# Como usar Git e Github na prática: Guia para iniciantes | Mayk Brito

<https://www.youtube.com/watch?v=2alg7MQ6_sI>

Para unir suas branch’s, vai para a branch master, e

$ git merge nomedabranchquequerenviar

$ git show - mostra todas as alterações feitas detalhadamente.

Resumo prévio:

$ git nit // Inicia a linha do tempo, cria um repositório.

$ git add // Adiciona ou atualiza midanças para irem para a linha do tempo

$ git commit -am // Adiciona um ponto na linha do tempo

$ git log // Visualiza os pontos na linha do tempo

$ git status // Informa o estado das alterações do nosso projeto

$ git show // Apresenta determinado ponto na história

$ git branch //gerencia as branch, ou linhas do tempo

$ git checkout // Manipula as linhas do tempo

$ git merge // Uni as linhas do tempo(branch’s)

$ push // Envia as alterações locais para o repositório remoto

$ git clone // Clona um repositorio

$ git checkout 12313comitquequervoltar // desfaz para aquele momento da linha do tempo

$ git pull // Atualiza o repositório local de acordo com o remoto (Antes de fazer um push, fazer um pull primeiro)

Na pasta do repositório local, para criar um vinculo com repositório remoto usa-se:

$ git remote add http://endereço do repositorio

Depois da um

$ git push -u origin master (CRIANDO ASSIM O BRANCH MASTER)

Dicadokray: Para não ter que colocar o ususário ea senha da o comando:

$ git config credential.helper store

# CURSO COMPLETO DE GIT

<https://www.youtube.com/watch?v=OuOb1_qADBQ&t=5158s>

Timeline:

[04:41](https://www.youtube.com/watch?v=OuOb1_qADBQ&t=281s) - O que é, pra que serve, como funciona?

[15:10](https://www.youtube.com/watch?v=OuOb1_qADBQ&t=910s) - Instalando o GIT

[21:42](https://www.youtube.com/watch?v=OuOb1_qADBQ&t=1302s) - Configurando o GIT

[27:08](https://www.youtube.com/watch?v=OuOb1_qADBQ&t=1628s) - Iniciando um repositório

[36:24](https://www.youtube.com/watch?v=OuOb1_qADBQ&t=2184s) - Branch, Readme, Commit

[51:59](https://www.youtube.com/watch?v=OuOb1_qADBQ&t=3119s) - Revertendo Modificações

[01:04:53](https://www.youtube.com/watch?v=OuOb1_qADBQ&t=3893s) - Trabalhando com diferentes Branches

[01:17:29](https://www.youtube.com/watch?v=OuOb1_qADBQ&t=4649s) - Diferença entre arquivos

[01:27:21](https://www.youtube.com/watch?v=OuOb1_qADBQ&t=5241s) - Criando um repositório no Github

[01:33:10](https://www.youtube.com/watch?v=OuOb1_qADBQ&t=5590s) - Conectando repositório local ao remoto

[01:46:19](https://www.youtube.com/watch?v=OuOb1_qADBQ&t=6379s) - Fazendo alterações no repositório remoto

[01:55:41](https://www.youtube.com/watch?v=OuOb1_qADBQ&t=6941s) - Ignorando arquivos do repositório (gitignore)

[02:00:57](https://www.youtube.com/watch?v=OuOb1_qADBQ&t=7257s) - Revertendo sem perder o código

[02:06:57](https://www.youtube.com/watch?v=OuOb1_qADBQ&t=7617s) - Deletando branches locais e remotos

[02:13:39](https://www.youtube.com/watch?v=OuOb1_qADBQ&t=8019s) - Puxando alterações de outras pessoas (pull)

[02:20:34](https://www.youtube.com/watch?v=OuOb1_qADBQ&t=8434s) - Clonando repositórios remotos (clone)

[02:26:10](https://www.youtube.com/watch?v=OuOb1_qADBQ&t=8770s) - Contribuindo com outros repositórios (fork / pul request)

Aula 01 - Introdução:

1 - Voltar a aplicação ao que era.

2 - Compartilhamento do código.

Versionamento de código.

-----------------------------------------------------------------------------------------------------------------

Aula 02 - github e bitbucket

$ git --version

git version 2.20.1

$ apt-get git

---------------------------------------------------------------------------------------------------------------

Aula 03 -Configuração pessoal

$ git config --global user.name “Ricardo Maranhão”

$ git config --global user.name

Ricardo Maranhão

$ git config --global user.email "cramioscaner@gmail"

$ git config --global user.email

cramioscaner@gmail

$ git config --global core.editor

sub

$ git config --list

user.email=cranioscaner@gmail.com

user.name=Ricardo Maranhão

core.editor=sub

----------------------------------------------------------------------------------------------------------------------

Aula 04 - Criar nosso repositório

$ git init

Initialized empty Git repository in /home/ricardo/modulogit/.git/

------------------------------------------------------------------------------------------------------------------

Aula 05 - Desenvolvendo recursos do GIT:

Branches ("ramos") - versões do sistema (Master).

commit - preparar para mandar alterações. Apenas as modificações feitas.

$ git status

$ git log

$git add -A

$ git commit -m “Descrição”

---------------------------------------------------------------------------------------------------------------

Aula 06 - Fazendo alterações, voltando versões

$ git branch

\* master

Mostra qual o branch que estamos trabalhando.

revertendo modificações:

$ git reset --soft, mixed e/ou hard rash do arquivo que quero que fique

Criando e trabalhando com outros branches

Criando um novo branch:

$ git branch teste

Trocando para outro branche:

$ git checkout teste

$ git branch

\* master

teste

$ git checkout teste

Switched to branch 'teste'

$ git branch

master

\* teste

Aula 07 - Quais foram as alterações realmente feitas?

$ git diff

detalha exatamente o que foi feito, logo após as modificações, e antes de commitar.

$ git diff --name-only

Mostra o nome dos arquivos que foram modificados.

$ git checkout HEAD -- README.md

Volta as alterações feitas nesse branch(HEAD) -- e nesse arquivo.

$ git checkout HEAD -- README.md

Volta as alterações nesse arquivo.

$ git commit -am “Descrição das alterações” - commita confirmando, sem o ADD .

Aula 08 - Trazer as alterações para o github, online.

Criar o cadastro no github.

Aula 09 - Transferir o repositório local para o repositório remoto.

$ git push origin master

e-mail, e senha para confirmar

Aula 10 - Consolidar as alterações no off no repositório remoto.

Aula 11 - Como ignorar certos arquivos:

Caso, ocultar arquivos que estão no local serão ocultos no repositório remoto.

Cria-se o arquivo:

.gitignore

e escreve nele as extensões ou pastas que quer ocultar.

Aula 12 - git revert

Esse é importante: Finalmente como descommitar um commit:

$ git revert --no-edit codigodocomitquedesejaapagar1231323213213213

Aula 13 - Deletar branch remota:

Para enviar um branch local para o repositório remoto, 1- estando no branch que quer enviar:

$ git push origin teste (nome do branch a ser enviado)

Para deletar o repositório remoto:

$ git push origin :teste

Usa dois pontos para deletar o branch remoto.

Para deletar o branch local, sai dele indo para outro branch, e usa:

$ git branch -D teste

Aula 14 - git pull

Trazendo o remoto para o local.

Aula 15 - Clonar projetos

$ git clone blalalalal

Aula 16 - Faz um fork, clona altera e faz um push. E manda pelo pull request.