

# Introducción a git



# ¿Que es git?

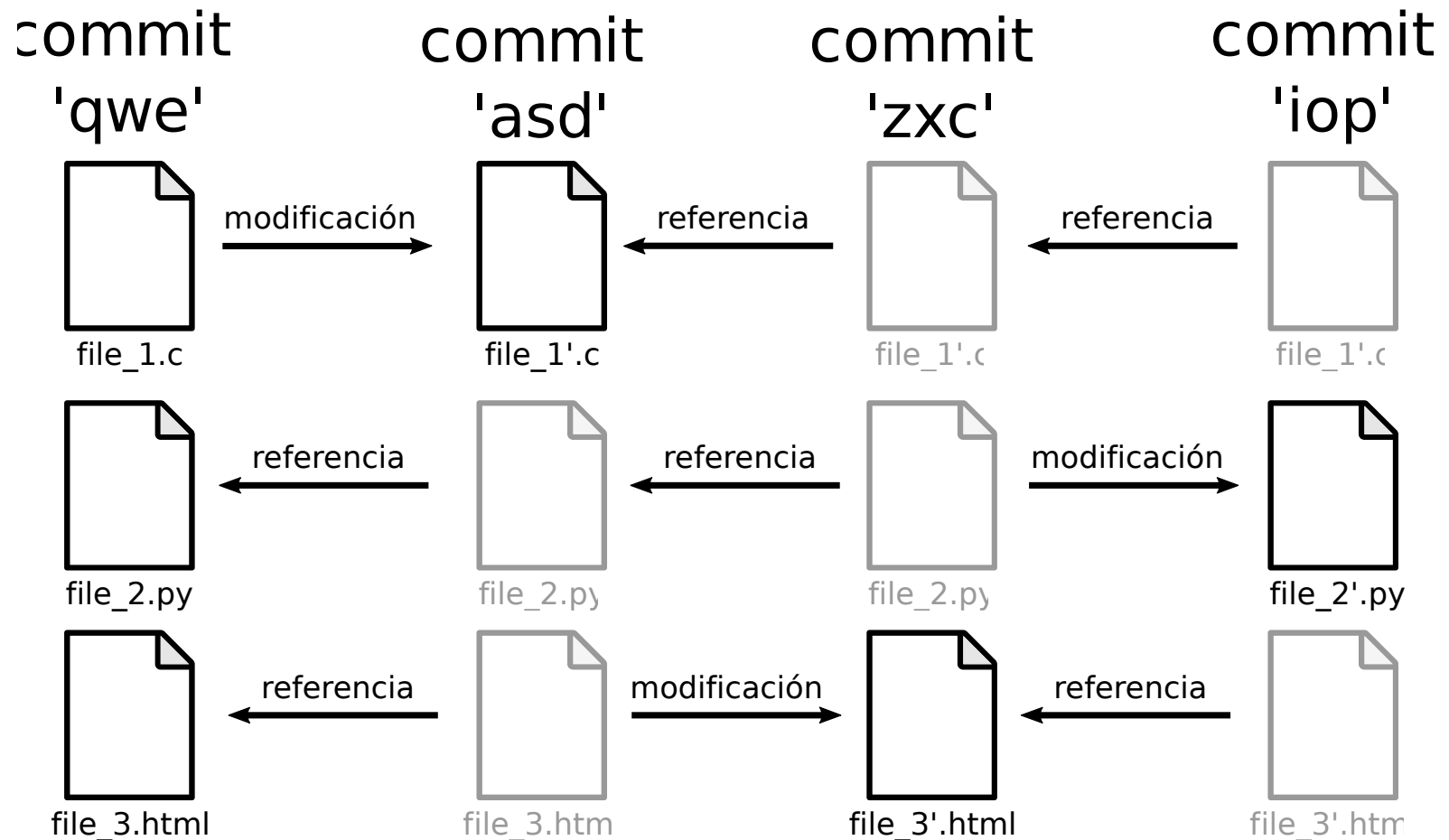
THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.



# Cada versión es un commit

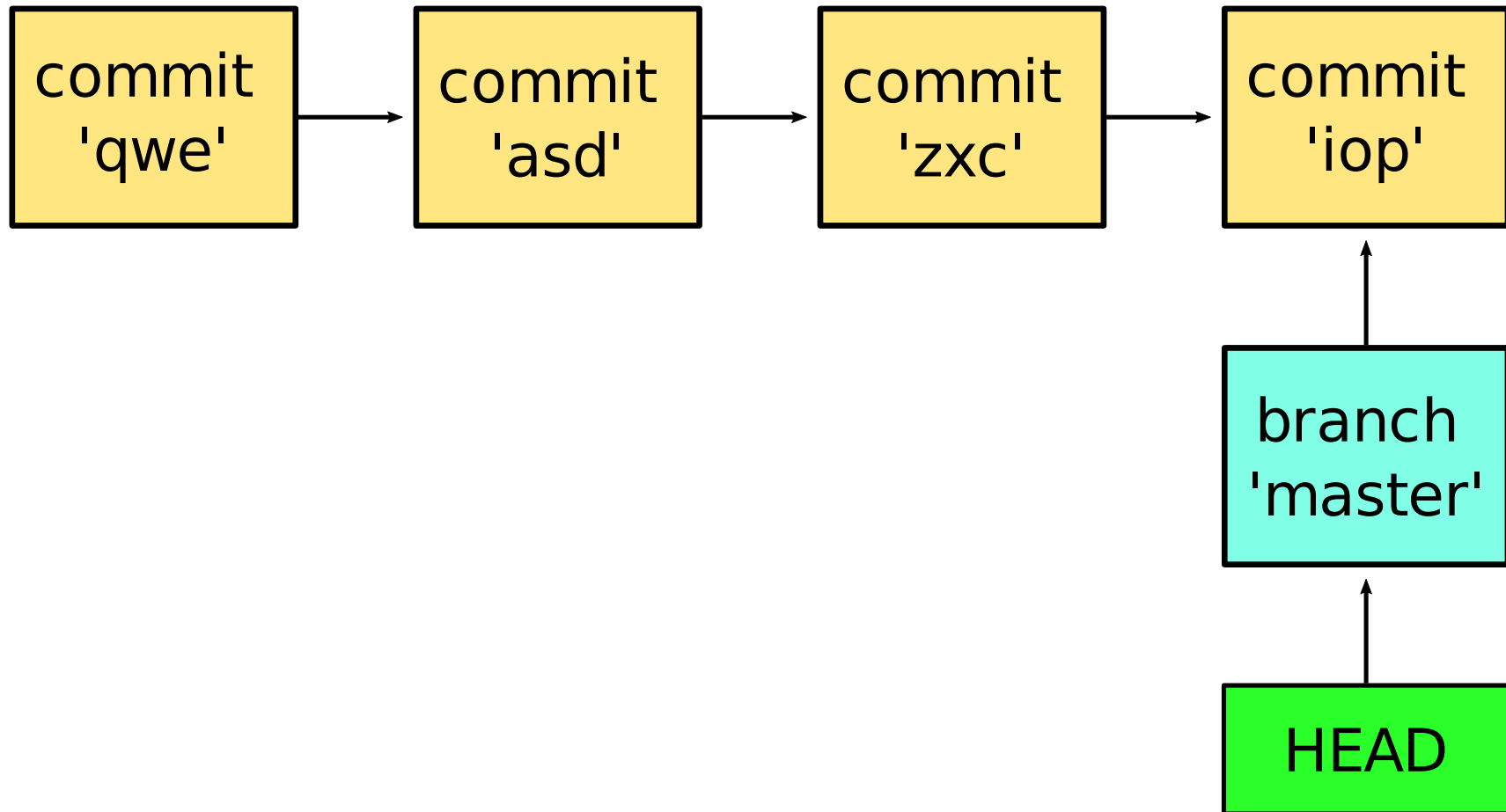


# ¿Que es un commit?

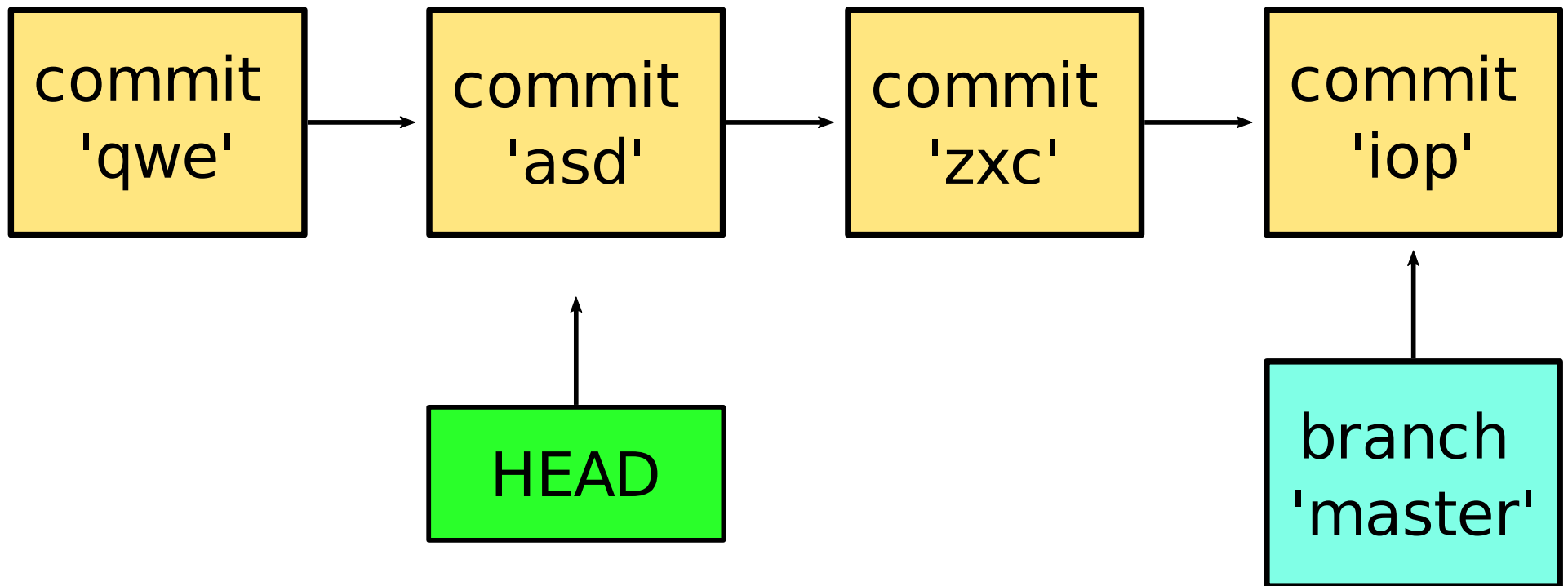
Técnicamente, es un objeto de git

- `id`: identificador único
- `commit anterior`
- `autor`
- `fecha`
- `mensaje`
- `blob` con el contenido del proyecto

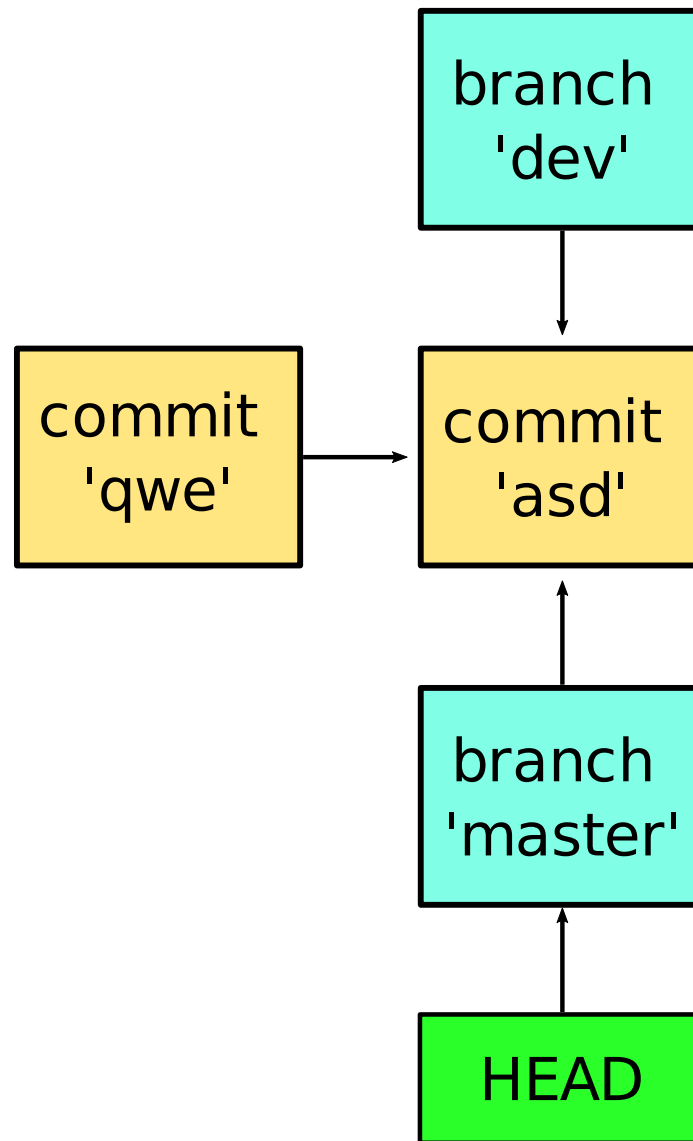
# Punteros y concepto de branches



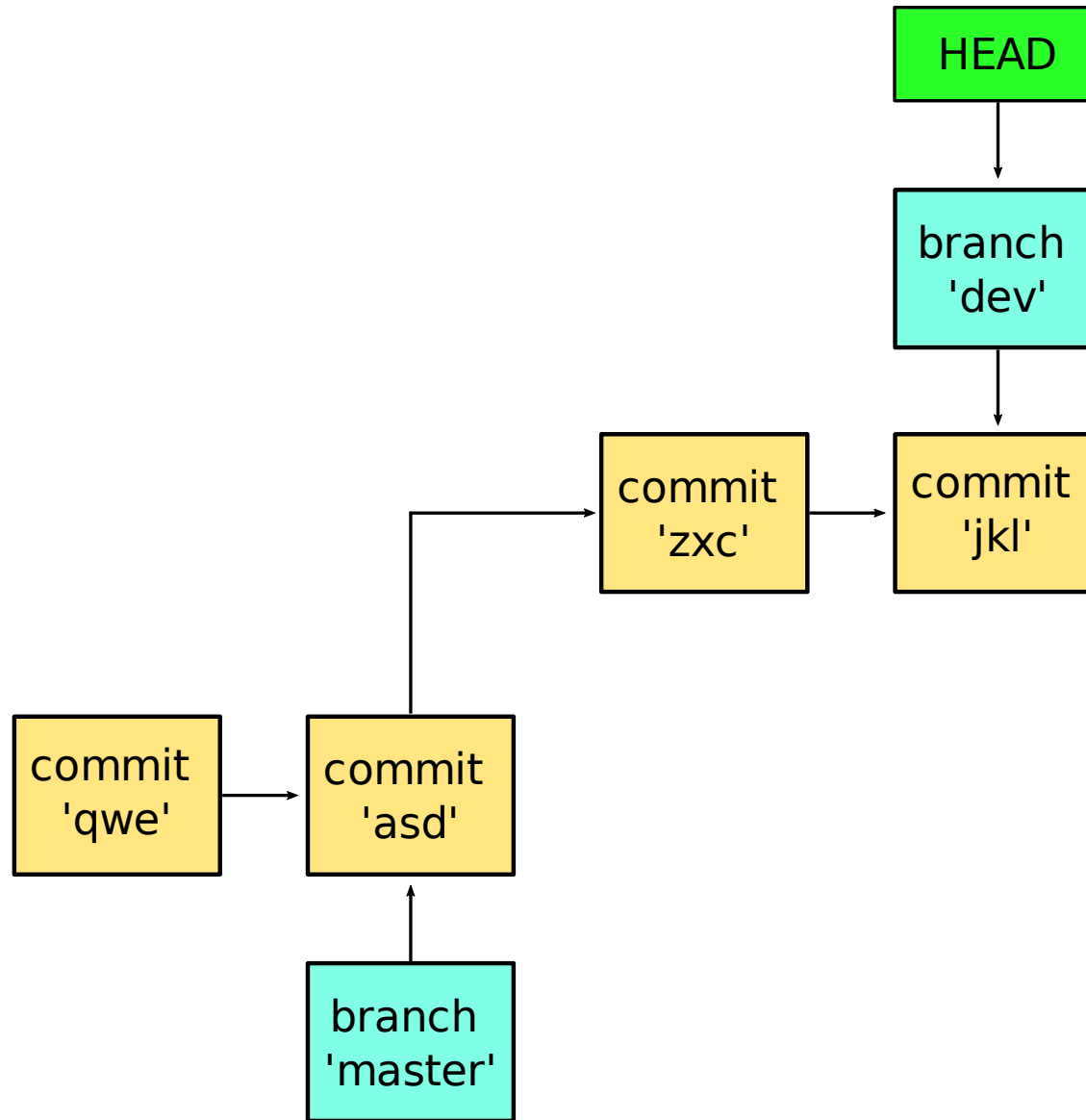
# Modo detached



# Creación de branches

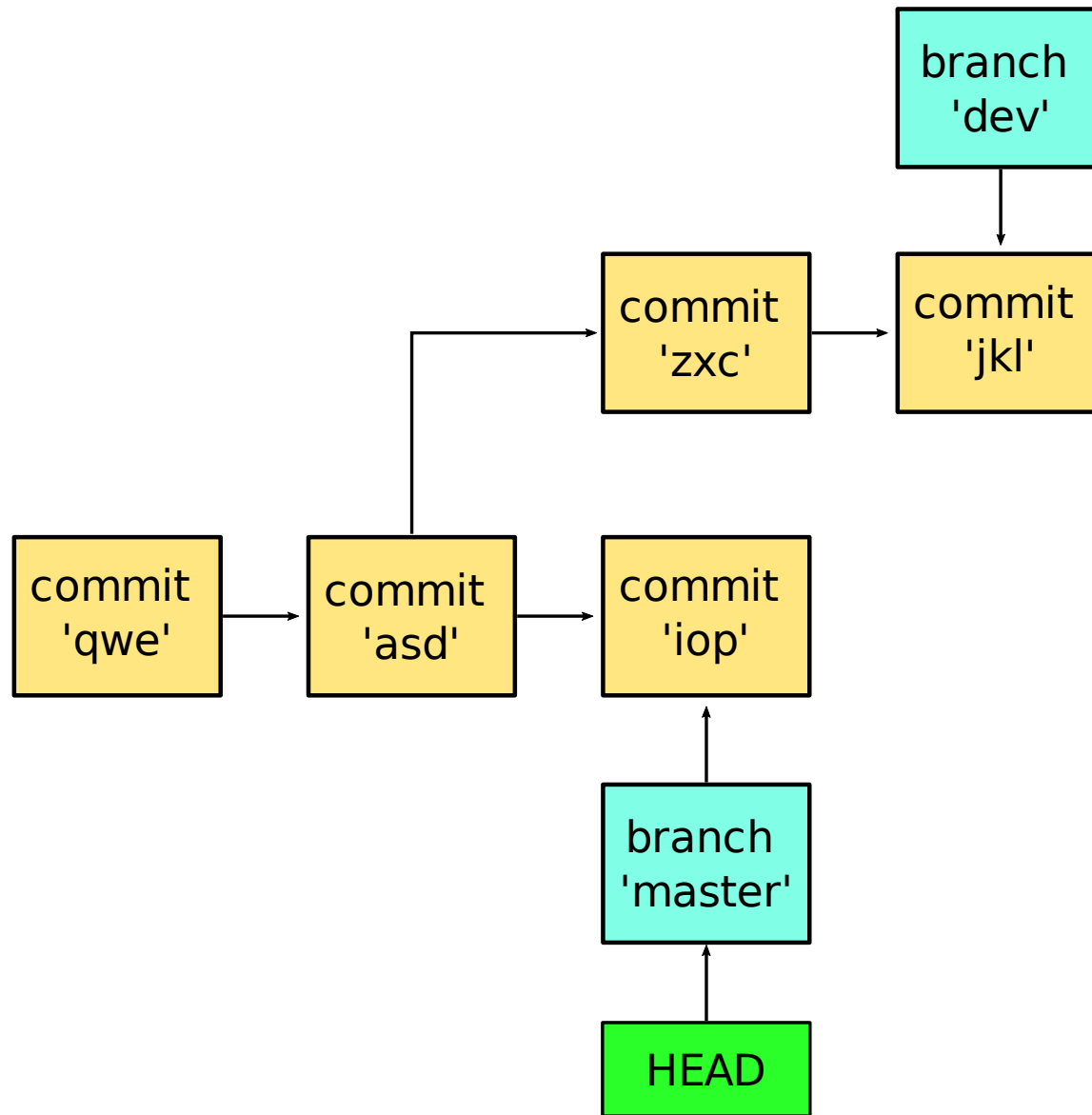


# Trabajando en nuevos branches

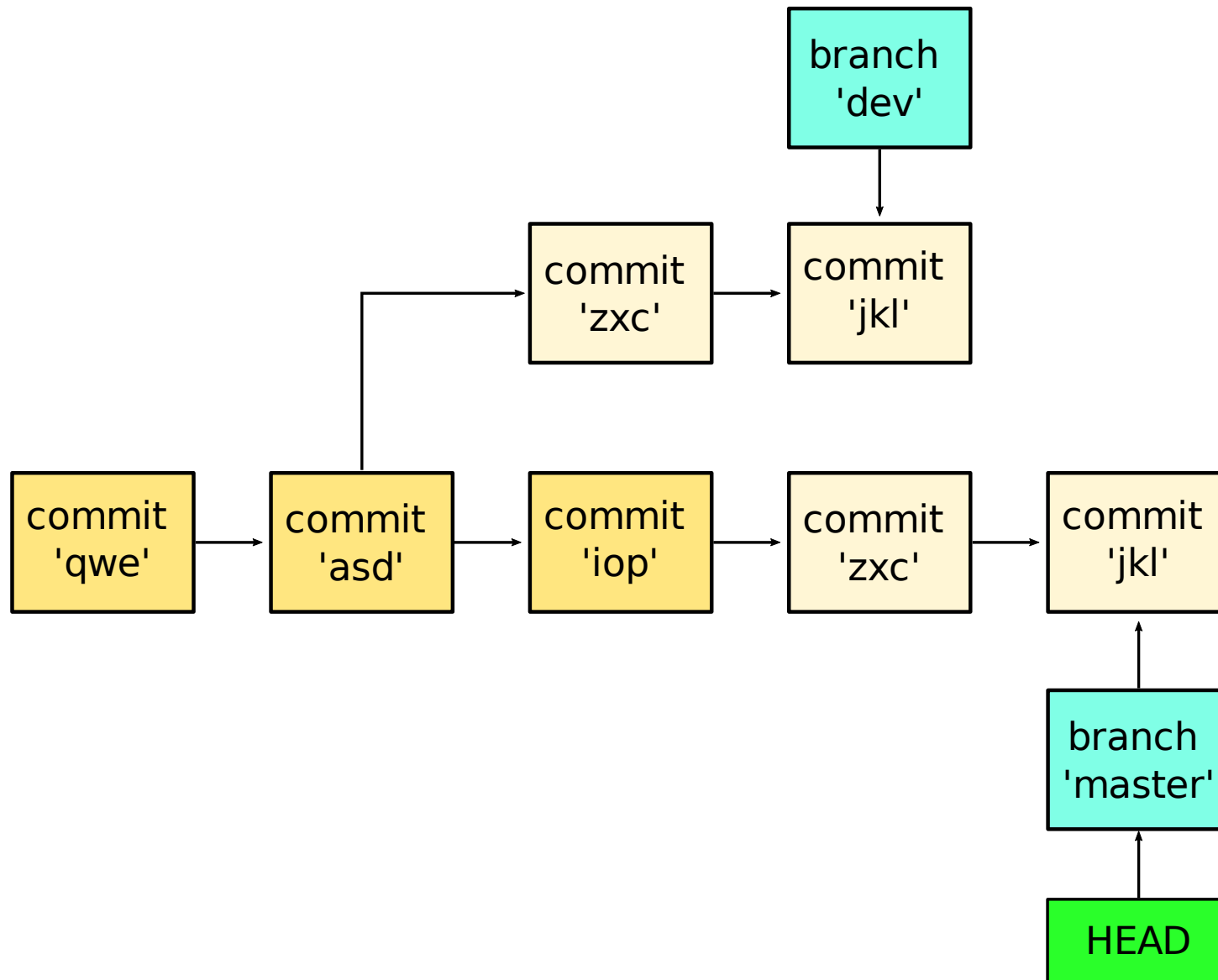




# Desarrollo divergente



# Operación merge



# Algunos comandos para operaciones locales

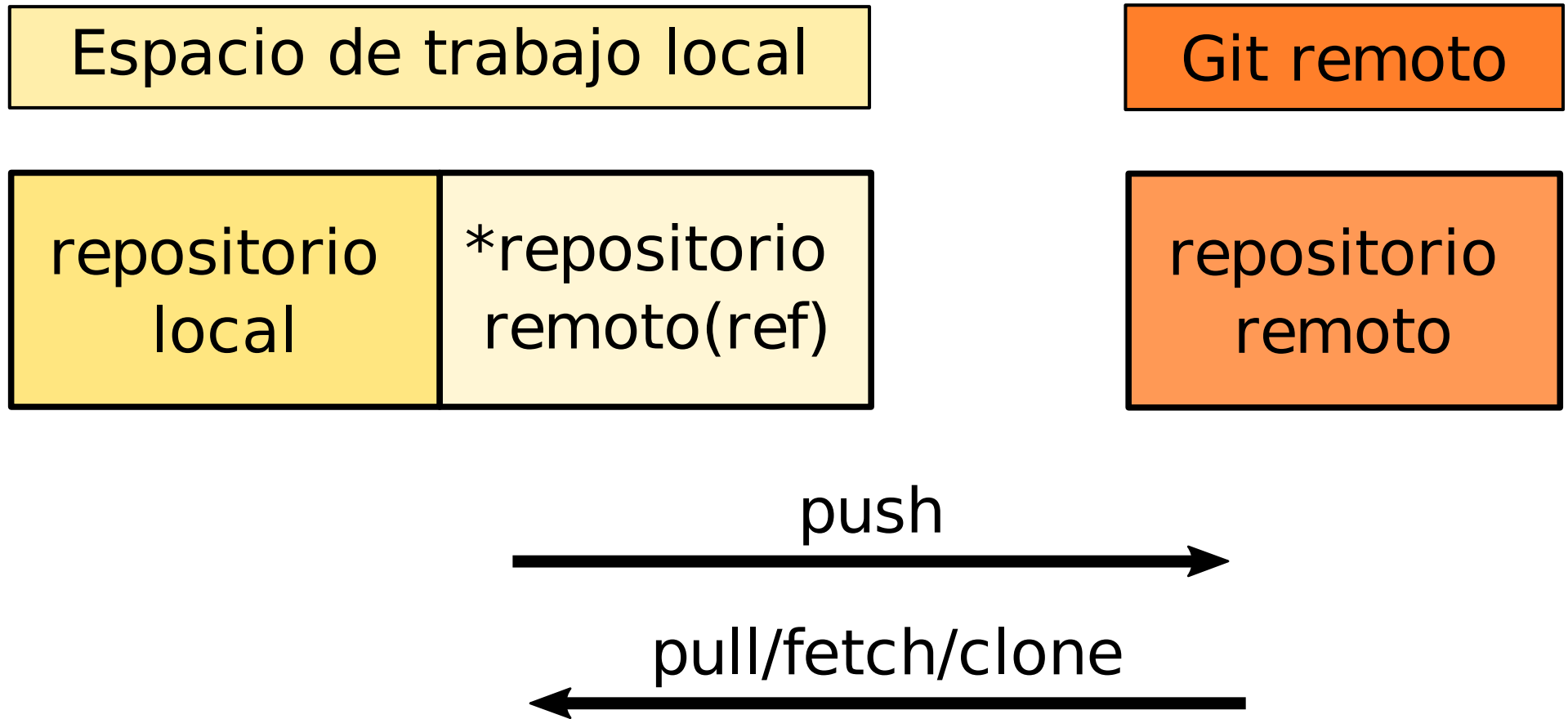
- `$git init`: inicializa un repositorio
- `$git status`
- `$git add`: agrega archivos al área de staging
- `$git commit -m "msg"`: crea el commit
- `$git log`: historial de commits
- `$git checkout`: mueve el puntero HEAD
- `$git merge`: trae commits de otro branch

# Operaciones remotas

- Local: repositorio propio de un usuario
- Remoto: repositorio que no es local
- Origin: repositorio central

***GitLab, GitHub, etc. no son git***

# Operaciones remotas



# Algunos comandos para operaciones remotas

- **\$git clone:** clona un repo completo
- **\$git pull:** "trae" cambios
- **\$git push:** "sube" cambios
- **\$git remote:** lista los remotos asociados
- **\$git remote add origin https://url.de.repo.remoto:** asocia un remoto (origin) con el repo local

# Otros recursos

- Nicolás Paez - Taller de git
- CS50W - Lecture 1 - Git
- freeCodeCamp - Manual esencial de git
- [git-scm.com](https://git-scm.com) - User manual

git add \*  
git commit  
git push

GitHubSetup.exe

git clone

git init

GitHub Desktop.app



← Estamos aquí

git pull origin master

git config

git add foo

Merge made by the 'recursive' strategy.

git remote add

'.gitignore'

git status



git reset --hard HEAD

git branch

git checkout

git merge

rm -rf bar; git clone https://github.com/foo/bar

'<<<<<<<< HEAD'

git fetch

git log

git rm



git diff

git commit -am

git checkout -- foo

git fetch upstream

git merge upstream/master

'====='

git remote -v

git stash



git push --force

git push origin :foo

git submodule

git push origin HEAD:refs/for/master

git cherry-pick

git grep

git blame

'>>>>>> bar'

git rebase -i

git tag



git subtree

git rev-parse --show-toplevel

'.gitattributes'

cp pre-commit.sh .git/hooks/pre-commit

'.git/info/exclude'

git merge - 'branch.master.mergeoptions = --no-ff'

git branch --merged | xargs git branch -d

git bisect



git reset -p HEAD^

git rebase

git rerere

git daemon --reuseaddr --verbose --base-path= ./git

git update-index --assume-unchanged

git worktree

git fsck

git filter-branch



'git gets easier once you get the basic idea that branches are homeomorphic endofunctors mapping submanifolds of a Hilbert space'

