

Git - GitHub

Gustavo Guanabara (Curso em vídeo)

https://www.youtube.com/watch?v=xEko29OWILE&list=PLHz_AreHm4dm7ZULPAmadvNhH6vk9oNZA

CURSO COMPLETO DE GIT (2 HORAS E 30 MINUTOS)

https://www.youtube.com/watch?v=OuOb1_qADBQ&t=25s

Razões:

- > O mínimo que precisa é saber usar controle de versionamento (importantíssimo)
- > Todos os projetos hospedados no github podem ser manipulados
- > Importante para o próprio projeto, qualquer plano poderá utilizar desses recursos
- > É uma MÁQUINA DO TEMPO para código, planejamento e gerenciamento

O que é Git? O que é versionamento? - Curso de Git e GitHub

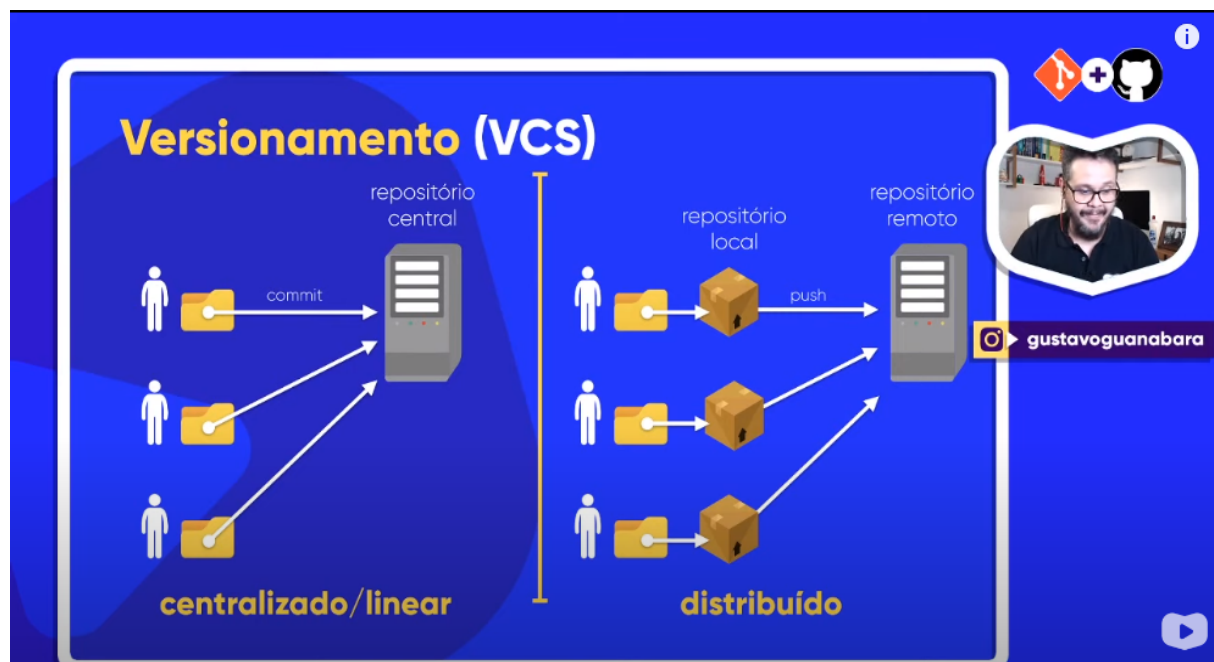
https://www.youtube.com/watch?v=xEKO29OWILE&list=PLHz_AreHm4dm7ZULPAmadvNhH6vk9oNZA&index=1

Git e Github são coisas diferentes

> **Git** é um software de controle de versão (VCS), muito importante para gerenciar completamente todo o projeto.

> **Github** é uma plataforma de redes sociais para PROGRAMADORES.

O software de versionamento do tipo distribuído (repositório local) permite que as diferentes versões sejam gravadas localmente. Porém, como se trata de salvar dados localmente é preciso armazená-los em um repositório remoto para que as informações se unam. Então se usa o comando **PUSH** que envia uma cópia desse repositório local para o repositório remoto.



(Nota da imagem acima: O versionamento distribuído é o sucessor do versionamento centralidade/linear).

Quais são as vantagens de usar software de controle de versão?

- 1) Controle de histórico das versões e acesso à elas, comparar versões, identificar alterações e responsáveis por elas
- 2) Trabalho em equipe

- 3) Ramificação do projeto, isso permite distribuir deveres através de segmentos no projeto. No final é possível juntar todas essas ramificações (comando **MERGE**) em um projeto principal, conhecido como BRANCH-MASTER
- 4) Segurança, alterações indesejadas entre diferentes segmentos não ocorrem.
- 5) Organização.

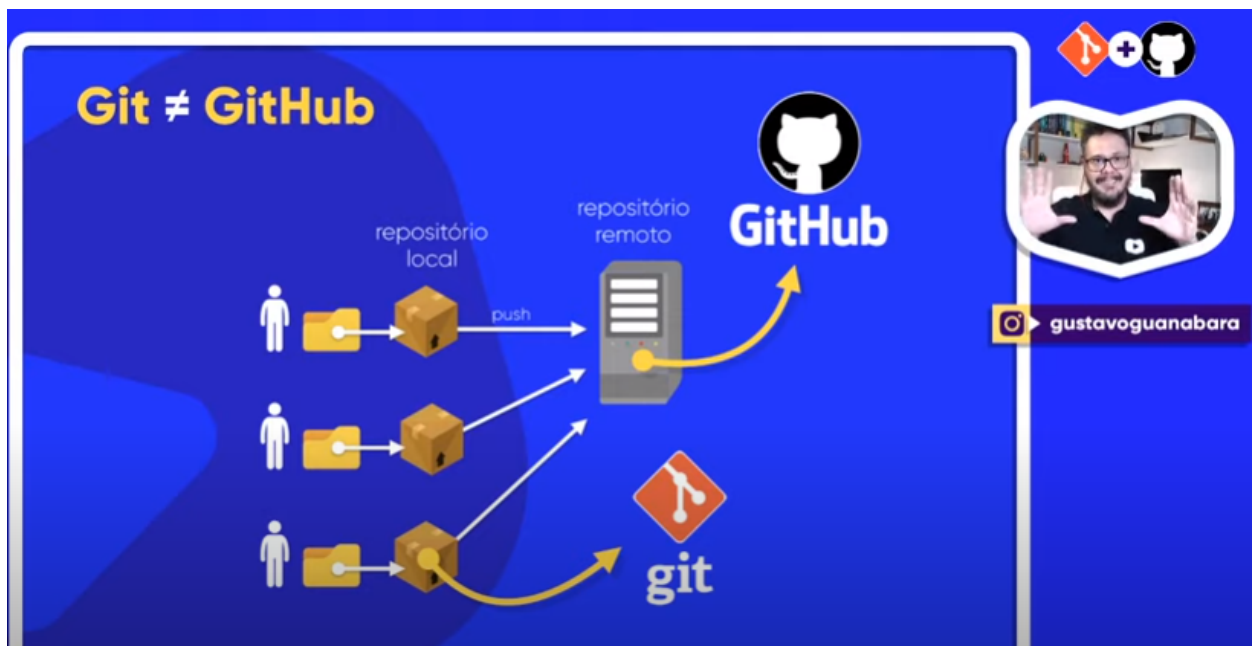
O que é GitHub? Pra que ele serve? - Curso de Git e GitHub

https://www.youtube.com/watch?v=hcZ0qtwvN1w&list=PLHz_AreHm4dm7ZULPAmadvNhH6vk9oNZA&index=2

O Git é o repositório local

O **Commit (ou commitar)** é fazer a alteração no código ainda em repositório local. Ele armazena os estados do código (como ele estava antes e depois). Ele faz a diferença entre as versões. É uma linha do tempo, exatamente tudo o que acontece ou aconteceu fica gravado em detalhes. Tudo isso dentro do repositório local, o Git.

O GitHub é um repositório remoto.



O GitHub tem:

- 1) Repositórios ilimitados (para projetos fechados não são permitidos mais do que três colaboradores na versão não paga).
- 2) Hospedagem de código fonte
- 3) Características de rede social
- 4) GitHub Pages integrado, permite que sites simples sejam hospedados.

- 5) Colaboração
- 6) Forks (realizar uma cópia de um projeto).



A Evolução do Git e GitHub - Curso de Git e GitHub

https://www.youtube.com/watch?v=CJtrNuTTs4Q&list=PLHz_AreHm4dm7ZULPAmadvNhH6vk9oNZA&index=4





O que quer dizer "Git"?

- › Três letras do alfabeto
- › Teimoso, cabeça-dura, pensa que sempre está certo
- › Global information tracker
- › Goddamn idiotic truckload of sh#t

gustavoguanabara

Guanabara: "A verdade é que 'git' é um comando de três letras que nenhum unix estava usando na época".

E o GitHub?

2008

→ **GitHub**

- ▶ Proprietário
- ▶ Hospedagem de código
- ▶ Baseado em **Git**
- ▶ 2009: **46k** repositórios
- ▶ 2010: **100k** users / **1M** repositórios
- ▶ 2011: Ultrapassou **SourceForge**
- ▶ 2013: **3M** users / **5M** repositórios
- ▶ 2016: **14º** lugar na lista da **Forbes**
- ▶ 2018: **Maior** ataque de **DDoS** da história



gustavoguanabara

E o GitHub?


2008

→ **GitHub**

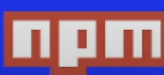
2018

→ **GitHub**

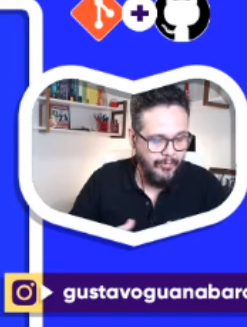
- ▶ Adquiridos pela **Microsoft**
- ▶ **US\$ 7.5 BILHÕES**
- ▶ Operação independente
- ▶ 2020: **GitHub compra a npm**



Microsoft



npm



gustavoguanabara

https://www.youtube.com/watch?v=gMh6lrXibWY&list=PLHz_AreHm4dm7ZULPAmadvNhH6vk9oNZA&index=4

(instalação)

Criando o primeiro Repositório - Curso de Git e GitHub

https://www.youtube.com/watch?v=5BYm7UdCrX0&list=PLHz_AreHm4dm7ZULPAmadvNhH6vk9oNZA&index=5

Aula Prática. Usamos o github desktop, junto com o editor de código Visual Code para criar, salvar, adicionar o tracker no projeto e então upar ele para o github. Letra U maiúscula no projeto aberto no visual code == não trackeado.

Clonando um Repositório - Curso de Git e GitHub

https://www.youtube.com/watch?v=OIArEishhQg&list=PLHz_AreHm4dm7ZULPAmadvNhH6vk9oNZA&index=7

Prática de clonar um projeto no repositório.

Para saber se os arquivos estão versionados, o arquivo oculto .git estará presente entre eles.

(19:00)

Versionando seus projetos antigos - Curso de Git e GitHub

https://www.youtube.com/watch?v=065NQCDSMb0&list=PLHz_AreHm4dm7ZULPAmadvNhH6vk9oNZA&index=8

Arquivo .gitignore significa exatamente o que o nome sugere, ele ignora qualquer arquivo depositado nele, assim nem o repositório local e nem o servidor manteriam esse arquivo.

Você sabe usar Issues? - Curso de Git e GitHub

https://www.youtube.com/watch?v=mUZOySyVcuQ&list=PLHz_AreHm4dm7ZULPAmadvNhH6vk9oNZA&index=9

Guanabara: “O que é Issue? É um problema, uma questão, um levantamento de alguma coisa.”

Exemplo: Encontramos um problema neste código e vamos falar dele com os desenvolvedores.

Guanabara: “Um pull request (também chamado de PR): encontrei um problema, preparei uma solução para esse problema, e quero mostrar essa solução para os desenvolvedores do programa”.

Dica do Guanabara: “Quando você se depara com um erro, você vai lá no repositório hospedado no github e procura pelo erro na aba das issues o seu mesmo problema, caso não consiga, é interessante abrir uma issue”. “Tudo isso serve para poupar tempo”.

O dono do repositório pode abrir issues também, como também encerrar definitivamente uma conversação de um issue (lock conversation).

É possível configurar respostas prontas, em saved replies

- 1) Personal settings
- 2) -> **Saved replies** <-

Guia da Linguagem Markdown - Curso de Git e GitHub

https://www.youtube.com/watch?v=LntSB-gl-ZI&list=PLHz_AreHm4dm7ZULPAmadvNhH6vk9oNZA&index=10

Markdown é uma linguagem de marcação, assim como HTML porém com foco na escrita de textos, os desenvolvedores do markdown defendem que este seja de mais fácil leitura que o HTML.

Vários sites e blogs (além do github) são compatíveis com o markdown.

Arquivos escritos em markdown tem extensão .md

Exemplos:

Arquivo.md

Arquivo2.md

README.md

Existem três área básicas dentro do github que faz uso do markdown:

-> Arquivo com extensão .md

-> Issues

-> Pull Requests

Explicação de como utilizar recursos do MARKDOWN.

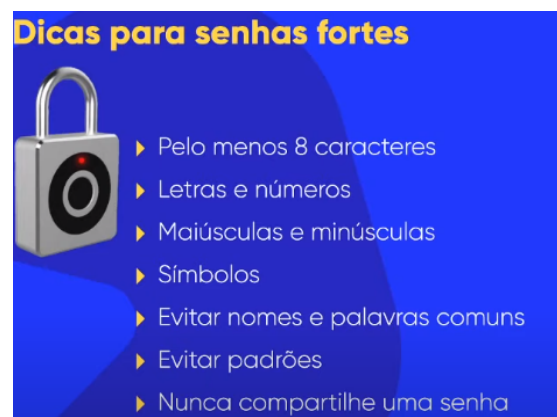
<https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet#headers>

Para utilizar lista de tarefas, para organizar o planejamento

- [] Quando a tarefa não estiver concluída.
- [x] Quando a tarefa estiver concluída.

Seu GitHub muito mais seguro - Curso de Git e GitHub

https://www.youtube.com/watch?v=bsl6P_IM_hg&list=PLHz_AreHm4dm7ZULPAmadvNhH6vk9oNZA&index=11



Git Branches de forma fácil e com exemplo - Curso de Git e GitHub

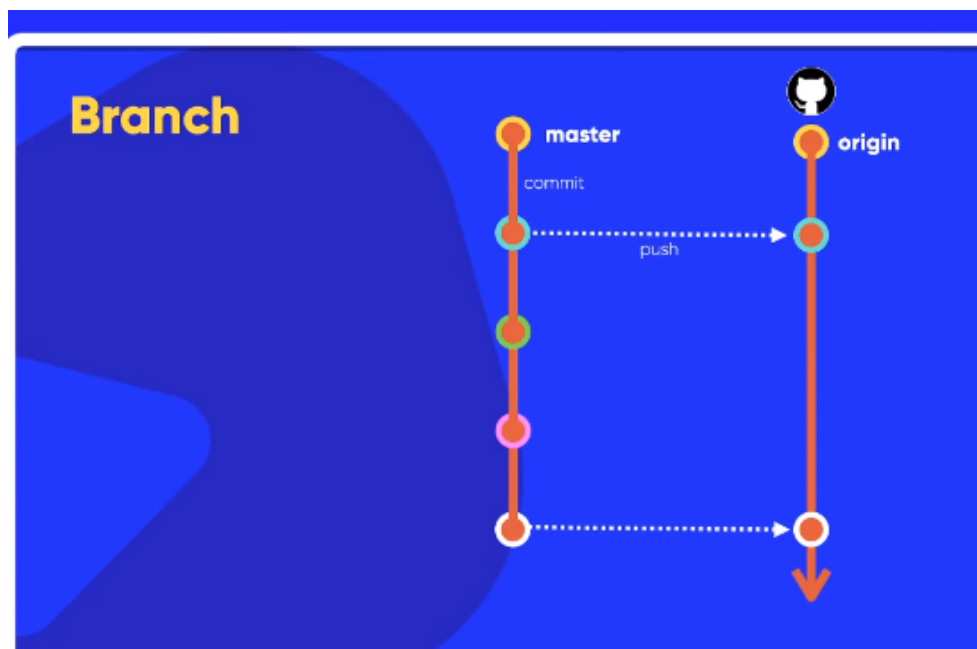
https://www.youtube.com/watch?v=xAOBQtSVI_k&list=PLHz_AreHm4dm7ZULPAmadvNhH6vk9oNZA&index=12



Branch são ramos paralelos.

O ramo principal é chamado de MASTER, esse é o ramo de produção.

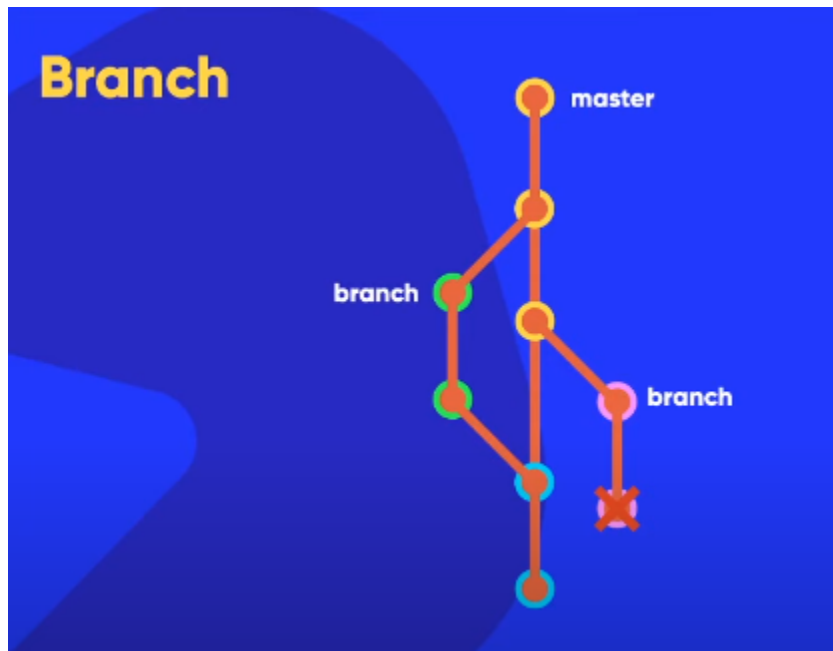
Origin é qualquer repositório remoto em que se encontra o projeto (nesse caso, o github).



Nota Guanabara: “Evite commitar TUDO na BRANCH MASTER”. Assim evita que erros sejam gravados no ramo principal.

No lugar dessa escolha, opte por criar uma branch nova e faça sua alteração.

A junção dessa branch paralela com o ramo MASTER se chama **MERGE**.



Encerrar uma branch não afeta a MASTER.