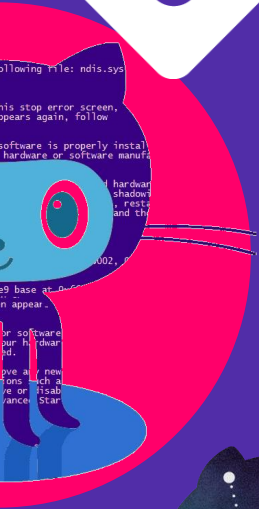


git x



Guia Git/Github

Como usar o GIT como GITHUB sem enrolação



“ Criando um novo repositório

crie uma nova pasta, abra-a e execute o comando

git init

para criar um novo repositório.



crie uma cópia de trabalho em um repositório local executando o comando

```
git clone /caminho/para/o/repositório
```

para criar um novo repositório.

quando usar um servidor remoto, seu comando será

```
git clone  
usuário@servidor:/caminho/para/o/re  
positório
```

Obtenha um

repositório



GitHub

Adicionar & confirmar



Você pode propor mudanças (adicioná-las ao **Index**) usando

```
git add <arquivo>
```

```
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

crie um novo branch chamado "funcionalidade_x" e selecione-o usando

```
git push origin master
```

retorne para o master usando

```
git checkout master
```

e remova o branch da seguinte forma

```
git branch -d funcionalidade_x
```

um branch *não está disponível a outros* a menos que você envie o branch para seu repositório remoto

```
git push origin <funcionalidade_x>
```



Ramificando

crie um novo branch chamado "funcionalidade_x" e selecione-o usando

```
git checkout -b  
funcionalidade_x
```

Este é o primeiro passo no fluxo de trabalho básico do git. Para realmente confirmar estas mudanças (isto é, fazer um *commit*), use

```
git remote add origin  
<servidor>
```

Agora o arquivo é enviado para o **HEAD**, mas ainda não para o repositório remoto.

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>