

## E vamos de git...

Gabriela Farias

O git é uma ferramenta de hospedagem de códigos ou arquivos utilizada principalmente no meio de desenvolvimento de software, ela possibilita que pessoas em computadores diferentes consigam alterar um determinado projeto de software, agregar funcionalidades e realizar modificações. Ela possibilita o trabalho em grupo, sem complicações e perda de código. Além disso, o Git pode hospedar vários tipos de arquivos, não apenas para desenvolvedores. Vale a pena conferir!

Atualmente, para desenvolvedores de software, a grande maioria das IDE utilizadas apresentam meios de realizar o versionamento de código pela própria IDE. Porém quando utilizamos meios para desenvolvimento de software é importante que conheçamos como esses funcionam, ou seja, nesse caso conhecer os comandos do git realizados no terminal para saber como utilizar quando for necessário. Além disso, não é sempre que a IDE vai estar presente, não é mesmo?

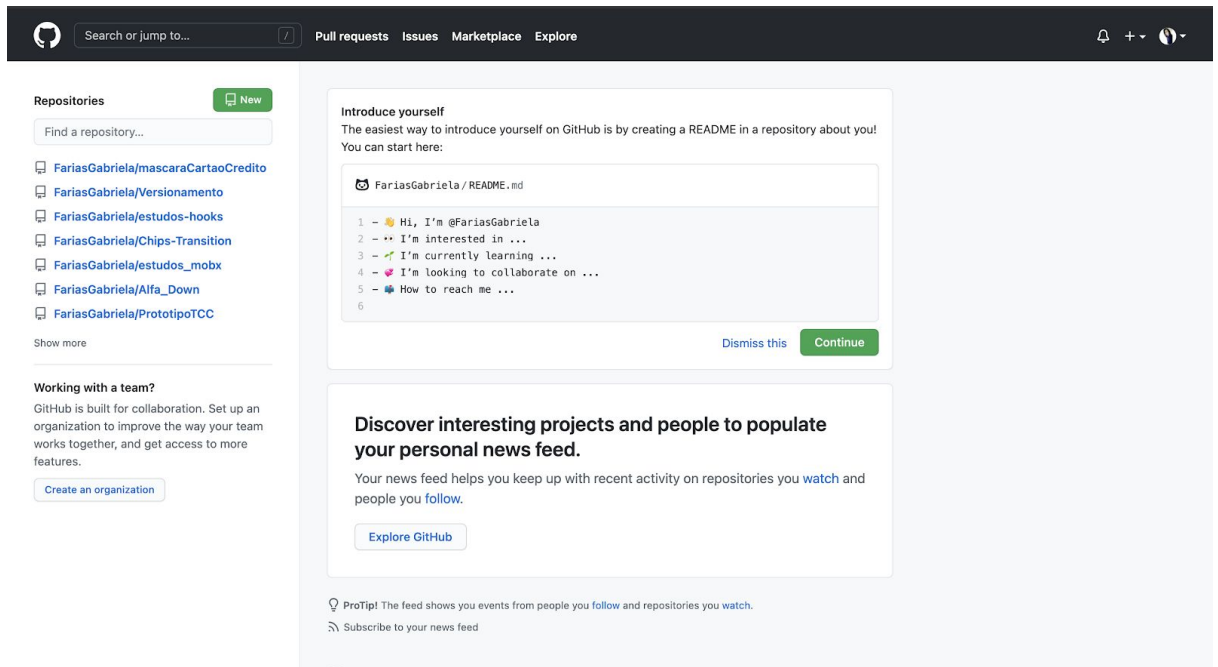
Primeiramente começamos com o seguinte comando:

***git init***

Esse comando inicializa o git, a partir de agora todos os arquivos dentro desta pasta, estão sendo vistos pelo git. Para conferir que arquivos estão contidos, pode-se utilizar o seguinte comando:

***git status***

Esse comando retorna uma lista de todos os itens que estão disponíveis para serem enviados pelo git. Em seguida, é preciso criar no git um novo projeto, para conseguir enviar suas alterações. No exemplo a seguir eu usei o GitHub, muito popular para desenvolvedores de software.



O canto superior direito contém um ícone de +, clique neste item para criar um novo projeto. Após a criação do projeto, será gerado um SSH, que nada mais é que uma chave que fará a ligação dos arquivos na sua máquina e o local que o git irá armazená-los. Copie essa chave SSH e cole na pasta do terminal juntamente com o seguinte comando:

***git add remote origin <CHAVE SSH>***

A partir de agora, todos os arquivos que estão nesta pasta, podem ser enviados para o repositório criado, mas para isso faltam alguns passos, vamos lá?

O próximo passo é escolher quais arquivos desta pasta você deseja enviar, temos duas opções. Caso você queira enviar tudo, basta rodar o comando:

***git add .***

E todos os arquivos serão enviados, mas se você quiser enviar uma parte deles apenas, basta rodar:

***git add <nome do arquivos>***

Esse comando pode ser rodado com quantos arquivos você achar necessário. Opa, deu ruim com algum arquivo que foi errado sem querer? Não se preocupe, basta rodar o comando:

***git reset <nome do arquivo>***

que esse arquivo irá sair da sua lista de arquivos para enviar. Após adicionar ao git os arquivos da sua escolha, chegou a hora de enviar, para isso, temos que lembrar que todo conjunto de arquivos enviados precisa de um título, certo? Para isso basta rodar o comando:

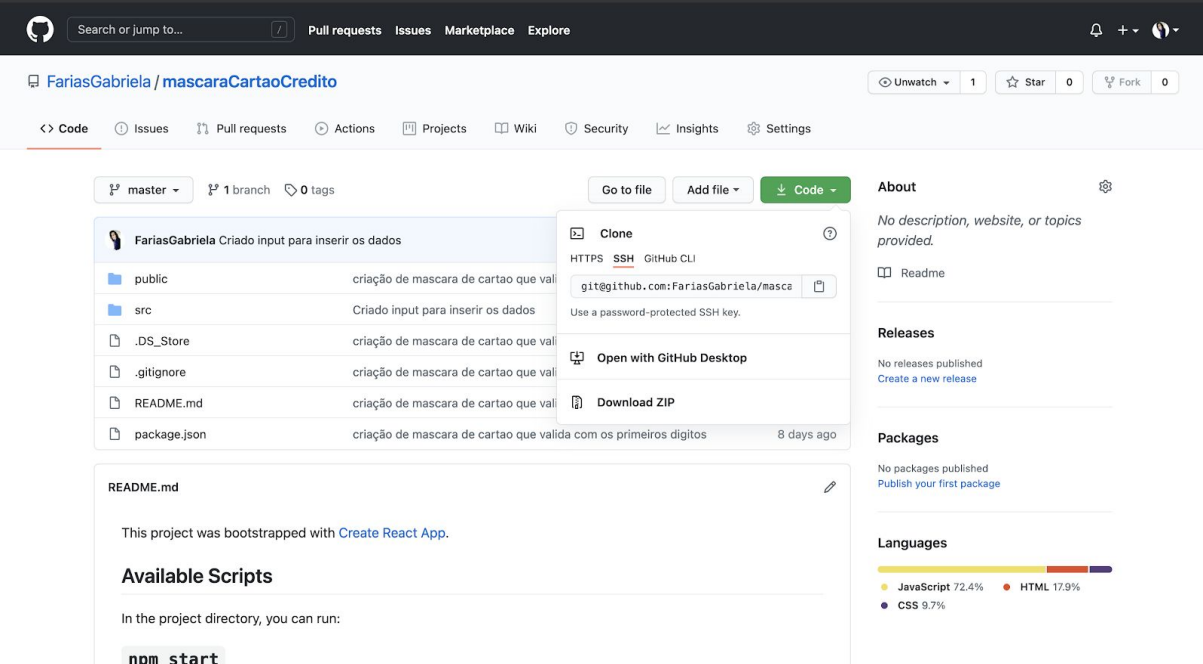
## ***git commit -m "Mensagem correspondente ao que está sendo enviado"***

Agora seus arquivos estão prontos para serem enviados, basta rodar o comando

## ***git push***

Em alguns casos, quando é o primeiro envio daquele ramo, conhecido como branch, vai aparecer uma mensagem após o git push, para inicializá-lo, basta copiar a mensagem e tudo certo!

Bom, mas e agora, se meu amigo quiser baixar essas alterações no computador dele? Fácil! Basta ele ter acesso a esse repositório, copiar a chave SSH, no GitHub, usado como exemplo anteriormente, fica no botão verde chamado Code



Basta copiar essa chave, navegar pelo terminal para a pasta onde ficará esses arquivos e rodar o comando:

## ***git clone < Chave SSH >***

E tudo resolvido, seu projeto estará em outro computador e pode ser modificado. E se você está se perguntando como baixar as alterações realizadas por outras pessoas, é só rodar o comando

## ***git pull***

Com esse comando, todas as alterações feitas por outras pessoas vão parar na sua máquina, simples né?

Bom, e se você quiser realizar apenas uns testes, mas não quer se esse código vá para o projeto principal, apenas espelhar seu código atual para alguns testes? Basta criar uma "cópia" dele, porém não se preocupe, o git junta automaticamente os dois, caso você mude de ideia posteriormente. Para criar essa cópia rodar o comando:

***git checkout -b <nome da branch>***

Esse comando também serve para navegar entre outras branch, basta retirar o -b e rodar

***git checkout <nome da branch>***

Após rodar esse comando, você já estará em outra branch, não se assuste se alguns dos seus arquivos sumir ou forem modificados, é porque essa branch está diferente daquela antiga que você estava. E se o teste deu certo e você quer juntar tudo, basta voltar para a branch antiga e rodar o seguinte comando no terminal:

***git merge <nome da branch com o código novo>***

A aula de git vai ficando por aqui, espero ter contribuído para o ensino e lembre-se, deixa esse PDF salvo num cantinho do seu notebook, além de um ótimo meio de estudar mais sobre como o git funciona, nunca se sabe quando a IDE vai te deixar na mão, não é mesmo?

E por último, essa aula de git é um recurso educacional aberto, após utilizá-lo, você pode agregar mais a ele, além de compartilhar com os amigos e usar como base.