**Git init** -> Inicia um repositório vazio na pasta onde vc deu o “Git Bash Here”.

**Git status** -> Vê o status do projeto, incluindo em qual branch você está, o numero de commits, assim como os arquivos a serem comitados e os arquivos que você ainda não colocou no controle de versão.

**Git add “nome do arquivo.txt”** -> Adiciona o arquivo no controle de versão.

**Git add .** -> Adiciona todos os arquivos da pasta no controle de versão.

**Git commit “commit inicial”** -> Ele salva uma versão do seu código. Na primeira vez que você faz isso ele pede para configurar o git no computador, basta seguir o que ele fala para colocar o email e nome.

**Git remote add url\_do\_github\_exemplo** -> Da uma origem para onde vai ficar salvo na nuvem os seus arquivos.

**Git push ->** Envia o arquivo para a nuvem, porém na primeira vez você ainda não definiu para qual brench você vai estar enviando o seu codigo. Basta mandar o código disponibilizado após esse comando para mandar para o master. Se você fez um novo branch, basta mandar o código disponibilizado também.

**Git reflog ->** Vê as versões adicionas até o momento, ou seja, os commits. A versão do topo é a mais atual.

**Git reset --hard id\_da\_versão\_do\_reflog\_anterior** -> Volta para uma versão anterior do codigo na varsão local mesmo. Cuidado com o uso so mesmo, só usar após conersar com os demais.

**Git branch** -> Mostra os branchs do projeto, a verde com \* é a brench que você está atualmente.

**Git branch staging** -> Cria um branch chamado staging, ou qualquer outro nome que você der.

**Git checkout staging** -> Muda de branch, agora você está na branch staging.

**Git pull** -> Pega a versão mais atual do servidor para a maquina atual.

**Git merge staging** -> faz uma merge do local onde vc está (geralmente o para o master, portanto tem que fazer um git checkout para mudar para onde vc quer puxar as atualizações feitas no staging).