Este rápido tutorial é pra você que já tem um conhecimento básico sobre ferramentas de controle de versão ou precisa de um guia rápido para Git com linha de comando. Também pode ser muito útil se você está apenas acostumado a usar outras ferramentas de desenvolvimento com Git e não sabe os comandos que são executados por trás dela. Aprenda os comandos e não ficará dependente de nenhum outro software. Veremos os comandos mais usados para se trabalhar com Git. Vou assumir que você já possui cadastro no [GitHub](https://github.com/) e já tem Git instalado em seu computador.

**Configure seu Usuário**

O controle de versão controla quem faz as alterações num projeto. Então você deve configurar seu usuário:

git config --global user.name "Bruno.Orlandi"

git config --global user.email "meu.email@gmail.com"

Substitua com seu Usuário e E-mail da sua conta no GitHub.

**Crie um Repositório**

O repositório é onde estarão os arquivos versionados, no servidor Git e no seu computador em uma pasta. Crie uma pasta, já com o nome do seu repositório e execute o comando dentro dela:

git init

No meu exemplo a pasta se chama **Tutorial**

**Crie o Commit**

Um Commit é um pacote de alterações feitas no repositório. Cada commit possui arquivos alterados, autor e uma mensagem de resumo.

Vamos criar o primeiro commit criando um arquivo especial para o GitHub chamado **README.md**. Dentro deste arquivo escreva:

**Meu repositório Git ====================**

Apenas repositório Git

Salve o arquivo e agora você irá usar o comando mais usado em Git:

git status

O resultado deve ser:

On branch master

Initial commit

Untracked files:

**(**use "git add <file>..." to include **in**what will be committed**)**

README.md

nothing added to commit but untracked files present **(**use "git add" to track**)**

O comando git status sempre vai retornar o estado atual do repositório e explicar o que você pode fazer em seguida. Neste caso não há commits criados ainda então você pode criar o primeiro com o arquivo README.md. Devemos primeiro adicionar o arquivo a um commit com o comando:

git add README.md

Agora o arquivo está pronto para ser empacotado em um commit. Escreva o comando de commit incluindo uma mensagem que explique o que sua alteração faz no repositório.

git commit -m "Adiciona arquivo README.md"

Pronto, você fez o primeiro commit. Pode listar os commits mais recentes com o comando:

git log

**Suba seus Commits**

Agora você precisa sincronizar o seu repositório com o GitHub, mas ainda não fizemos nenhuma conexão entre eles. Acesse seu GitHub e na página inicial você encontrará o botão https://brorlandi.github.io/img/git_start/newrepo.png

Crie o repositório com o nome que desejar, neste exemplo vou usar **Tutorial**. Após criado, o repositório já exibirá os comandos necessários para iniciar e configurar o seu repositório localmente. Como já criamos o repositório localmente e queremos subí-lo no GitHub seguiremos com a opção *“push an existing repository from the command line”*. Primeiro devemos configurar qual o repositório remoto. Execute usando a URL do seu repositório:

git remote add origin <https://github.com/BrOrlandi/Tutorial.git>

Com isso você configurou o remote origin para o GitHub. Você já pode subir seus commits para lá usando git push, mas na primeira vez o comando deve ser executado para configurar a branch master para o GitHub. Opa, logo falarei sobre branchs em outro post. Ao executar o comando será solicitado o usuário e senha para autenticar no GitHub.

git push -u origin master

Olhe o repositório no GitHub e verá seu arquivo **README.md** lá e exibido na página inicial do repositório. Pronto, você aprendeu os principais comandos para se trabalhar com Git. Git é uma ótima ferramenta para se trabalhar em equipe também mas apenas estes comandos não são suficientes.

**Sincronize o repositório**

Você pode editar os arquivos de texto direto no GitHub. Vamos editar o README.md lá, clique nele e sem seguida clique no botão de editar . Ao final da tela de edição você deverá preencher uma mensagem de commit. Sim, você está fazendo um commit direto pelo GitHub.

Agora o repositório possui dois commits no GitHub e no seu computador apenas um. Para sincronizar o seu repositório você deve puxar as atualizações.

git pull

Você verá que seu repositório local foi atualizado com o novo commit.

**Baixe outros repositórios**

Você pode querer baixar um repositório Git no seu computador e começar a commitar nele. Para isso existe o git clone que basicamente faz o download do repositório, seus arquivos, commits e configuração de remotes. Você pode clonar qualquer repositório público no GitHub e outros servidores Git. Você precisa da URL Git desses repositórios, você pode obtê-la no botão https://brorlandi.github.io/img/git_start/clone.png

git clone <https://github.com/BrOrlandi/Tutorial.git>

O comando irá criar a pasta **Tutorial** com o repositório.

**Continue aprendendo…**

Ainda farei outros posts sobre Branchs e Forks com Git. Assim dando mais um passo ao trabalho colaborativo com Git.