***GIT, um B-A-Bá***

**Primeiro passo – Criar uma conta no GitHub**

- Acesse <http://github.com> e crie sua continha grátis;

- Feito isso, guarde esse endereço, que é o repositório criado pela @Dinara: <https://github.com/dinaralima/acabecadobuda>;

**Segundo passo – Baixe o Git Bash**

- O Git Bash é a plataforma com o terminal do Git, onde você vai executar os comandos. Instale no procedimento padrão (clicando Next até o fim da vida).

- Depois de instalado, rode os seguintes comandos pra fazer a configuração básica:

- *git config –-global user.name “*Nome Sobrenome*”*

- *git config –-global user.email “*seuemail@servidor.com*”*

- Feito isso, vamos aos commandos e interação com o Hub;

**Criando uma chave SSH**

A chave SSH é uma credencial de acesso baseada no protocolo SSH (Secure SHell) – basicamente uma senha *on steroids* que fornece um canal seguro em uma rede insegura, e autentica o computador e usuário.

Pra fazer isso, abra o Git Bash e digite:

*ssh-keygen -t rsa -b 4096 -C “seuemail@servidor.com”* (O MESMO CADASTRADO LÁ EM CIMA)

Em seguida você tem duas opções:

- Digitar *cat ~/.ssh/id\_rsa.pub*

- Ou ir no caminho C:\Users\*Seu usuário*\.ssh e abrir o arquivo *id.rsa* com qualquer editor de texto (não sei como fica isso no Mac – sorry, Otto)

Feito isso, copie essa chave e vá ao GitHub. Clique no seu usuário, no canto superior direito e vá em Setings. No menu da esquerda, clique em:

SSH and GPG Keys

New SSH Key (botão verde na parte superior direita)

Dê um título no campo Title (p.e., “PC de casa”)

Cole o textão no campo Key

Add SSH Key (botão verde logo embaixo)

Feito ;)

**Clonando repositórios**

- Abra o terminal (Git Bash) e digite:

*git clone git@github.com:dinaralima/acabecadobuda.git*

Ele vai clonar o repositório em uma pasta padrão. Se você quiser mudar, vá nas Propriedades (botão direito no ícone do GIt Bash) e mude o local no campo Start In.

Ele vai perguntar se você tem certeza e pedir para digitar *yes/no.* Só colocar yes e pronto. Feito isso, não vai mais ser preciso clonar o repositório.

Toda a navegação é por linha de comando, e isso dificulta um tanto até pegar a manha. Mas no Title do Bash você consegue se localizar em que pasta está. A partir disso, a navegação entre uma pasta e outra é:

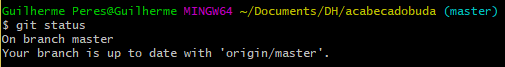
*cd* – abreviação pra “Change directory”: Se eu estou, por exemplo, em C:\Users e quero ir pra pasta Guilherme Peres dentro dela, eu digito *cd “Guilherme Peres”*. Para voltar, basta digitar *cd*

*ls* – digitando isso o terminal vai listar todas as pastas ou arquivos existentes. As pastas vão ser sempre os nomes com uma barra/ no final. Com isso fica mais fácil saber pra onde você está indo

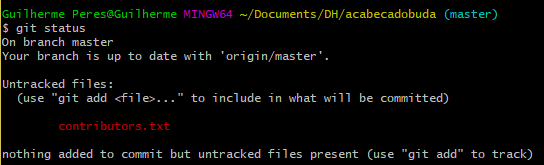
O terminal sempre vai indicar em amarelo em qual pasta você está. Para facilitar, criem pastas sem espaço no nome. Quando estiverem na pasta do repositório clonado, ele vai aparecer tudo em amarelo com um (master) em azul, indicando que esse é o branch principal do projeto. Agora dá pra começar.

**Depois de alterar os arquivos**

Antes de qualquer movimento, digitem *git status* – esse comando vai dizer em que pé você está com o repositório. Se vocês alteraram qualquer arquivo – ou se outra pessoa alterou -, ele vai acusar que tem arquivos não trackeados e que você está desatualizado com o projeto. Senão, ele vai falar que está tudo ok (*you are up to date)*



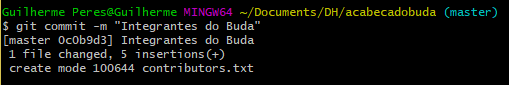
Ou



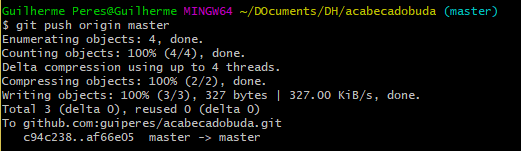
No segundo caso, o arquivo não está trackeado. Eu simplesmente criei/alterei na minha máquina e o Git não sabe o que fazer com isso, por isso avisou que tem um corpo estranho no repositório.

Para começar a trackear, vou digitar *git add contributors.txt*. Com isso, o arquivo vai para uma “sala de espera”, ou um estágio chamado **unstaged**.

Em seguida, é a hora de commitar, que nada mais é do que informar ao Git que aquele arquivo que foi adicionado irá para o repositório. O comando é *git commit -m “Título da alteração”*



Próximo passo é despachar o arquivo para o repositório. Para isso, use o comando *git push origin master*



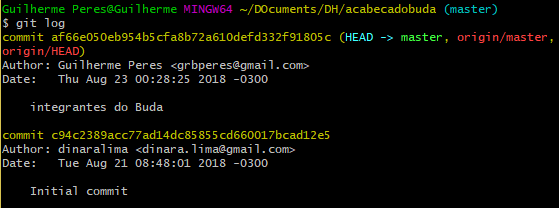
Pronto, o arquivo já vai para o repositório.

**AH MAS E SE EU TÔ MEXENDO NO MESMO ARQUIVO QUE OUTRA PESSOA?**

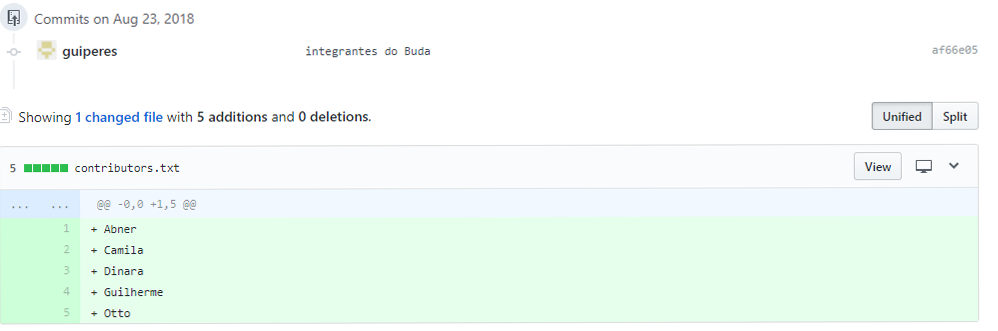
O Git vai rejeitar o push e te obrigar a fazer um *pull*, ajustar qualquer conflito e então fazer o *push*

**DÁ PRA VER AS ALTERAÇÕES FEITAS?**

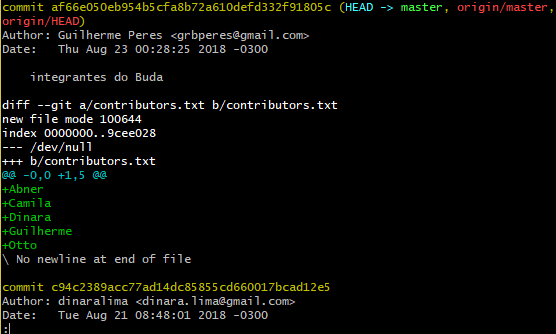
Dá, com o comando *git log*. Ele aparece assim, com toda a capivara do fulano que mexeu no repositório:



No GitHub você consegue ver os detalhes dessa alteração, e qualquer conflito vai aparecer em vermelho. Aqui não, porque só adicionei um arquivo novo:



Dá pra ver também no próprio terminal, com o comando *git log -p*



**E SE EU FIZ CAGADA E QUERO VOLTAR UMA CASA?**

Tem processos diferentes, tanto pra quando você esquecer de um arquivo no commit quanto pra tirar um arquivo commitado ou já enviado. Eu não consigo explicar melhor do que o que tá aqui: <https://git-scm.com/book/pt-br/v1/Git-Essencial-Desfazendo-Coisas>

Esses é basicamente todo o processo que a gente vai utilizar. Agora é ir errando e aprendendo.

bjks,

Gui.