Treinando comandos Git

Considerando que o git tenha sido instalado, é necessário fazer as configurações iniciais, o mais comum é configurar usuário e email globalmente

$ git config --global user.name "nome" (configura o nome usuário local)

$ git config --global user.email "email"(este é muito importante pois é o que liga com repositório remoto)

Obs: posteriormente cria uma pasta/diretório e inicia o git

$ git init

$ git status (verificar o status atual do repositorio)

$ git add Readme.md (adiciona arquivo para ser "rastreado")

$ git commit -m "comentario" (envia arquivo ao repositorio -m é de mensagem, esta identicará o que foi feito)

$ git commit -am "comentario" (tendo sido add anteriormente, este adiciona e faz commit de itens que estão em unmodified e modified)

$ git log (apresenta informações hash, autor, data, msg)

$ git log --decorate

$ git log --author="Paulo" (inf commit do autor)

$ git shortlog (resumo de commits ordem alfabetica autor)

$ git log --graph (formato grafos, quando há outras branch)

Obs: com a hash identificamos o que foi feito no commit.

$ git show + hash (mostra o que foi feito APÓS commit)

$ git diff (apresenta mudanças ANTES de fazer commit)

$ git diff --name-only (diz somente nome do arquivo modificado)

$ git checkout + nome.extensao (Volta o arquivo para antes da edição, checkout é antes de add e commit)

$ git reset HEAD nome.extensão (Arquivo foi add porém esta no staged, ele voltará para modified podendo usar o git checkout)

$ git reset ALTERA historico do commit, no HARD evite depois que estiver no repositorio remoto sendo compartilhado

$ git reset --soft + hash anterior (mata o commit e volta para staged)

$ git reset --mixed + hash anterior (mata o commit e volta para ANTES staged ou seja modified )

$ git reset --hard + hash anterior (ignora a existencia do commit, mata tudo que foi feito neste commit)

AGORA INTEGRANDO GIT & GITHUB

$ git push origin main (origin é onde estou enviando, no caso repositorio remoto é o origin. Main é branch onde estou)

$ git clone + chave SSH (permite clonar um repositório existente no repositório remoto)

$ FORK (faz cópia de projeto que não é seu, para o seu repositório remoto, fork é feito em projetos de outros, clone é nos meus)

$ BRANCH (ponteiro movel que leva o commit, poder modificar sem alterar o local principal, evita conflitos)

$ git checkout -b nomebranch (cria um novo branch e já entra nele)

$ git branch (mostra os branch existente)

$ git checkout -D nomebranch (remove a branch)

$ git merge nomebranch (unir branch criada com a master/main )

$ git rebase nomebranch (unir branch rebasetest com master/main)