GUÍA PASO A PASO

Git para trabajar con agentes de IA

El flujo de git para pedirle cambios a un agente de IA sin miedo: punto seguro, rama del agente, revisión, commits pequeños y cómo volver atrás. Cada comando lo corrimos en un repositorio de prueba, y al final está la chuleta.

PARA	Claude Code, Cursor o cualquier agente que edite tu repositorio
PROBADO CON	git 2.54 · Node 22 · Windows 11 (Git Bash y PowerShell)
ACTUALIZADA	30 de octubre de 2026
DOCUMENTACIÓN	git-scm.com/docs
ARTÍCULO	ecomlabs.dev/g/git
AUTOR	Luis Morales · @luismoralesdev

Índice

Paso 01	Deja un punto seguro
Paso 02	Crea la rama del agente
Paso 03	Pide el cambio
Paso 04	Revisa antes de guardar
Paso 05	Guarda en commits pequeños
Paso 06	Si salió mal, vuelve atrás
Paso 07	Integra o tira la rama
Chuleta	Qué comando para cada caso
Al terminar	Checklist

PASO 01

Deja un punto seguro

Las salidas de esta guía son de un repositorio de prueba con git 2.54: una tienda donde el agente agrega un descuento al carrito. Para practicar sin tocar tu proyecto, crea uno con `git init -b main`. Antes de pedir nada, mira si tienes trabajo sin commit:

```
$ git status --short
 M README.md
```

Una línea por archivo: `M` es modificado sin pasar al stage, `??` es un archivo nuevo que git no sigue y, si no sale nada, no hay nada pendiente. Guarda lo tuyo:

```
$ git add -A
$ git commit -m "Documenta cómo correr las pruebas"
[main 7994890] Documenta cómo correr las pruebas
 1 file changed, 2 insertions(+)
```

Anota el hash (`7994890`): es el punto al que puedes volver.

PASO 02

Crea la rama del agente

```
$ git switch -c agente/descuento-volumen
Switched to a new branch 'agente/descuento-volumen'
```

`-c` crea la rama y se cambia a ella en un paso. El prefijo `agente/` es una convención: así sabes qué ramas puedes tirar sin pensarlo dos veces.

PASO 03

Pide el cambio

Dile al agente qué esperas al final:

```
Agrega 10% de descuento desde 3 piezas en totalCarrito, con sus pruebas.
No hagas commit ni push: al terminar, dime qué archivos tocaste y corre npm test.
```

Si aun así hace commits, estás en su rama: el paso 06 explica cómo deshacerlos.

PASO 04

Revisa antes de guardar

```
$ git status --short
 M src/carrito.js
?? test/descuento.test.js
$ git diff --stat
 src/carrito.js | 5 ++++-
 1 file changed, 4 insertions(+), 1 deletion(-)
```

`git diff` no cuenta el test nuevo: compara tus archivos con el stage, y un archivo que git no sigue no está ahí. Lee el diff completo con `git diff` y después, para ver también lo nuevo:

```
$ git add -A
$ git diff --staged --stat
 src/carrito.js         |  5 ++++-
 test/descuento.test.js | 15 +++++++++++++++
 2 files changed, 19 insertions(+), 1 deletion(-)
```

Corre las pruebas (`npm test`; en la demo, 4 de 4). En el diff, busca archivos que no esperabas, depuración olvidada (en la demo, un `console.log`) y pruebas viejas que cambiaron para poder pasar.

PASO 05

Guarda en commits pequeños

Deja fuera lo que no pediste. Saca el archivo del stage y elige bloque por bloque:

```
$ git restore --staged src/carrito.js
$ git add -p src/carrito.js
(1/2) Stage this hunk [y,n,q,a,d,k,K,j,J,g,/,e,p,P,?]?
(2/2) Stage this hunk [y,n,q,a,d,K,J,g,/,s,e,p,P,?]?
$ git commit -m "Agrega 10% de descuento desde 3 piezas"
[agente/descuento-volumen 90b20a1] Agrega 10% de descuento desde 3 piezas
 2 files changed, 18 insertions(+), 1 deletion(-)
 create mode 100644 test/descuento.test.js
$ git restore src/carrito.js
```

En `git add -p`: `y` agrega el bloque, `n` lo deja fuera, `q` sale y `s` divide un bloque cuando se puede. En la demo respondimos `n` al `console.log` e `y` al descuento. El último `git restore` descarta lo que quedó fuera del commit.

Mensajes: una primera línea de no más de 50 caracteres que diga qué cambia. `git log --oneline` se vuelve tu lista de puntos seguros.

PASO 06

Si salió mal, vuelve atrás

SITUACIÓN	COMANDO
Un archivo quedó mal	`git restore <archivo>`
Pasaste algo al stage por error	`git restore --staged <archivo>`
Todo lo que no tiene commit sobra	`git reset --hard`
Quedaron archivos o carpetas nuevos	`git clean -nd`, luego `git clean -fd`
El commit del agente sobra	`git reset --hard <hash del punto seguro>`

> **CUIDADO**
> `git reset --hard` borra todos los cambios sin commit de los archivos que git sigue, y lo que nunca tuvo commit no se recupera ni con `git reflog`. Úsalo solo en la rama de trabajo del agente, nunca en `main` ni en una rama compartida, y siempre después de mirar `git status`.

Ojo: `git reset --hard` no borra archivos nuevos. En la demo, `src/tmp-depuracion.js` siguió ahí hasta `git clean -f`. Usa siempre `-n` primero: muestra qué borraría sin borrar.

Si te arrepientes de un reset a un commit anterior, los commits siguen en el reflog (por defecto 90 días; 30 si ya no están en ninguna rama):

```
$ git reset --hard 7994890
HEAD is now at 7994890 Documenta cómo correr las pruebas
$ git reflog -3
7994890 HEAD@{0}: reset: moving to 7994890
90b20a1 HEAD@{1}: reset: moving to HEAD
90b20a1 HEAD@{2}: commit: Agrega 10% de descuento desde 3 piezas
$ git reset --hard 'HEAD@{1}'
HEAD is now at 90b20a1 Agrega 10% de descuento desde 3 piezas
```

Las comillas en `'HEAD@{1}'` son para PowerShell, donde sin ellas el comando falla; en bash y zsh dan igual.

PASO 07

Integra o tira la rama

```
$ git switch main
Switched to branch 'main'
$ git merge --ff-only agente/descuento-volumen
Updating 7994890..90b20a1
Fast-forward
$ git branch -d agente/descuento-volumen
Deleted branch agente/descuento-volumen (was 90b20a1).
```

`--ff-only` integra solo si `main` no avanzó por otro lado; si avanzó, se niega y tú decides con calma cómo integrar. Para tirar un intento que no sirve, `git branch -D agente/<tarea>`: `-d` se niega con ramas sin integrar.

Si el cambio ya está en `main` o lo subiste, no uses reset: deshazlo con un commit nuevo.

```
$ git revert --no-edit HEAD
[main 0768f21] Revert "Agrega 10% de descuento desde 3 piezas"
```

CHULETA

Qué comando para cada caso

QUIERES…	COMANDO
Ver qué cambió	`git status --short` · `git diff`
Punto seguro	`git add -A` · `git commit -m "..."`
Rama para el agente	`git switch -c agente/<tarea>`
Guardar solo una parte	`git add -p <archivo>`
Descartar un archivo	`git restore <archivo>`
Sacar del stage	`git restore --staged <archivo>`
Descartar todo sin commit (cuidado)	`git reset --hard`
Borrar archivos nuevos	`git clean -n` · `git clean -f` (con carpetas: `-nd`, `-fd`)
Volver a un commit (cuidado)	`git reset --hard <commit>`
Recuperar tras un reset	`git reflog` · `git reset --hard 'HEAD@{1}'`
Integrar la rama	`git switch main` · `git merge --ff-only agente/<tarea>`
Borrar la rama	`git branch -d` (integrada) · `git branch -D` (descartada)
Deshacer algo compartido	`git revert --no-edit <commit>`

AL TERMINAR

Checklist

- [ ] `git status` limpio, o tu trabajo en un commit, antes de pedir; el agente, en su rama.
- [ ] Revisaste `git status --short`, los `??` y el diff completo.
- [ ] Las pruebas pasan y ninguna prueba vieja cambió sin motivo.
- [ ] `git reset --hard` y `git clean -f`, solo en la rama del agente.
- [ ] Integraste con `--ff-only` y borraste la rama.