

Guia de uso do git/github

Repositório

O repositório é uma pasta onde ficam todos os arquivos do projeto e o histórico de revisão de cada arquivo.

Repositório local

Primeiro passo é criar uma pasta comando: `$ mkdir praticasgit`

```
Katiele@DESKTOP-37IMCOM MINGW64 ~  
$ mkdir praticasgit
```

Acessar a pasta comando: `cd praticasgit`

```
Katiele@DESKTOP-37IMCOM MINGW64 ~  
$ cd praticasgit  
  
Katiele@DESKTOP-37IMCOM MINGW64 ~/praticasgit  
$
```

Dentro da pasta tornamos ela um repositório com o comando `$ git init`

```
Katiele@DESKTOP-37IMCOM MINGW64 ~/praticasgit  
$ git init  
Initialized empty Git repository in C:/Users/Katiele/praticasgit/.git/  
  
Katiele@DESKTOP-37IMCOM MINGW64 ~/praticasgit (master)  
$
```

Obtendo um repositório existente

Clonar um repositório existente para repositório local com url com comando: `$ git clone https://github.com/KatieleSouto/engenharia-software.git`

```
Katiele@DESKTOP-37IMCOM MINGW64 ~/praticasgit/.git (GIT_DIR!)  
$ git clone https://github.com/KatieleSouto/engenharia-software.git  
Cloning into 'engenharia-software'...  
remote: Enumerating objects: 3, done.  
remote: Counting objects: 100% (3/3), done.  
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0  
Receiving objects: 100% (3/3), done.  
  
Katiele@DESKTOP-37IMCOM MINGW64 ~/praticasgit/.git (GIT_DIR!)  
$
```

Add e Commit

Armazena o conteúdo atual do índice em um novo commit, juntamente com uma mensagem de registro do usuário que descreve as mudanças.

Primeiro usamos o comando `$ git add .`

```
Katiele@DESKTOP-37IMCOM MINGW64 /c/AulaPratica (master)  
$ git add .
```

Após fizemos o comando `$ git commit -m "Atividade 1"`

```
Katielle@DESKTOP-37IMCOM MINGW64 /c/AulaPratica (master)
$ git commit -m "Atividade 1"
[master (root-commit) 3e03637] Atividade 1
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Atividade - Diagrama de caso de uso Sindicato.pdf
```

Push

Comando em que você transfere commits a partir do seu repositório local para um repositório remoto

```
Katielle@DESKTOP-37IMCOM MINGW64 /c/AulaPratica (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
```

Pull Request

Pull request é um mecanismo onde um desenvolvedor pode gerar uma notificação que sinaliza a conclusão do desenvolvimento de uma feature qualquer. Isso permite que todos os envolvidos saibam que precisam fazer o merge do código na branch principal.

```
~/cli/cli$ gh pr create

Creating pull request for octocat:docs into master in cli/cli

? Title Update docs
? Body <Received>
? What's next? [Use arrows to move, type to filter]
> Preview in browser
  Submit
  Cancel
```

Branch

É um ponteiro móvel para um dos commits, branch principal (master branch) que aponta para o último commit que você fez. Cada vez que você faz um commit ele avança automaticamente.

```
MINGW32:/c/Users/ /PET/GITHUB
~/PET/GITHUB (master)
$ git branch <nome_do_branch>
```

Merge

Combina várias sequências de commits em um histórico unificado. Nos casos de uso mais frequentes, o git merge é utilizado para combinar dois branches.

```
git checkout master  
git merge nome_branch
```