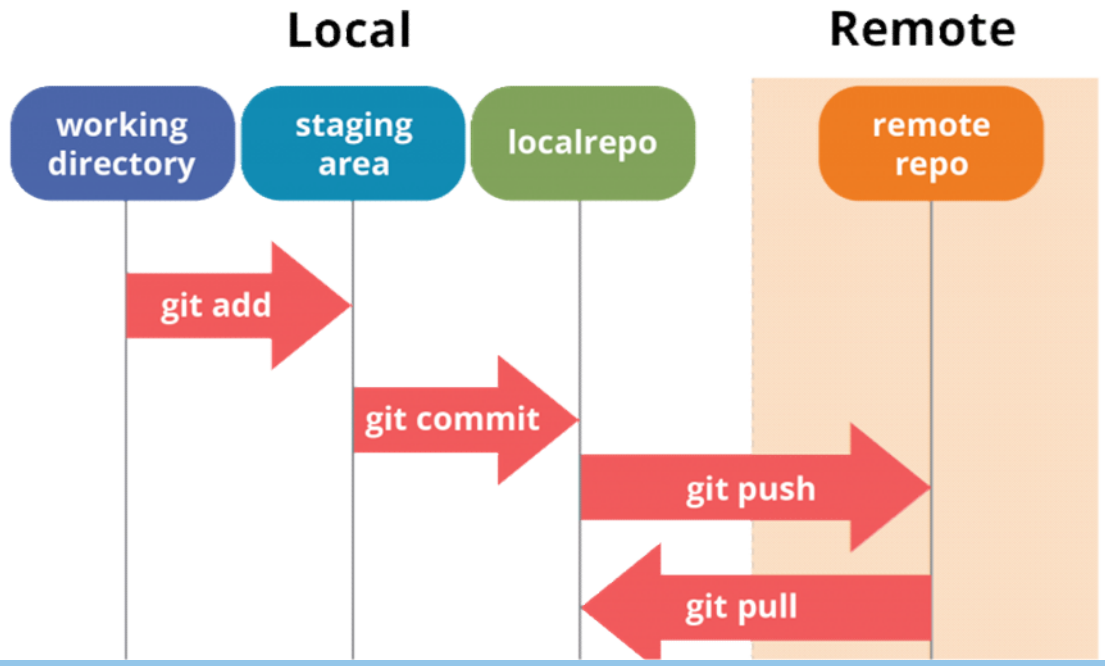
git cheat sheet

Formador > Ezequiel Llarena Borges


Local
Remote
working directory
staging area
localrepo
remote repo
git add
git commit
git push
git pull

WORK FLOW

crear nuevo repositorio	`git init`
configurar	`git config [--global] user.name <nombre_usuario>`
	`git config [--global] user.email <email_usuario>`
mostrar configuración	`git config -l | --list`

Create

ver **estado** del repo local	`git status`
clonar repositorio **local**	`git clone /path/to/repository`
clonar repositorio **remoto**	`git clone http[s]://host[:port]/path/to/repo.git/`
clonar rama repositorio **remoto**	`git clone -b | --branch <branch_name> <repoURL>`
clonar repo con **SSH**	`git clone ssh://[user@]host[:port]/path/to/repo.git/`
clonar repo con **Git SSH** protocol	`git clone git@hostname:username/reponame.git`

Create & clone

añadir cambios a Staging	`git add <filename>`
añadir todos los cambios a Staging	`git add .`
eliminar archivos del Staging. –cached solo elimina del index/staging. Los ficheros siguen en el directorio.	`git rm --cached <filename>`

ADD & REMOVE

registrar cambios add + commit	`git commit -m "Commit message/description"` `git commit -am "mensaje"`
vincular repo local con repo remoto	`git remote add origin <url-repo>`
subir cambios al repositorio **remoto**	`git push origin master`
desvincular repo local de repo remoto	`git remote remove origin`
actualizar repo local con los cambios repo remoto	`git pull`

COMMIT & SYNCHRONIZE

crear nueva rama	`git checkout -b <branch>`
cambiar a rama **master**	`git checkout master`
eliminar rama	`git branch -d <branch>`
subir rama al repositorio **remoto**	`git push origin <branch>`

BRANCHES

fusionar cambios desde otra rama	`git merge <branch>`
revisar cambios entre dos ramas	`git diff <source_branch> <target_branch>`

MERGE

crear etiqueta	`git tag <tag> <commit ID>`
obtener IDs de los commits	`git log`

TAGGING

shows the **committed history**	`git log [--online]` `git reflog`
undo and create new commits which contain an inverse of the specified commits changes*	`git revert <hash-commit-to-undo>`
undo changes**	`git reset –soft|--mixed|--hard <hash-commit-to-undo>`

UNDOING CHANGES

reemplazar cambios locales	`git checkout -- <filename>`
deshacer todos los cambios locales y commits	`git fetch origin` `git reset --hard origin/master`

RESTORE

ENLACES Y RECURSOS

Clientes gráficos

- SourceTree
- GitHub for Mac
- GitBox
- Comparativa clientes gráficos git

Guías

- Documentación Oficial Git
- Git, la guía sencilla
- Tutorial “Learn Git” de Bitbucket
- Git Community Book
- Think like a git
- GitHub Help
- A Visual Git Guide