**Aluno:**  Bruno Camargo Manso

1. Diferencie Git de Github

O Git é um programa de controle de versão que contém comandos que permitem o gerenciamento de diferentes versões do mesmo código fonte mostrando a linha do tempo de cada ação efetuada. Seus comandos tem a capacidade de criar ou re-iniciar um repositório, de clonar tanto um código separadamente, como todo um conjunto deles. Tais códigos serão reconhecidos como objetos. O Git também pode: adicionar, mover, resetar, remover, bem como gerenciar logs, mostrar objetos contidos no projeto, ver o status do projeto, listar, criar bifurcações, gravar mudanças, mostrar a diferença entre scripts, concatenar scripts, mudar prioridade dos arquivos já *gitados*, remanejar gits, imputar tags e logicamente, baixar objetos e o remanejar em uma bifurcação específica, mandar objetos específicos para determinadas bifurcações ou puxar os mesmos para outras bifurcações (forks).

Já o Github é uma forma de armazenar todo conteúdo gerado pelo Git em nuvem, se tornando uma imensa comunidade colaborativa de softwares. A vantagem é que colaboradores em um determinado projeto podem, remotamente, alterar todo, ou apenas uma parte, do conteúdo. Assim há um sistema de gerenciamento de usuários no qual pode ser realizado alteração de permissões destes, limitando assim o acesso. Não apenas isso, mas sua importância é de armazenar repositórios contendo toda a estrutura de funcionamento de um software, percebe-se tal importância à medida que grandes empresas utilizam o Github para implementação, atualização, adaptação e otimização de seus processos.

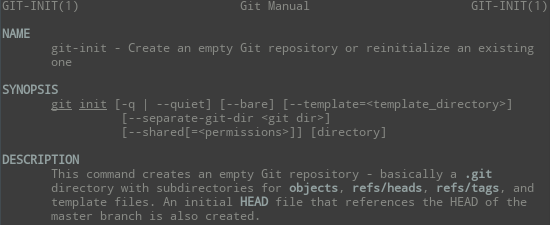
1. Descreva a função de cada um dos comandos abaixo:
   1. Version

Mostra a versão instalada na máquina, Git é nativo em sistemas linux.



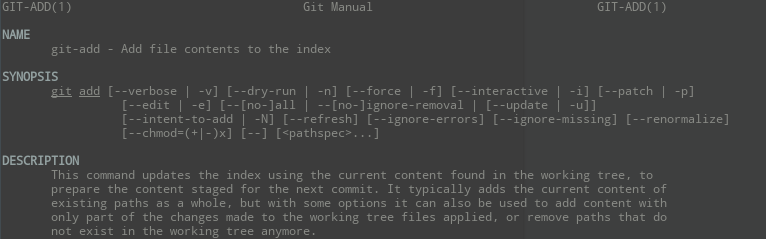
* 1. Init

Cria um repositório Git vazio ou reinicia um já existente. Basicamente um diretório .git com subdiretórios para objetos, refs/heads, refs/tags e modelos de arquivos. Um cabeçalho inicial que refere ao cabeçalho de uma bifurcação principal também será criado



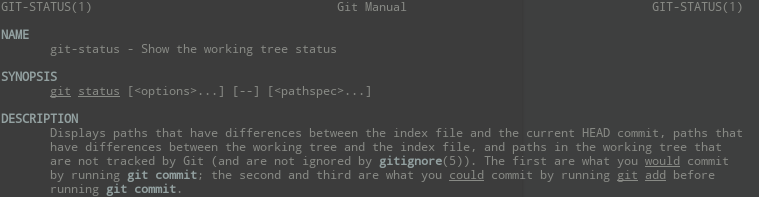
* 1. Add

Adiciona volume de arquivos no index. Esse comando atualiza um index usando o conteúdo corrente encontrado em uma árvore de arquivos funcional e prepara o volume organizando-o para um próximo envio. Normalmente adicionando todo o conteúdo e seus respectivos caminhos de uma só vez, porém utilizando as opções pode também ser usado para adicionar conteúdos com apenas parte de mudanças feitas para os arquivos de uma árvore de arquivos em funcionamento, ou remover caminhos que não existem na árvore de arquivos mais.



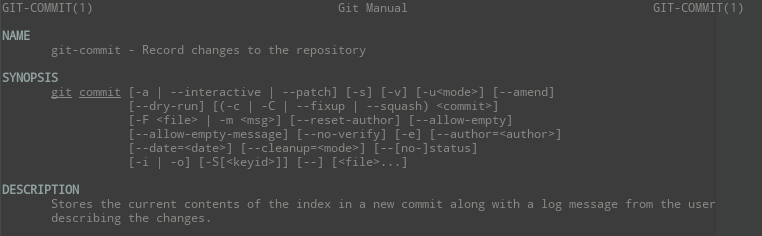
* 1. Status

Mostra o caminho que se diferenciam entre o arquivo de index e o atual cabeçalho, os caminhos que tem diferenças entre árvores funcionais e arquivo de index, caminhos na árvore funcional que ainda não foi verificada pelo Git. Primeiramente os que se *gostaria* que fossem enviados usando o git commit, o segundo e o terceiro são os que deveriam ser enviados pelo comando git add antes de usar o git commit.



* 1. Commit

Grava mudanças no repositório. Salva o volume atual de um *index* em um novo envio, juntamente com uma mensagem de registro do usuário descrevendo mudanças.



* 1. Log

Mostra os registros dos envios. O comando utiliza de opções aplicáveis para o comando git rev-list para controlar o que e como é mostrado, e suas opções aplicáveis ao comando git diff-\* de controle para controle de como mudanças cada introdução de envios são mostrados.

