

Xuleta de Git

Les comandes del dia a dia, en una pàgina

El cicle mínim

<code>git status</code>	On estic i què tinc sense consolidar
<code>git diff</code>	Què he tocat i encara no he consolidat
<code>git add fitxer.sql</code>	Preparar eixe fitxer per al commit
<code>git commit -m "Motiu (INC-1234)"</code>	Consolidar-ho, amb el seu motiu
<code>git push</code>	Pujar-ho al repositori compartit

`git status` abans i després de cada passa. Al principi és el que orienta.

Veure què ha canviat

<code>git diff</code>	El que he tocat jo i encara no he consolidat
<code>git diff main...la-meua-branca</code>	Tot el que aporta una branca respecte de <code>main</code>
<code>git log --oneline -- fitxer.sql</code>	Totes les versions d'un fitxer: data, autor i motiu
<code>git log --oneline --all -- fitxer.sql</code>	Igual, però sense amagar cap branca
<code>git show al83fe2</code>	Un canvi concret, amb el seu motiu i el seu autor
<code>git blame fitxer.sql</code>	Quin commit va deixar cada línia com està

`blame` no és per a buscar culpables: és per a arribar al commit i llegir el motiu.

Branques

<code>git switch -c nom-de-la-branca</code>	Crear-la i passar-hi
<code>git switch main</code>	Tornar a la principal
<code>git branch --show-current</code>	En quina estic
<code>git push -u origin nom-de-la-branca</code>	Publicar-la la primera vegada

Recuperar

<code>git show a183fe2:fitxer.sql</code>	Mirar com estava, sense tocar res
<code>git restore --source=a183fe2 fitxer.sql</code>	Tornar-lo enrere en el meu directori
<code>git restore fitxer.sql</code>	Descartar el que he tocat i no he consolidat
<code>git revert a183fe2</code>	Desfer un canvi ja integrat, amb un commit nou

Copieu el hash de `git log`; no compteu posicions amb `HEAD~2`, que depèn de quants commits hàgeu fet pel camí.

En una branca compartida, sempre `git revert`.

Un conflicte

Git s'atura quan els dos heu tocat la mateixa línia i deixa els dos candidats dins del fitxer:

```
<<<<<< HEAD
la vostra versió
=====
la versió de l'altre
>>>>>> main
```

Obriu-lo, deixeu el que ha de quedar, lleveu **les tres línies de marcadors**, i:

```
git add fitxer.sql
git commit
```

Un conflicte no és una avaria: és Git negant-se a triar per vosaltres en l'únic cas en què no pot fer-ho sense inventar.

El que no s'ha de fer

	Per què
<code>git push --force</code> en una branca compartida	Descarta el que hi ha al servidor i trenca el clon de tothom
Reescriure l'historial de <code>main</code>	Per a desfar-hi coses, <code>git revert</code>
Posar una contrasenya en un commit	L'historial la guarda per sempre: cal rotar la credencial
Un commit que ho barreja tot	Impossible de revisar i de revertir per separat

En la vostra branca, encara sense fusionar, sí que podeu reescriure: `git rebase -i` ajunta els commits de prova en un que explique el canvi de veres. I si heu de forçar, `git push --force-with-lease`, que falla si algú havia pujat res.

Quan el commit no diu per què

```
Canvia timeout a 60                                     ← el diff ja ho diu  
Evita timeout durant la càrrega de matrícula (INC-1234) ← això no està enlloc més
```