
Capacitación Git

Módulo 6

Utilidades

por Gonzalo Barro Gil, fuente <https://git-scm.com/book/es/v2>

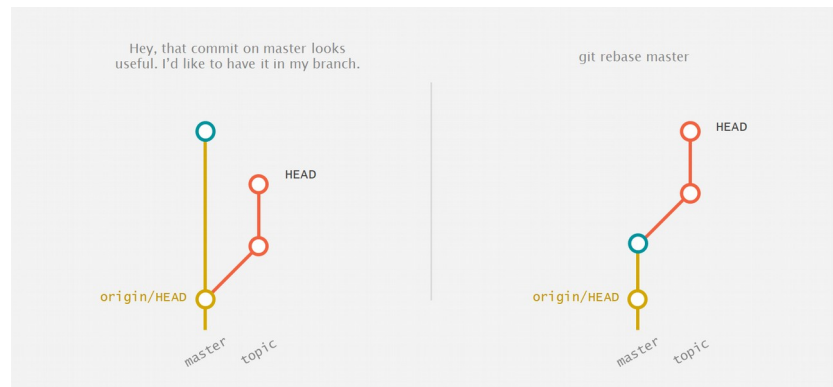
<https://github.com/GonzaloBarroGil/git-course>

Stash

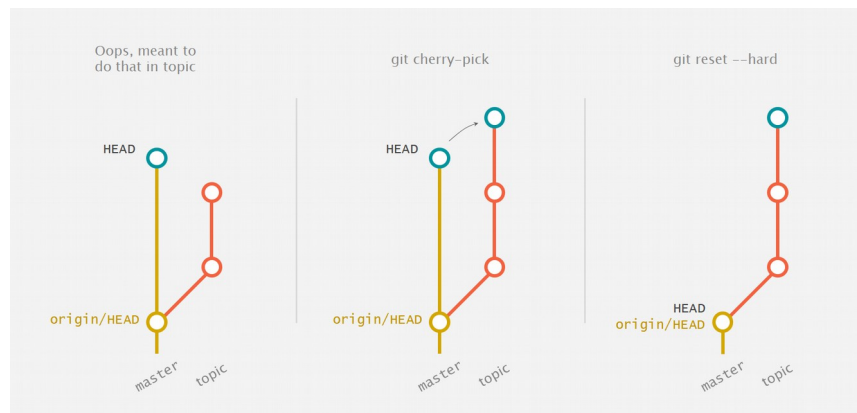
- Cuando queremos salir de un branch es necesario que tanto el directorio de trabajo como el index estén en línea con el HEAD. Si por algún motivo debemos movernos a otro branch y tenemos trabajo pendiente va a ser necesario completar los tres pasos (tareas, git add, git commit).
- Hay algunas situaciones en las que se requiere salir rápido de un branch, por ejemplo resolver un bug crítico. El stash es un área especial de Git que funciona como un "portapapeles" o un área de guardado rápido:
 - `git stash`
Por defecto guarda sólo lo agregado al stage. El branch mantiene los cambios pendientes, si los hay.
 - `git stash -u`
Guarda también los cambios no agregados al stage (untracked). El directorio queda limpio.
- Se puede ejecutar más de un guardado. Cada stash queda almacenado con un identificador.
 - `git stash list`
Lista todos los guardados con su id.
 - `git stash apply`
Recupera el último guardado en cualquier branch. Se puede especificar un stash por su id.
 - `git stash drop`
Elimina todos los guardados efectuados. Se puede eliminar sólo uno especificando su id.
 -
- En lugar de eliminar se puede crear un branch nuevo a partir del stash:
 - `git stash branch nuevoBranch`
Crea un nuevo branch con los cambios pendientes y preparados guardados en el stash y elimina el stash.

Cherry-Pick

- Vimos con el rebasing que se aplicaba un commit a partir de la base de un branch, o sea el ancestro común con el branch padre, y transformar ese commit en la nueva base y aplicar los cambios del mismo a los commits siguientes transformándolos



- En este caso se toma un commit en particular y se aplica sobre el HEAD. En este caso el commit que se transforma es el aplicado; pasa a ser un nuevo commit cuyo padre pasa a ser el commit del HEAD.



Deshacer rebase

- Buscar el id anterior de la cabecera:

```
tail .git/logs/refs/heads/develop
```

```
75fd394... a535260... usuario <email> ... rebase finished: ...
```

- Buscar la base anterior al rebase

```
git log
```

```
a535260 (HEAD -> ...) commit cabecera (modificado en el rebase)  
298db05 nueva base (agregada en el rebase)  
9769642 base (anterior al rebase)
```

- Volver el HEAD a la base anterior:

```
git reset --hard 9769642
```

- Cherry-pick de la cabecera anterior:

```
git cherry-pick 75fd394
```

```
git log
```

```
05c7718 (HEAD -> ...) commit cabecera (modificado en el cherry-pick)  
9769642 base (anterior al rebase)
```