Tutorial GitHub

Principais comandos do git:

git config -–list » Lista as configurações do Git, se estiver dentro do repositório, lista mais itens

git config -–global user.name "Meu Nome" » Define o nome de usuário para o Git

git config -–global user.email "email@dominio.com" » Define o e-mail de usuário para o Git (tem de ser o cadastrado no GitHub)

git config -–global core.editor vim » Define o editor de texto padrão para abrir automaticamente arquivos informados pelo Git

git init » Inicializa um repositório Git

git status » Vê o estado atual do projeto

git add arquivo.txt » Adiciona o arquivo arquivo.txt ao projeto

git commit -m "Minhas mudanças efetuadas" » Armazena as mudanças efetuadas e descreve o que foi alterado

git log » Mostra todas as mudanças que já foram efetuadas: commit, autor e data

git push -u origin master » Envia todos os arquivos modificados e “commitados” para o repositório no github

-u - faz com que o Git armazene esse comando e da próxima vez basta utilizarmos **git push**

origin- diz que o repositório no github possui o mesmo nome do projeto/diretório que você está enviando

master - é o nome da *branch*.

git pull origin master » Verifica as mudanças efetuadas por outros colaboradores do projeto

git diff HEAD » Verifica as partes dos arquivos alterados no último commit.

git reset arquivo.txt » Remove um arquivo do projeto

git checkout – arquivo.txt » Desfaz a última alteração feita num arquivo

git rm "*.txt" » Remove 1 ou mais arquivos utilizando “curinga”

git clone https://github.com/user/project.git » Copia um projeto pro seu PC

info git » Obtém a Documentação do git

man git » Obtém o Manual do git

Configurando o ambiente de acesso

Deve possuir o git instalado em sem computador e uma conta no github. Para baixar o git basta ir no site oficial selecionar o sistema operacional, baixar e seguir o guia de instalação.
Após a instalar é necessário configurar seu nome e email que serão exibidos em seus commits, para isso basta digitar no terminal os comandos a baixo:
git config --global user.name "Seu Nome"
git config --global user.email " email@xxxx.com"

Criando um novo repositório

Dentro da aplicação crie uma nova pasta e utilize o comando **git init** para criar um novo repositório.

Como obter um repositório

Crie uma cópia de trabalho em um repositório local executando o comando no terminal: **git clone /caminho/do/repositório**
Caso utilize um servidor remoto utilize o comando:
git clone user@servidor /caminho/do/repositorio

Fluxo de trabalho

Seus repositórios locais consistem em três "árvores" mantidas pelo git. A primeira delas é sua Working Directory que contém os arquivos vigentes. A segunda Index que funciona como uma área temporária e finalmente a terceira que é HEAD que aponta para o último commit que você fez.

Comandos adicionar & confirmar

Você pode propor mudanças (adicioná-las ao Index) usando
git add <arquivo> ou **git add *.**
Este é o primeiro passo no fluxo de trabalho básico do git. Para realmente confirmar estas mudanças (isto é, fazer um *commit*), use

git commit -m "comentários das alterações".
Agora o arquivo é enviado para o HEAD, mas ainda não para o repositório remoto.

Enviando alterações para o repositório

Suas alterações agora estão no HEAD da sua cópia de trabalho local. Para enviar estas alterações ao seu repositório remoto, execute
git push origin máster.
Altere master para qualquer ramo (branch) desejado, enviando suas alterações para ele.
Se você não clonou um repositório existente e quer conectar seu repositório a um servidor remoto, você deve adicioná-lo com o comando:
git remote add origin <servidor>
Agora você é capaz de enviar suas alterações para o servidor remoto que desejar.

Como fazer Ramificando

Branches são usados para desenvolver funcionalidades isoladas umas das outras. O branch master é o branch "padrão" quando você cria um repositório. Use outros branches para desenvolver e mescle-os ao branch master após a sua conclusão.
Crie um novo branch chamado "funcionalidade_x" e selecione-o usando **git checkout -b funci_x ,**retorne para o master usando **git checkout máster** e remova o branch da seguinte forma **git branch -d funci_x** um branch não está disponível a outros a menos que você envie o branch para seu repositório remoto **git push origin <funci_x>**

Como atualizar & mesclar

Para atualizar seu repositório local com a mais nova versão, execute
git pull na sua pasta de trabalho para *obter* e *fazer merge* (mesclar) alterações remotas.
Para fazer merge de um outro branch ao seu branch ativo (ex. master), use **git merge <branch>** em ambos os casos o git tenta fazer o merge das alterações automaticamente. Infelizmente, isto nem sempre é possível e resulta em *conflitos*. Você é responsável por fazer o merge
estes conflitos manualmente editando os arquivos exibidos pelo git. Depois de alterar, você precisa marcá-los como merged com **git add <arquivo>**
antes de fazer o merge das alterações, você pode também pré-visualizá-as usando **git diff <branch origem> <branch destino>.**

Como fazer rotulação

É recomendado criar rótulos para releases de software. Este é um conhecido conceito, que também existe no SVN. Você pode criar um novo rótulo chamado *1.0.0* executando o comando **git tag 1.0.0 1b2e1d63ff**
o **1b2e1d63ff** representa os 10 primeiros caracteres do id de commit que você quer referenciar com seu rótulo. Você pode obter o id de commit com **git log** você pode também usar menos caracteres do id de commit, ele somente precisa ser único.

Como sobrescrever alterações locais

No caso de você ter feito algo errado (que seguramente nunca acontece ;)) você pode sobrescrever as alterações locais usando o commando **git checkout -- <arquivo>** isto substitui as alterações na sua árvore de trabalho com o conteúdo mais recente no HEAD. Alterações já adicionadas ao index, bem como novos arquivos serão mantidos.
Se ao invés disso você deseja remover todas as alterações e commits locais, recupere o histórico mais recente do servidor e aponte para seu branch master local desta forma **git fetch origin** e **git reset --hard origin/master**