

MIN-006 – Git e GitHub

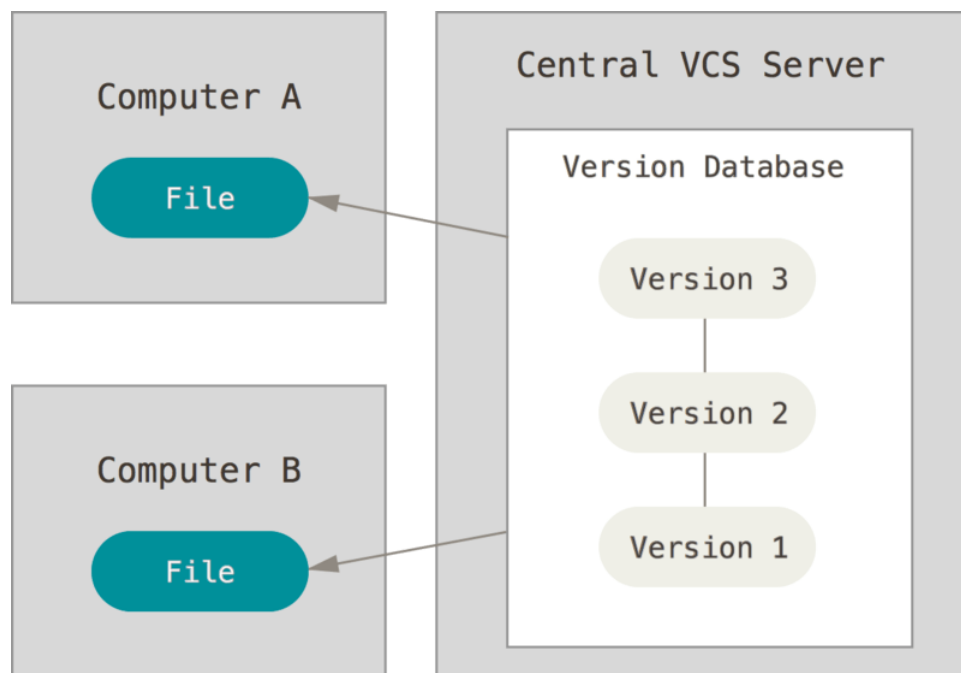
Júlia Euzébio da Silva Rodrigues
Ulisses Roque Tomaz

Controle de Versão

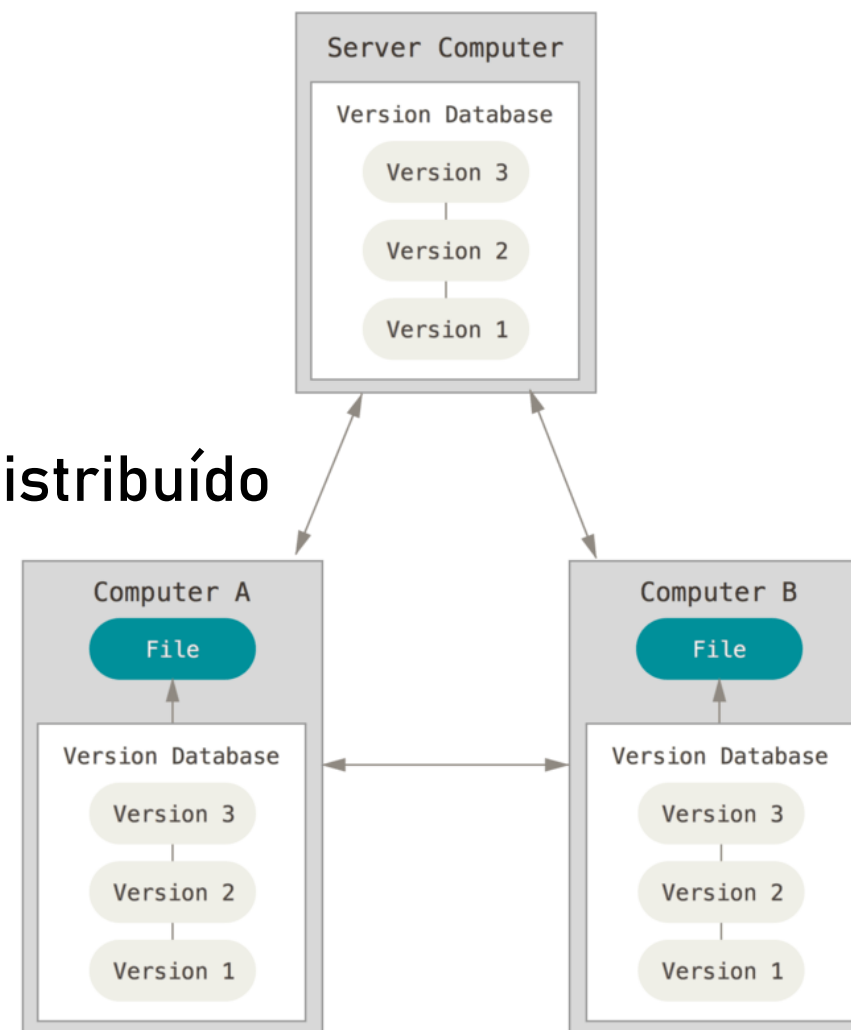
Sistema que registra as alterações realizadas em um arquivo ou conjunto de arquivos ao longo do tempo, possibilitando a recuperação de versões anteriores.

Controle de Versão

Centralizado



Distribuído



Sistemas de Controle de Versão



git



Sistemas de Controle de Versão



git



darcs



Sistemas de Controle de Versão



Centralizado

PERFORCE



git



darcs



Sistemas de Controle de Versão



Centralizado

PERFORCE



git



darcs



Sistemas de Controle de Versão



Centralizado

PERFORCE



git



Distribuído

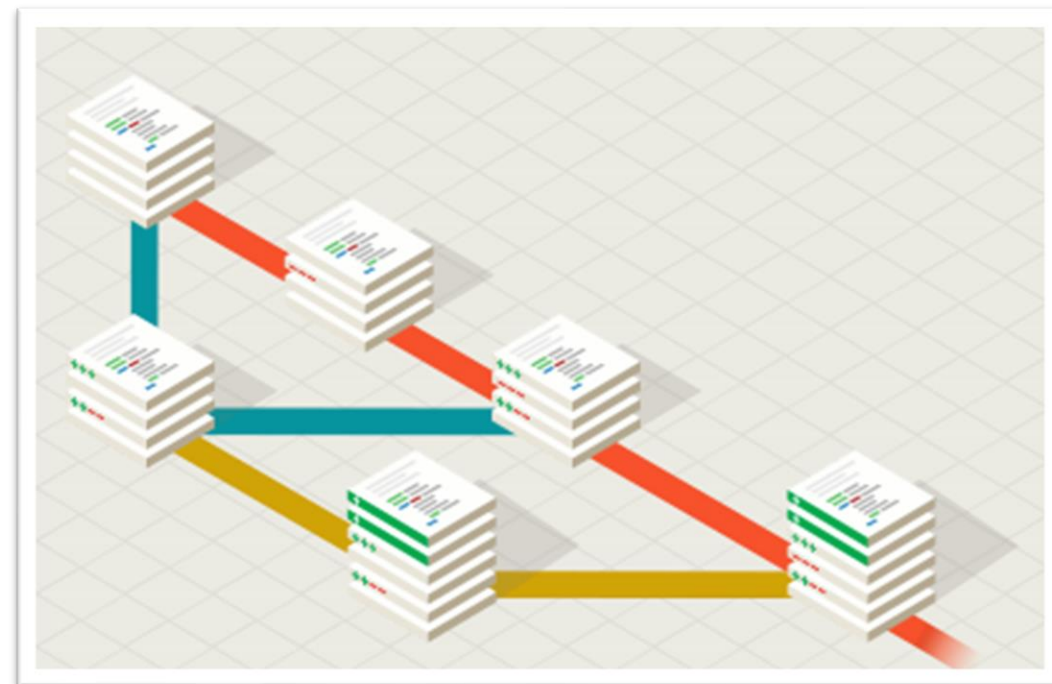


darcs





Criado em 2005 por Linus Torvalds, criador do Linux, para ser utilizado no controle de versão do *kernel* do Linux. Mantido, atualmente, por Junio C. Hamano.



Fonte: <https://git-scm.com/>



Companies & Projects Using Git

Google

FACEBOOK

Microsoft



LinkedIn

NETFLIX



Fonte: <https://git-scm.com/>

Instalação

Linux

Para instalar o Git no Linux, acesse:

<https://git-scm.com/download/linux>

Mac

Um instalador OSX Git é mantido e disponível para download no site do Git:

<https://git-scm.com/download/mac>

Windows

A compilação oficial está disponível para download no site.

<https://git-scm.com/download/win>

Instalação

Linux

Para instalar o Git no Linux, acesse:

<https://git-scm.com/download/linux>

Mac

Um instalador OSX Git é mantido e disponível para download no site do Git:

<https://git-scm.com/download/mac>

Windows

A compilação oficial está disponível para download no site.

<https://git-scm.com/download/win>

Git – Configurações básicas

```
$ git config --global user.name "Nome do Usuário"  
$ git config --global user.email nomedousuario@mail.com
```

Git – Criando um repositório local

```
$ git init
```

ou

```
$ git init nomedapasta
```

Em ambos os casos, após o comando ser executado um subdiretório oculto chamado `.git` será criado dentro da pasta.

```
Initialized empty Git repository in  
/c/Users/NomedoUsuário/nomedapasta/.git/
```

Git – Rastreando arquivos

```
$ git status
```

ou

```
$ git status -s
```


Git – Rastreando arquivos

Algo parecido com isto ...

On branch master

Initial commit

nothing to commit (create/copy files and use "git add" to track)

Caso o diretório esteja vazio.

Git – Rastreando arquivos

Após a criação de um arquivo no diretório a saída ...

On branch master

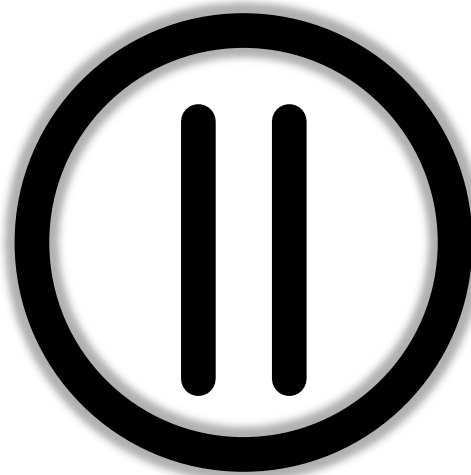
Initial commit

Untracked files:

(use "git add <file>..." to include in what will be committed)

index.php

nothing added to commit but untracked files present (use "git add" to track)



Simples assim...

Se por algum motivo quisermos parar de usar o Git, basta remover o diretório `.git`.



Git – Rastreando arquivos

Após a criação de um arquivo no diretório a saída ...

On branch master

Initial commit

Untracked files:

(use "git add <file>..." to include in what will be committed)

index.php

nothing added to commit but untracked files present (use "git add" to track)

Git – Rastreando arquivos

Após a criação de um arquivo no diretório a saída ...

On branch master

Initial commit

Untracked files:

(use "git add <file>..." to include in what will be committed)

index.php

nothing added to commit but untracked files present (use "git add" to track)

ARQUIVO NÃO
RASTREADO!

Git – Rastreando arquivos

Após a criação de um arquivo no diretório a saída ...

On branch master

Initial commit

Untracked files:

(use "git add <file>..." to include in what will be committed)

index.php

nothing added to commit but untracked files present (use "git add" to track)

ARQUIVO NÃO
RASTREADO!

Git – Rastreando arquivos

```
$ git add index.html
```

```
$ git add .
```

```
$ git add *.html
```

```
$ git add * (ignora os arquivos começados com '.')
```

Git – Rastreando arquivos

```
On branch master
```

```
No commits yet
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```
new file:   index.php
```

Git – Rastreando arquivos

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: index.php

ALTERAÇÕES PARA
CONFIRMAÇÃO

Git – Rastreando arquivos

On branch master

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: index.php

Nvovos arquivos foram adicionados no diretório de trabalho!

Git – Rastreando arquivos

```
$ git status
```

e depois ...

```
$ git add .
```

Git – Rastreando arquivos

On branch master
No commits yet

Changes to be committed:

(use "git rm -cached <file>..." to unstage)

new file: index.php

Untracked files:

(use "git add <file>..." to include in what will be committed)

css/estilo.css
conecta.php

ARQUIVOS NÃO
RASTREADOS!

Git – Rastreando arquivos

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: index.php
new file: css/estilo.css
new file: conecta.php

ALTERAÇÕES PARA
CONFIRMAÇÃO

Git – Rastreando arquivos

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: index.php
new file: css/estilo.css
new file: conecta.php

ALTERAÇÕES PARA
CONFIRMAÇÃO

Ops! O arquivo index.html precisou ser alterado...

Git – Rastreando arquivos

```
$ git status
```

e depois ...

```
$ git add .
```

Git – Rastreando arquivos

On branch master
No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: index.php
new file: css/estilo.css
new file: conecta.php

MODIFICAÇÕES NÃO PREPARADAS
PARA CONFIRMAÇÃO

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: index.php

Git – Rastreando arquivos

On branch master
No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: index.php
new file: css/estilo.css
new file: conecta.php

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: index.php

ARQUIVO MODIFICADO!

Git – Rastreando arquivos

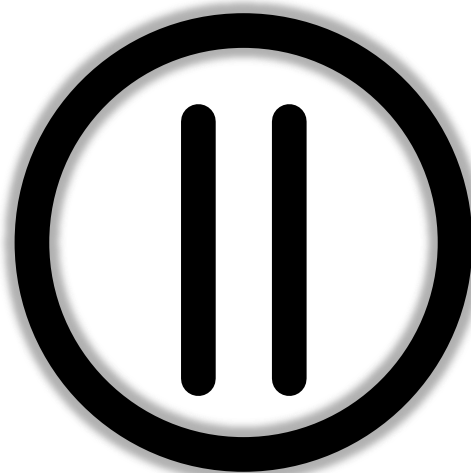
On branch master
No commits yet

Changes to be committed:

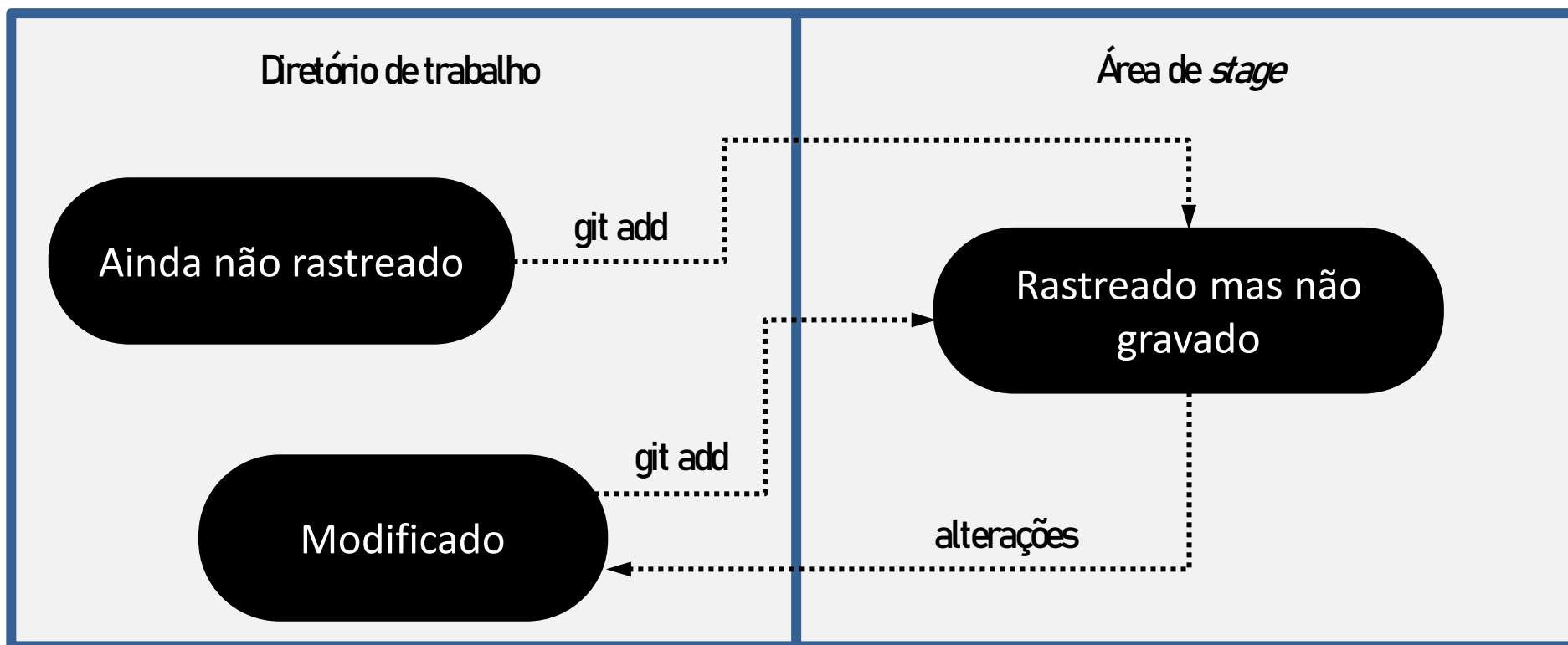
(use "git rm --cached <file>..." to unstage)

new file: index.php
new file: css/estilo.css
new file: conecta.php

ARQUIVOS RASTREADOS



Diretório de trabalho x área de *stage*





Git – Pontos na história, os *commits*

On branch master
No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: index.php
new file: css/estilo.css
new file: conecta.php

ARQUIVOS RASTREADOS

Git – Pontos na história, os *commits*

```
$ git commit -m “meu primeiro commit”
```

e depois... Podemos ver o status do nosso repositório com...

```
$ git status
```

Git – Visualizando os *commits*

```
$ git log
```

```
commit: ca82a6dfff817ec66f44342007202690a93763949  
Author: UlissesRTomaz <ulisses.rtomaz@gmail.com>  
Date: Mon Mar 17 21:52:11 2021 -0700
```

```
meu primeiro commit
```

```
$ git log --stat (estatísticas)
```

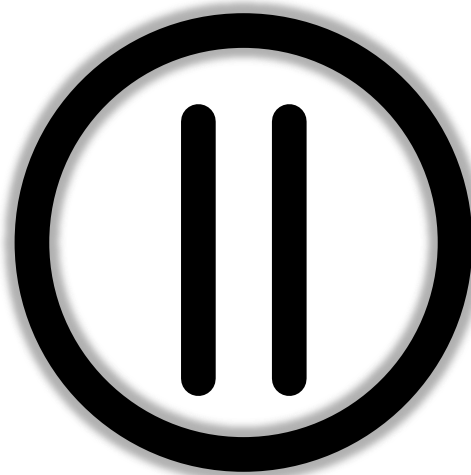
Git – Verificar mudança nos *commits*

```
$ git show <id_commit>
```

Para ver as alterações de um commit específico

```
$ git show
```

Para ver o último ponto na história



Git – Branchs, um ramo novo

“Se você, até aqui, já tem o poder de controlar o tempo,
agora terá o poder de criar universos paralelos”



Git – Branchs, um ramo novo

```
$ git branch <nome_da_branch>  
ex.: git branch design
```

depois de criada, usamos o *checkout* para mudarmos para essa branch.

```
$ git checkout design
```

Git – Branchs, um ramo novo

```
$ git branch
```

Lista todas as branch's

Podemos cria-las de forma mais rápida:

```
$ git checkout -b design
```

Daremos um git status para entendermos o que está acontecendo com o nosso novo ramo.

Git – Branchs, um ramo novo

```
$ git add <parametro>
```

Adicionaremos ele na branch

```
$ git commit -m “botão finalizado”
```

```
$ git status
```

```
$ git log
```

Git – Branchs, um ramo novo

Vamos voltar para o nosso ramo principal

```
$ git checkout master
```

Verifiquemos os arquivos deste ramo

```
$ ls -a
```

Git – Unindo ramos, merge nas branches

```
$ git merge design
```

Verifiquemos os arquivos deste ramo

```
$ git log
```

Para verificar a união das branches

Git – Deletando ramos

```
$ git branch -D design
```

O parâmetro -D é de Delete

```
$ git branch
```

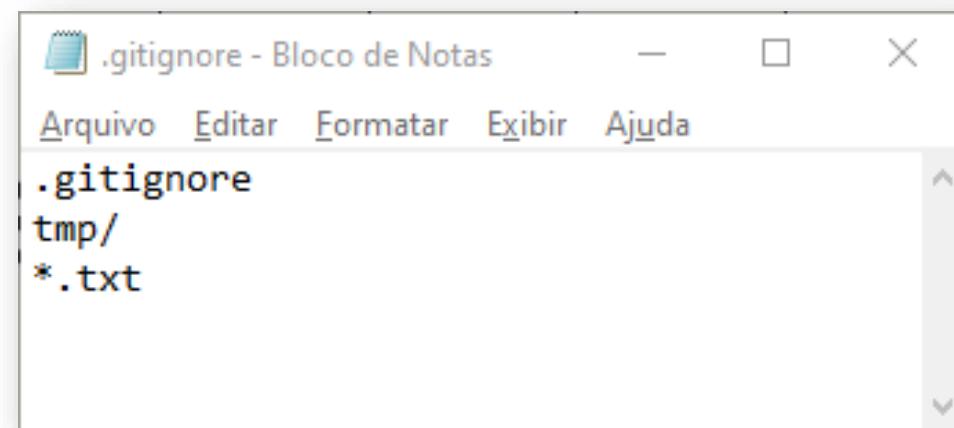
Para listar as branches

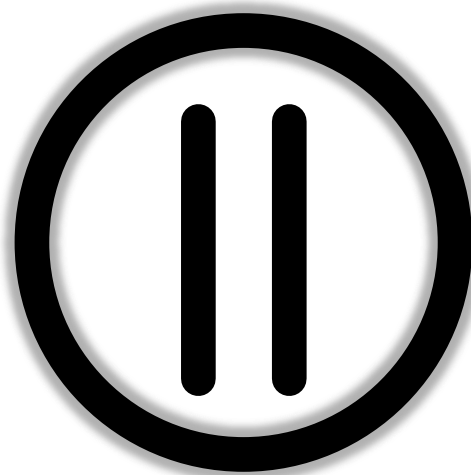
Git – Ignorando arquivos

Usando o .gitignore

```
$ ls -a
```

```
./      .git/  casa.txt      css/  todo.txt  
../     .gitignore  conecta.php  index.php  tmp/
```





Git

“O que já fizemos até agora?”



GitHub



GitHub

GitHub – Algumas curiosidades

- Em 28 de fevereiro de 2018, o GitHub foi vítima do segundo maior ataque de negação de serviço distribuído (DDoS) da história, com o tráfego de entrada chegando a um pico de cerca de 1,35 terabit por segundo.
- Em 4 de junho de 2018, a Microsoft anunciou que havia chegado a um acordo para adquirir o GitHub por US \$ 7,5 bilhões. [23]
A compra foi encerrada em 26 de outubro de 2018.

GitHub – Criando um repositório remoto

Já havíamos criado um repositório, mas apenas de forma local.
Vejam, agora, como funciona na nuvem!

GitHub – Adicionando um repositório remoto

```
$ git remote add origin <link_https_do_teu_repo>
```

```
$ git remote -v
```

Lista todos os repositórios remotos

GitHub – “Empurrando”

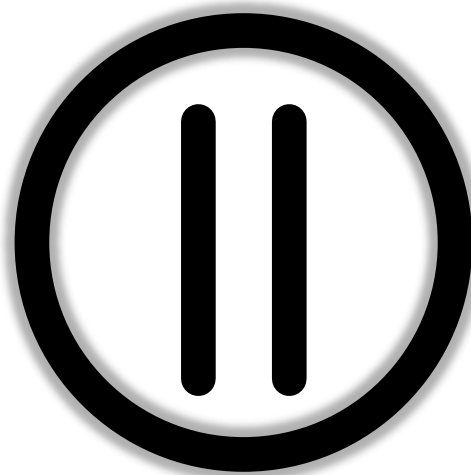
```
$ git push -u origin master
```

O ‘-u’ é de ‘--set-upstream’ e serve para criar o ramo master no repositório remoto. Usamos esse comando quando é a primeira vez que fazemos o push.

```
$ git push
```


GitHub – Clonando um repositório

```
$ git clone <link_https_do_repo_da_tua_equipe>
```



GitHub

“Deu um conflito! E agora!? Como resolveremos?”



GitHub – Conflitos

Vamos gerar um conflito proposital... Vamos criar o nosso ramo

```
$ git branch conflito && git checkout conflito
```

ou

```
$ git checkout -b conflito
```

GitHub – Conflitos

Vamos alterar um arquivo qualquer

```
$ git status
```

Para vermos o que acabamos de fazer

```
$ git add . && git commit -m “modifiquei o arquivo x”
```

ou

```
git commit -am “modifiquei o arquivo x”
```

Comando mais curto, usado para adicionar o arquivo modificado à trilha, isto é, de modo que esteja pronto para ser commitado

GitHub – Conflitos

Vamos voltar para o ramo principal, ou seja, a branch master

```
$ git checkout master
```

Agora vamos alterar a mesma linha que alteramos lá na branch de conflito.

```
$ git status
```

```
$ git add .
```

```
$ git commit -m “modifiquei o arquivo x dentro da master”
```

GitHub – Conflitos

Vamos unir as linhas do tempo, isto é, fazer um merge nas branches

Dentro da master, dou o comando:

```
$ git merge conflito
```


GitHub – Conflitos

Depois de resolvido todos os conflitos, vamos marcar o ponto no tempo

```
$ git status
```

```
$ git commit -am “resolvi o conflito”
```

```
$ git log
```

GitHub – Atualizar o repositório local

Vamos simular que alguém da nossa equipe de desenvolvimento tenha feito alguma alteração no repositório remoto.

Como puxaremos as atualizações para o nosso repositório local?

Por que é importante deixar sempre atualizado?

Ora, se quiséssemos enviar algo para nuvem, devemos nos certificar que o nosso repositório local está igual ao da nuvem, para que não haja conflitos!

```
$ git pull
```

E depois que fizermos o pull, podemos enviar com o push!

GitHub – Voltar no tempo

Vamos supor que precisamos recuperar um arquivo.

```
$ git log
```

Para verificarmos para onde iremos voltar

```
$ git checkout <id_commit> -- <arquivo_pra_recuperar>
```

o checkout manipula a linha do tempo

```
$ git status
```

```
$ git commit -am “restaurei o arquivo x”
```