DISCENTE: Caio Gonzaga Bernils
REPOSITÓRIO GIT: https://github.com/caiogbernils/Aula_Mobile

## 1. GIT APPLY

Segundo CHACON (2020), esse comando permite ler a saída do *diff* atual e aplicar uma nova atualização (*patch*) sobre o arquivo. Este comando aplica o *patch,* porém não cria um *commit*.

## 2. GIT BRANCH

Através do comando *git branch* é possível realizar o gerenciamento de ramificações (*branchs*) dentro de um repositório Git, de modo a alterar arquivos separadamente da ramificação padrão *master*. Um *branch* não estará disponível a outros se não for enviado para o repositório remoto. (DUDLER, 2017).

## 3. GIT CLEAN

O comando *git clean* realiza uma limpeza na árvore de trabalho, removendo os arquivos que não estão sob controle de versão. Geralmente, apenas os arquivos desconhecidos para o *Git* são removidos. (CHACON; STRAUB, 2020).

## 4. GIT COMMIT

Esse comando permite criar um *commit*. O *commit* permite enviar a confirmação dos dados que estão no index, adicionando-os ao *head*, porém esses arquivos ainda não estarão no repositório remoto. (DUDLER, 2017).

## 5. GIT DIFF

Segundo STRAUB (2020), o comando git diff é utilizado para identificar as mudanças entre a árvore de trabalho e o index. O comando git diff é freqüentemente usado junto com o git status e o git log para analisar o estado atual de um repositório Git.

## 6. GIT FETCH

O comando *git fetch* é semelhar ao comando *git pull*, porém diferente desse último os novos dados vindos do repositório remoto não são mesclados automaticamente, devendo, portanto, serem mesclados manualmente (CHACON; STRAUB, 2020).

## 7. GIT REMOTE

Utilizando o comando *git remote* é possível criar, visualizar e excluir conexões com outros repositórios remotos. Repositórios remotos são versões do seu projeto hospedadas na Internet ou na rede em algum lugar. (CHACON; STRAUB, 2020).

## 8. GIT REVERT

Conforme dito por CHACON (2020), utilizando o comando *git revert* é possível reverter alterações introduzidas por determinado patch. *Git revert* é usado para registrar novos *commits* de modo a reverter o efeito de algum commit anterior.

## 9. GIT SHOW

O comando git show é utilizado para visualizar informações sobre qualquer objeto git. (HERTEL, 2017).

Dependendo do modo como for utilizado, apresenta modos e informações diferentes para cada comando. Exemplo, para *commits* mostra o *log* e as *diffs*, para *tags* mostra a mensagem da *tag* e objetos referenciados e para árvores, mostra o nome. (CHACON; STRAUB, 2020).

## 10. GIT STASH

Segundo HERTEL (2017), o comando *git stash* é um dos comandos básicos menos conhecidos, através dele é possível salvar alterações que não devem ser cometidas imediatamente em uma base temporária.

## REFERÊNCIAS


DUDLER, Roger. **Git - guia prático**: apenas um guia prático para começar com git. sem complicação ;). 2017. Disponível em: https://rogerdudler.github.io/git-guide/index.pt_BR.html. Acesso em: 14 abr. 2020.

CHACON, Scott; STRAUB, Ben. **Pro Git**. 2. ed. Mountain View: Creative Commons, 2020. 507 p. Disponível em: https://git-scm.com/book/pt-br/v2; https://www.git-scm.com/docs. Acesso em: 14 abr. 2020.

HERTEL, Rafael. **Comandos Básicos de Git**, 2017**.** Disponível em: https://www.hostinger.com.br/tutoriais/comandos-basicos-de-git/ . Acesso em: 14 abr. 2020.