

configuração inicial do usuario -identidade
\$ git config --global user.name "Fulano de Tal"
\$ git config --global user.email fulanodetal@exemplo.br

alterar o editor de texto padrão

\$ git config --global core.editor emacs

teste chave espfica

git config <key>:

\$ git config user.name
Fulano de Tal

lista configurações definidas no teu git
git config --list

iniciando repositorio

\$ cd /home/user/your_repository

depois git init

\$ git add *.c
\$ git add LICENSE
\$ git commit -m 'initial project version'
repositorio com commit inicial

clonar um repositorio existente
git clone comando
\$ git clone https://github.com/libgit2/libgit2

verificando status do arquivos
\$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean

adicionar rastreamento a um arquivo no git

\$ git add README Readme nome arquivo que quero rastreie assim o verei com comando git status

comando para renomar arquivos no git

\$ git mv arq_origem arq_destino

captura a versao do meu projeto pro git

\$ git commit nome arquivo

historico de commits

git log

exibição de um repositório remoto

git remote add nome repositório

criar uma branch no git

git branch <nome da branch>

listando a branch git branch -v

O comando git pull é usado para buscar e baixar conteúdo de repositórios remotos

git pull [<options>] [<repository> [<refspec>...]]

O comando git push para atualizar repositório remotos

git push [--all | --mirror | --tags] [--follow-tags] [--atomic] [-n | --dry-run] [--receive-pack=<git-receive-pack>]
[--repo=<repository>] [-f | --force] [-d | --delete] [--prune] [-v | --verbose]
[-u | --set-upstream] [-o <string> | --push-option=<string>]
[--[no-]signed|--signed=(true|false|if-asked)]
[--force-with-lease[=<refname>[:<expect>]]] [--force-if-includes]]
[--no-verify] [<repository> [<refspec>...]]

com seus argumentos dependendo do seu objetivo atual.

git-request-pull - Gera um resumo das alterações que ainda estão pendentes

git request-pull [-p] <início> <url> [<fim>]

git-merge - Junte dois ou mais históricos de desenvolvimento juntos

git merge [-n] [--stat] [--no-commit] [--squash] [--[no-]edit]
[--no-verify] [-s <strategy>] [-X <strategy-option>] [-S[<keyid>]]
[--[no-]allow-unrelated-histories]
[--[no-]rerere-autoupdate] [-m <msg>] [-F <file>] [<commit>...]

git merge (--continue | --abort | --quit)

com seus argumentos dependendo do seu objetivo atual.