

Limpendo arquivos untracked (Não adicionados) - git clean

quarta-feira, 28 de junho de 2023

16:25

- Funciona quando inserimos vários arquivos na pasta do projeto porém não queremos mais utilizar sem ficar apagando uma por uma
- Usar quando os arquivos ainda não foram adicionados com o **git add**
- Utilizado também quando temos vários arquivos gerados automaticamente e não estão no .gitignore ainda
- Deve-se usar o comando **git add** em arquivos que iremos precisar pro código-fonte antes de utilizar o git clean nos arquivos untrackeds.
- Usa-se o comando **git clean** ou **git clean -f**

Excluindo arquivos não necessários

```
C:\sacbd\77ed0c22 novabranh -> novabranh
PS D:\github_teste\repo_teste2\pasta1> git status
On branch novabranh
Your branch is up to date with 'origin/novabranh'.

nothing to commit, working tree clean
● PS D:\github_teste\repo_teste2\pasta1> git status
On branch novabranh
Your branch is up to date with 'origin/novabranh'.

● Untracked files:
  (use "git add <file>..." to include in what will be committed)
      aad.lo
      g.log

nothing added to commit but untracked files present (use "git add" to track)
PS D:\github_teste\repo_teste2\pasta1> git clean
fatal: clean.requireForce defaults to true and neither -i, -n, nor -f given; refusing to clean
⊗ PS D:\github_teste\repo_teste2\pasta1> git clean -f
Removing aad.lo
● Removing g.log
```

Otimizando repositório - git gc

quarta-feira, 28 de junho de 2023

16:36

- É uma abreviação de garbage collector
- Identifica arquivos desnecessários e exclui
- Otimiza o repo em relação a performance
- Utiliza-se o comando **git gc**

Limpa o repo em relação a performance

```
PS D:\github_teste\repo_teste2\pasta1> git gc
Enumerating objects: 68, done.
Counting objects: 100% (68/68), done.
Delta compression using up to 8 threads
Compressing objects: 100% (41/41), done.
Writing objects: 100% (68/68), done.
Total 68 (delta 20), reused 48 (delta 12), pack-reused 0
PS D:\github_teste\repo_teste2\pasta1>
```

Verificando integridade dos arquivos do projeto - git fsck

quarta-feira, 28 de junho de 2023

16:59

- Utiliza-se para verificar se existe algum arquivo corrompido dentro do nosso projeto
- Deve-se usar o comando **git fsck**

```
PS D:\github_teste\repo_teste2\pasta1> git fsck
Checking object directories: 100% (256/256), done.
Checking objects: 100% (68/68), done.
Verifying commits in commit graph: 100% (26/26), done.
```

Reflog- git reflog

quarta-feira, 28 de junho de 2023

17:14

- Parecido com o log mostra todas as alterações feitas, desde a criação de branches quanto todas os commits feitos no projeto
- Deve-se usar o comando **git reflog**
- Com isso podemos usar o reflog com o reset para ir para uma modificação específica com o **git reset --hard numerolog**

Usando o git reflog com o git reset --hard

```
PS D:\github_teste\repo_teste2\pasta1> git reflog
5affb97 (HEAD -> main, origin/main, origin/HEAD) HEAD@{0}: checkout: moving from novabranch to main
7ed6c22 (origin/novabranch, novabranch) HEAD@{1}: commit: Enviando arquivos
e5314a4 HEAD@{2}: merge main: Merge made by the 'ort' strategy.
c78acbb HEAD@{3}: commit: adicionadno arquivo
95300ab HEAD@{4}: checkout: moving from main to novabranch
5affb97 (HEAD -> main, origin/main, origin/HEAD) HEAD@{5}: checkout: moving from BRanch to main
ad00063 (origin/BRanch, BRanch) HEAD@{6}: commit: Criando projeto no submodulo
082e88d HEAD@{7}: checkout: moving from main to BRanch
5affb97 (HEAD -> main, origin/main, origin/HEAD) HEAD@{8}: commit: Criando arquivo no teste
082e88d HEAD@{9}: merge BRanch: Fast-forward
95300ab HEAD@{10}: checkout: moving from BRanch to main
082e88d HEAD@{11}: commit: Enviando
95300ab HEAD@{12}: checkout: moving from novabranch to BRanch
95300ab HEAD@{13}: checkout: moving from main to novabranch
95300ab HEAD@{14}: checkout: moving from novabranch to main
95300ab HEAD@{15}: checkout: moving from main to novabranch
95300ab HEAD@{16}: clone: from https://github.com/Geronimonetto/repo_teste.git
PS D:\github_teste\repo_teste2\pasta1> git reset --hard 95300ab
warning: unable to rmdir 'submodulo': Directory not empty
HEAD is now at 95300ab Criando pasta2
```








Comprimindo o repositório - git archive --format zip output master_files.zip master

quarta-feira, 28 de junho de 2023 17:26

- Usa-se o comando para comprimir nosso projeto em um arquivo zip e enviar para outros devs ou para alguém em específico
- Deve-se usar o comando git archive --format zip --output nomedoarquivo.zip master

Zipando o projeto

```
nothing to commit, working tree clean
PS D:\github_teste\repo_teste2\pasta1> git archive --format zip --output arquivo.zip main
```

	.gitignore	28/06/2023 14:14	Documento de Te...	1 KB
	alow	28/06/2023 14:14	Documento de Te...	1 KB
	arquivo	28/06/2023 17:58	Arquivo ZIP do Wi...	1 KB
	arquivo_novo	28/06/2023 14:14	Documento de Te...	1 KB
	oi	28/06/2023 14:14	Documento de Te...	1 KB
	oii	28/06/2023 14:14	Documento de Te...	1 KB
	README	28/06/2023 14:14	Arquivo Fonte Ma...	1 KB