



Comandos útiles de Git

FIDE.DEV

007

¿Algún desarrollador agregó un bug? **Encuentra al culpable con:**

```
git blame archivo
```

Esto nos dirá quién hizo qué en ese archivo; o puedes hacerlo más específico por línea:

```
git blame archivo -L11,14
```

008

Deja de estar escribiendo comandos largos, ponles un apodo con:

```
git config --global alias.grapho "git log --all --graph --decorate --oneline"
```

Ejecutalo con:

```
git grapho
```

003

¿Sabes cómo restaurar un conjunto de archivos?

Conjunto de archivos con la misma extensión:

```
git restore '*.css'
```

El directorio actual:

```
git restore .
```

004

¿Después de hacer el commit te das cuenta de que te faltaron cosas? Ejecuta:

```
git clean -f
```

Esto agregará los cambios sin dejar registros en la historia de tu repositorio.

FIDE.DEV

005

¿Quieres limpiar el directorio de todo aquello que nos trackeado? **Ejecuta:**

```
git clean --dry-run
```

Para ver cuáles son los archivos que eliminará; ahora eliminalos con:

```
git clean -f
```

006

¿Quieres buscar cierta variable o función que usaste con anterioridad? **Usa:**

```
git log -S miFuncionPerdida
```

Te permite rastraer cualquier cosa que alguna vez hayas comiteado.

FIDE.DEV

001

¿No te diste cuenta de que estabas haciendo cambios en la rama master? **Solo usa:**

```
git checkout -b nueva-rama
```

Todos los cambios que no hayas agregados a git add se pasarán a la nueva rama

002

¿Quieres cambiar de rama, pero no estás listo para hacer commit en la branch actual? **ejecuta:**

```
git stash
```

Almacena el avance en stash y cambia de rama sin ningún problema.