

**git add<arquivos..>:** Este comando adiciona o(s) arquivo(s) em um lugar que chamamos de INDEX, que funciona como uma área do git no qual os arquivos possam ser enviados ao Github. É importante saber que ADD não está adicionando um arquivo novo ao repositório, mas sim dizendo que o arquivo (sendo novo ou não) está sendo preparado para entrar na próxima revisão do repositório.(SCHMITZ, 2015)

**git blame:** Anota cada linha no arquivo fornecido com informações da revisão que modificou a linha pela última vez. Opcionalmente, comece a anotar a partir da revisão fornecida. (GIT, 2020)

**git clean:** Limpa a árvore de trabalho removendo recursivamente os arquivos que não estão sob controle de versão, iniciando no diretório atual. (GIT, 2020)

**git commit -m “comentário”:** Este comando realiza o que chamamos de “commit”, que significa pegar todos os arquivos que estão naquele lugar INDEX que o comando add adicionou e criar uma revisão com um número e um comentário, que será vista por todos. (SCHMITZ, 2015)

**git daemon:** Um daemon TCP Git realmente simples que normalmente escuta na porta "DEFAULT\_GIT\_PORT" aka 9418. Aguarda uma conexão solicitando um serviço e servirá esse serviço se estiver ativado. (GIT, 2020)

**git hash-object:** Calcula o valor do ID do objeto para um objeto com tipo especificado com o conteúdo do arquivo nomeado (que pode estar fora da árvore de trabalho) e, opcionalmente, grava o objeto resultante no banco de dados do objeto. Informa o ID do objeto na saída padrão. Quando <type> não é especificado, o padrão é "blob". (GIT, 2020)

**git init:** Para transformar qualquer diretório em um repositório GIT, o simples comando git init <directory> pode ser utilizado. Uma pasta chamada .git também deve começar a existir no diretório em que o comando foi executado. (HERTEL, 2017)

**git fetch:** Busque ramificações e / ou tags (coletivamente, "refs") de um ou mais outros repositórios, juntamente com os objetos necessários para completar seus históricos. (GIT, 2020)**git pull:** (...) existe um comando importante neste processo, que é o git pull. Ele é usado para trazer todas as modificações que estão no github para o seu projeto local. Isso é vital quando existem projetos mantidos por mais de uma pessoa, ou se você possui duas máquinas e precisa manter a sincronia entre elas. (SCHMITZ, 2015)

**git status:** Para transformar qualquer diretório em um repositório GIT, o simples comando git init <directory> pode ser utilizado. Uma pasta chamada .git também deve começar a existir no diretório em que o comando foi executado. (HERTEL, 2017)

**git show:** Mostra um ou mais objetos (blobs, árvores, tags e confirmações). Para confirmações, mostra a mensagem de log e a diferença de texto. Ele também apresenta o commit de mesclagem em um formato especial, produzido pelo git diff-tree --cc. (GIT, 2020)

## **LINK REPOSITÓRIO GIT:**

Kaíque: [https://github.com/kaiqw/Aula\\_mobile.git](https://github.com/kaiqw/Aula_mobile.git)

Marcelo: [https://github.com/marceloito/Aula\\_Mobile](https://github.com/marceloito/Aula_Mobile)

## **REFERÊNCIAS:**

HERTEL, Rafael. **Comandos Básicos de GIT**. 2017. Disponível em: <https://www.hostinger.com.br/tutoriais/comandos-basicos-de-git/>. Acesso em: 21 abr. 2020.

SCHMITZ, Daniel. **Tudo que você queria saber sobre Git e GitHub, mas tinha vergonha de perguntar**. 2015. Disponível em: <https://tableless.com.br/tudo-que-voce-queria-saber-sobre-git-e-github-mas-tinha-vergonha-de-perguntar/>. Acesso em: 21 abr. 2020.

GIT. Disponível em: <https://git-scm.com/docs>. Acesso em: 20 abr. 2020.