Versionamento: controle de versão de software definido através de numerações de históricos diferentes. Tudo que vc fizer mudança vai estar associado a um número. Assim dá pra ver qd aconteceu a alteração. Se não comitar o que foi salvo, não é salvo.

Repositório: pasta onde o arquivo é colocado. Tem um .git nele pra ele ser rastreável pelo git. Ele pode ter subpastas nele.

Commit Conjunto de alterações do git, se vc quiser salvar no git o que vc mudou, tem que aplicar um commit. Vc tb pode adicionar comentários e dados sobre que fez (quem fez, o que fez, quando fez), data. É considerado mudança: criação, renomeação, exclusão de arquivo, inserção ou exclusão de linha (sendo que se vc mudou a linha, ela é considerda excluída e incluída). Dá pra voltar pra um commit específico se quiser.

Branch: separação de códigos que permite que muitas pessoas atuem no mesmo projeto de forma independente. Toda Branch inicial é a master, e depois tem a develop que é para uso em tempo de desenvolvimento. Depois de testar o que foi feito na máquina, se coloca o código nesse ambiente. Depois vai pro homolog que é para onde vai ser usado pela galera de negócios depois de ser aprovado no desenvolvimento e antes de ir pra produção. Master vai ser onde estará o código final que tá em produção. A euipe de desenvolvimento cria novas branchs para separar e desenvolver novas soluções para um produto de forma simultânea ou melhorar o que já existe de forma paralela.

Merge: une duas branchs em que duas pessoas estavam atuando simultaneamente. Quando isso é feito, é registrado um commit de merge.

Clone: transfere o depositório pra máquina local.

PUll: atualiza o repositório local. É realizado um merge entre o repositório online e o local pra evitar conflitos entre eles.

Push: pode enviar as alterações locais no repositório remoto, de forma que mantenha o online atualizado com o que você fez na máquina. A interface do git serve pra clonar o repositório, abrir uma pullr request e pra ver o que tem realmente no espaço.

Fork. Como contribuir com o repositório de outra pessoa, vc pode forkar o repositório, que fará que ele passe existir sem ser clonado no seu repositório exatamente e, se vc quiser adicionar algo lá, vc faz um pull request. Só faz fork no próprio site.

Pull request: após forkar o repositório, você pode resolver problemas dele e fazer pull requestts enviando soluções de erros e adicionando novos conteúdos ao que já existe. Quem é dona vai dizer se aceita ou não a pull request e se ela quiser vai fazer o merge e beleza.

UPStream é a origem remota da nossa pasta, é com ela que o pull e o push conversa.

Pra inicializar o git: git init

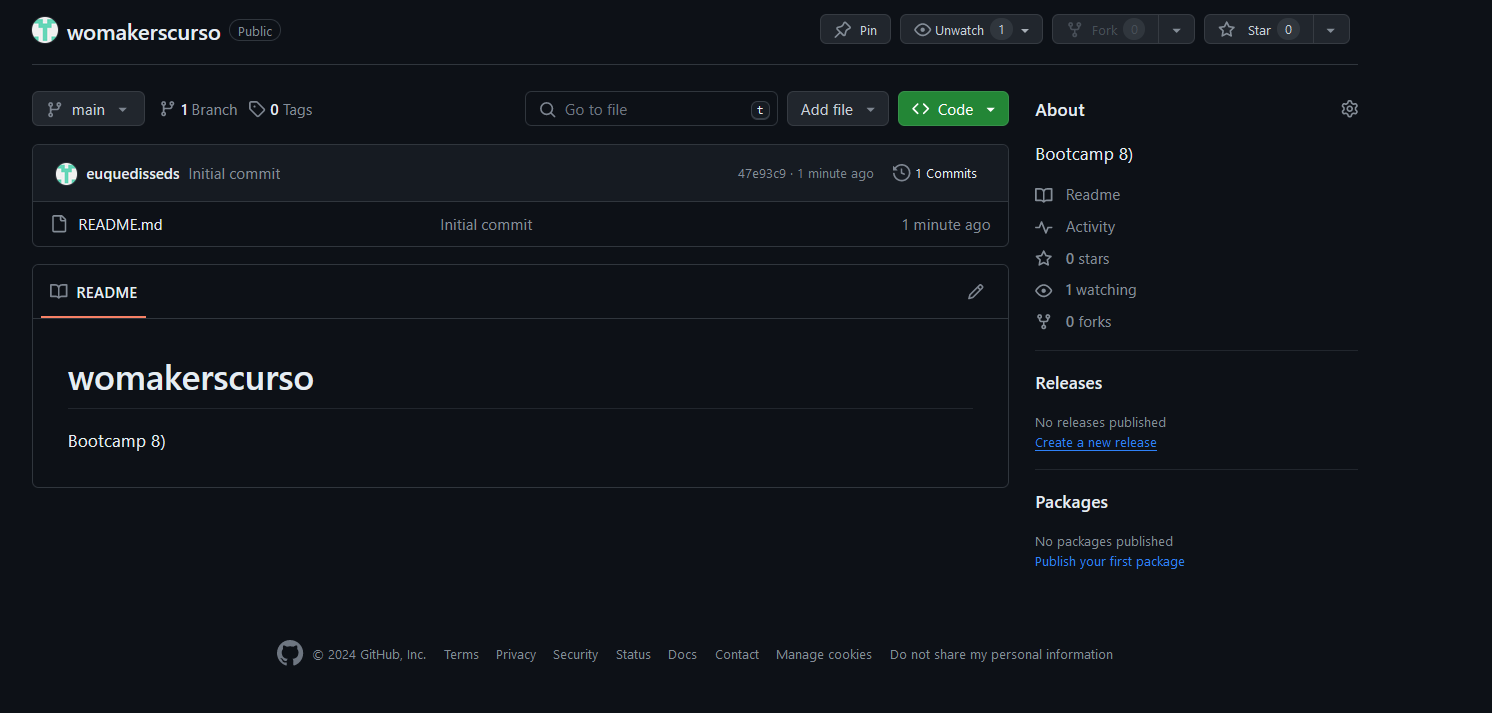
Para clonar: cria a pasta que vc quer no seu computador e escreve “git config –list” no termina do git

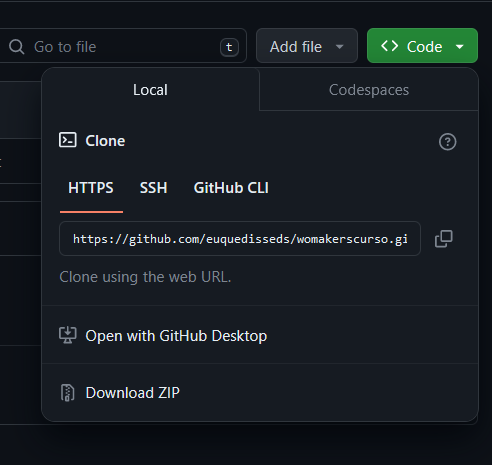
Git status verifica o status dos arquivos em relação ao online, inclusive mostra se te coisa q vc precisa subir ainda

Pra unir o git com o que vc quer, cria ele no site e aplica o comando

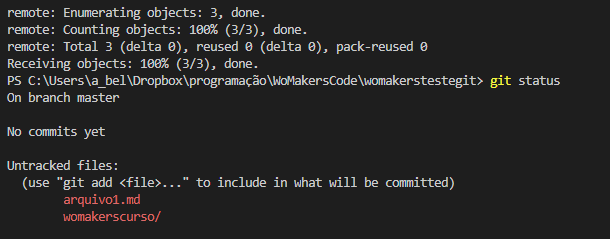
git remote add origin <REMOTE\_URL>

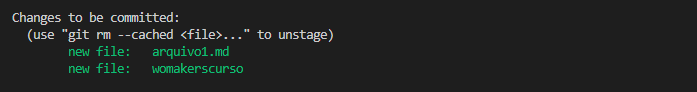
com o link que está aqui:



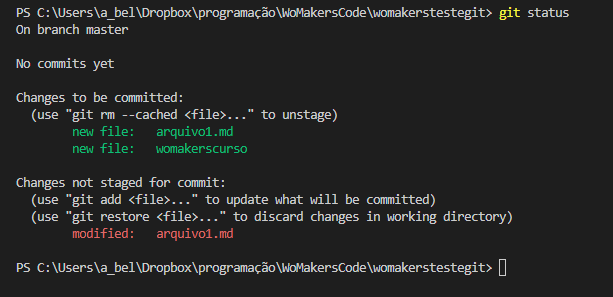


Ao fazer o git status, vc tem os arquivos que vc mexeu mas não adicionou nem subiu em vermelho:



Quando adiciona 

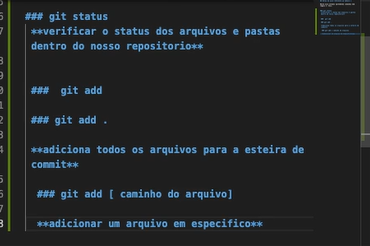
E se vc muda coisa, ele volta pro vermelho



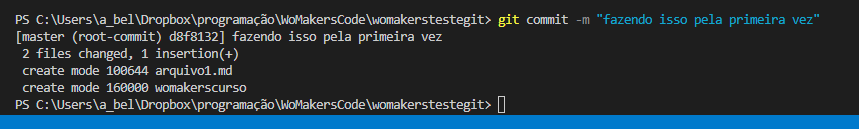
Pra adicionar um arquivo na “fila” do git, vc coloca git add nomedoarquivo :



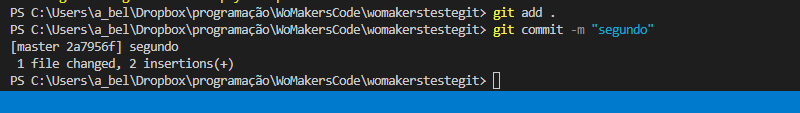
Se vc coloca git add . ele vai colocar td mundo que precisa



Pra fazer o comit, tem que dar o comando git commit -m “comentário”

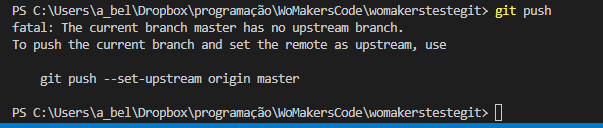


Tudo que vc fizer, vai ser apresentado com o que foi feito. Daí o passo é sempre fazer o add e depois o commit

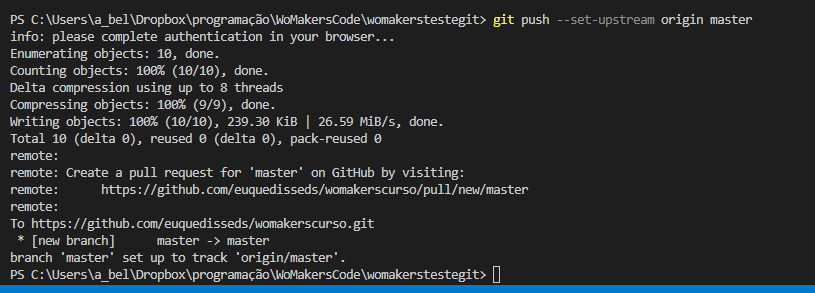


Depois de fazer o commit, vc tem que fazer o push:

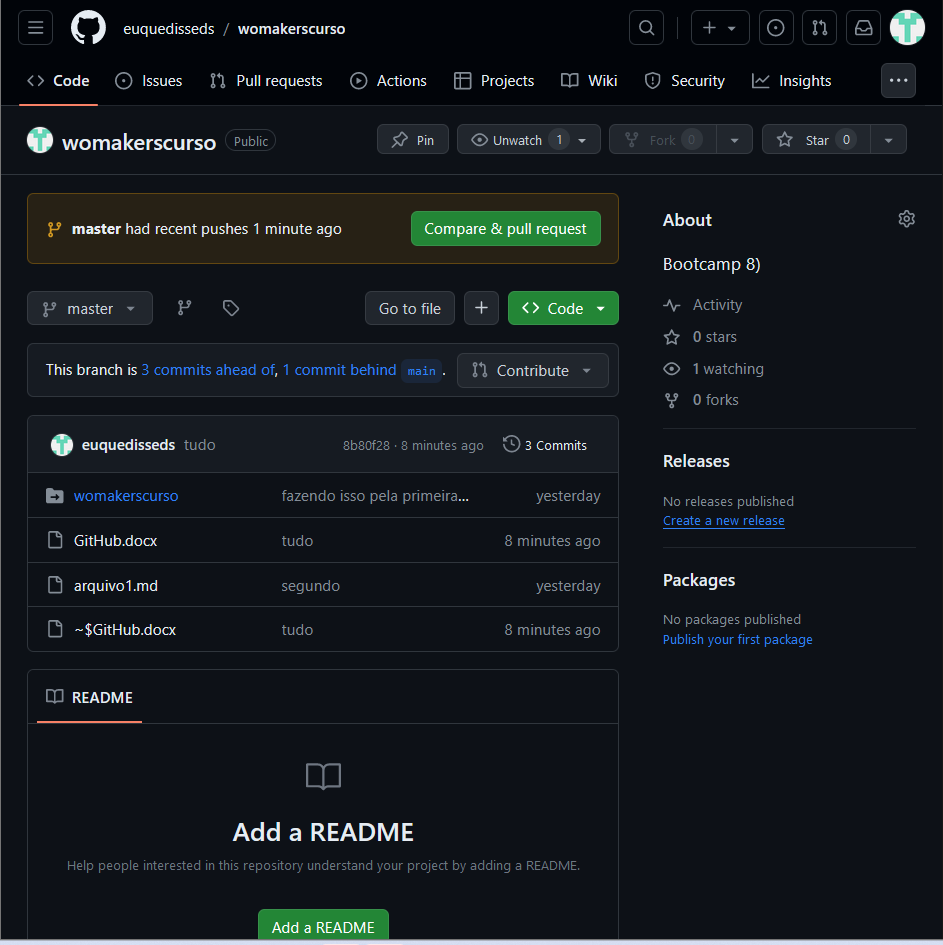
Git push



Mas se vc não tiver o upstream, vai dar ruim, então vc tem que criar uma uptream com o comando que ele fala



Daí vai atualizar lá no arquivo na plataforma:



O próprio arquivo vai mostrar o que foi feito ou não pra vc controlar a mudança.

Se vc mudar a coisa no repositório online e quiser trazer pra sua máquina, usa git pull, vai ser mais útil quando tiver mais gente mexendo ele.

Branch permite que diversas pessoas possam trabalhar no mesmo projeto

