

Alunos: Eduardo da Silva Lamim

Wagner Lopes

Conceitos:

### 1. Configurações de valores do usuário

Um dos comandos git mais usados é o git config que pode ser usado para definir valores de configuração específicos do usuário, como e-mail, algoritmo preferido para diff, nome de usuário e formato de arquivo etc. (HERTEL, 2017).

Comando: `$ git config --global`

### 2. Criação de um repositório

Todo repositório Git armazena as informações dentro de uma pasta oculta chamada `“/.git”`. Para que os arquivos de uma pasta possam ser versionados pelo Git, é preciso iniciar o repositório. (HANASHIRO, 2019).

Comando: `$ git init`

### 3. Desfazendo a última alteração

Caso você tenha feito alterações e já tenha chegado a realizar um commit, para desfazer é necessário usar o revert. Será criado um commit indicando que o último commit foi desfeito. Esse comando apaga novos arquivos. (HANASHIRO, 2019).

Comando: `$ git revert HEAD`

#### 4. Download de um repositório

Conforme informa Roger Dudler (2017):

“Para criar uma cópia de trabalho em um repositório local execute o comando “git clone /caminho/para/o/repositório” quando usar um servidor remoto, seu comando será “git clone usuário@servidor:/caminho/para/o/repositório”

Comando: `$ git clone`

#### 5. Enviando alterações para o servidor

Suas alterações agora estão no HEAD da sua cópia de trabalho local. Para enviar estas alterações ao seu repositório remoto, execute `git push origin master`.

Altere master para qualquer ramo (branch) desejado, enviando suas alterações para ele. (DUDLER, 2017).

Comando: `$ git push origin master`

#### 6. Excluir um repositório

Conforme Akira Hanashiro (2019) explica:

Há momentos em que não queremos apagar nossos arquivos, mas queremos remover as informações sobre aquele repositório criado com o `$ git init` (talvez criar um repositório novo com os mesmos arquivos).

Para isso não usamos o Git. Lembre-se que um repositório Git armazena as informações dentro de uma pasta oculta chamada `/.git`. Então basta apagar esta pasta oculta que o seu atual diretório deixará de ser um repositório

Comando: `$ rm -rf .git`

## 7. Listando arquivos modificados

“Esse comando indica o estado do seu repositório. Em outras palavras, ele vai listar todos os arquivos que foram modificados desde o seu último commit.” (HANASHIRO, 2019).

Comando: `$ git status`

## 8. Revertendo arquivos para o último “commit” realizado

Se o arquivo foi modificado, mas você ainda não executou git add, um simples git checkout removerá as alterações, deixando o arquivo como ele estava no último commit. Passe o nome do arquivo a ter as alterações desfeitas ou . para desfazer as alterações em todos os arquivos modificados. Muito útil se você está apenas experimentando um código, mas não quer que ele seja salvo. (HANASHIRO, 2019).

Comando: `$ git checkout .`

## 9. Salvando alterações

[...] ainda não estamos persistindo os dados no histórico do Git, mas adicionando a uma área temporária onde podemos ficar levando e trazendo alterações até garantirmos que algo realmente deve ser salvo, então rodamos o git commit.

Para fazer um commit, precisamos adicionar uma mensagem ao pacote, então rodamos com o parâmetro -m "mensagem". (OLIVEIRA, 2018).

Comando: `$ git commit -m “mensagem”`

## 10. Verificando os arquivos que foram alterados

Conforme explica Rafael Hertel (2017): “O comando git status exibe a lista de arquivos alterados juntamente com os arquivos que ainda não foram adicionados ou confirmados.”

Comando: `$ git status`

## Referências:

OLIVEIRA, William. **COMANDOS MAIS UTILIZADOS NO GIT**. 2018.

Disponível em: <https://woliveiras.com.br/posts/comandos-mais-utilizados-no-git/>.

Acesso em: 15 abr. 2020.

HANASHIRO, Akira. **Comandos do Git que você precisa conhecer – Parte 1**. 2019.

Disponível em: <https://www.treinaweb.com.br/blog/comandos-do-git-que-voce-precisa-conhecer-parte-1/>. Acesso em: 15 abr. 2020.

HERTEL, Rafael. **Comandos Básicos de GIT**. 2017. Disponível em:

<https://www.hostinger.com.br/tutoriais/comandos-basicos-de-git/>. Acesso em: 15 abr. 2020.

DUDLER, Roger. **Git - guia prático**. 2017. Disponível em:

[https://rogerdudler.github.io/git-guide/index.pt\\_BR.html](https://rogerdudler.github.io/git-guide/index.pt_BR.html). Acesso em: 15 abr. 2020.

Link dos repositórios:

Eduardo da Silva Lamim: [https://github.com/Eduardo-Lamim/Aula\\_Mobile.git](https://github.com/Eduardo-Lamim/Aula_Mobile.git)

Wagner Lopes: [https://github.com/wagnerlopes-univ/Aula\\_Mobile](https://github.com/wagnerlopes-univ/Aula_Mobile)