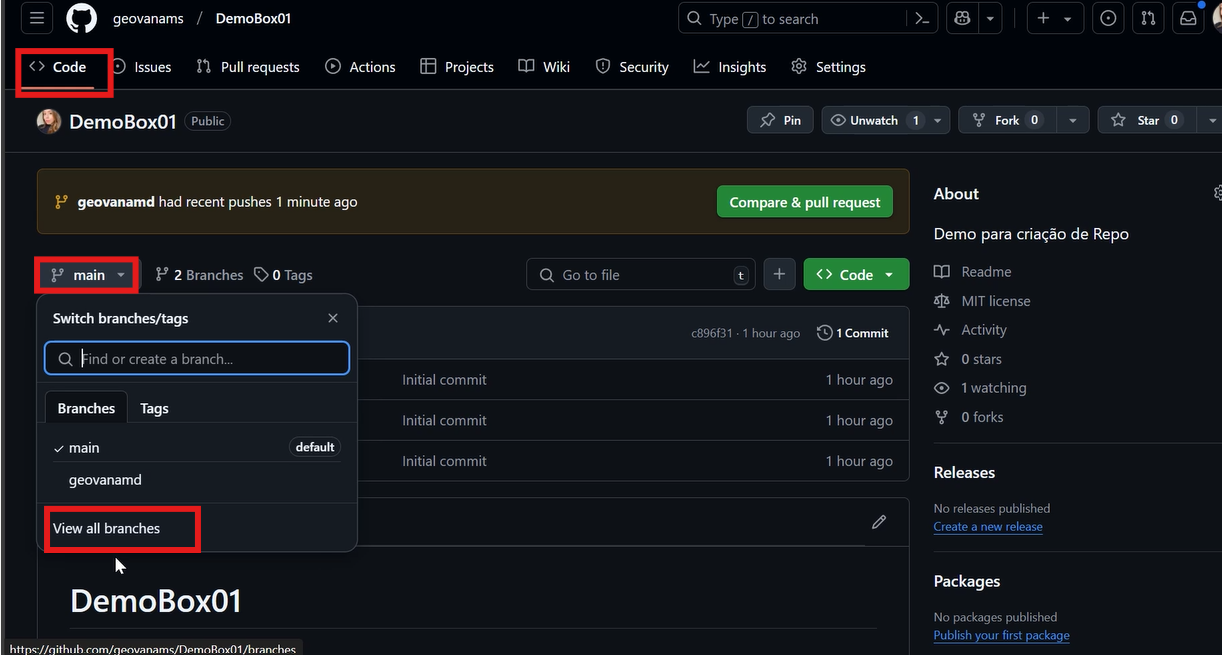


Para fazer alterações precisa criar uma Branch





Criar de forma mais rápida



Para subir de forma mais fácil, posso pegar um arquivo e anexar via upload ou criar um novo arquivo. É necessário fazer commit deixando a msg do que se refere

