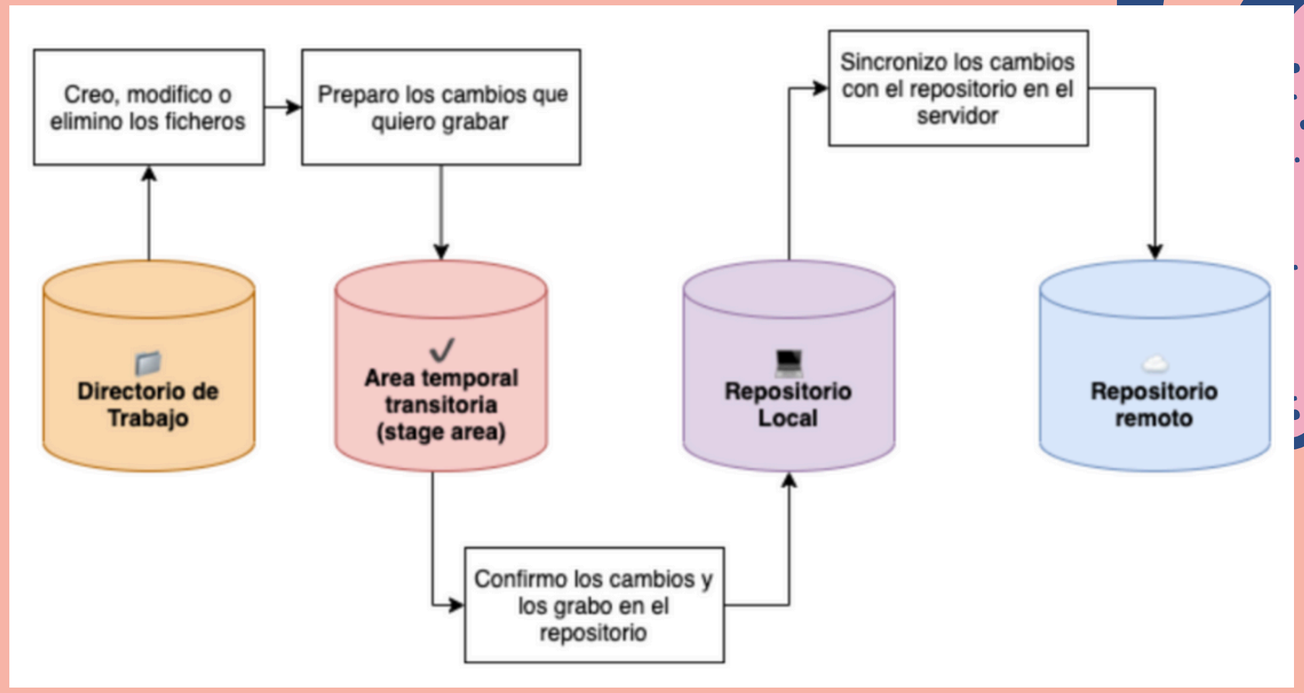
States y commits

Los 3 estados de Git



Modified

El archivo ha sido creado, eliminado o contiene cambios que no han sido marcados como confirmados.

Staged

El archivo ha sido marcado como preparado para ser confirmado en el repositorio local.

Commited

El archivo se encuentra grabado en el repositorio local. Esta acción recibe el nombre de *commit*.

¿Qué es un commit?

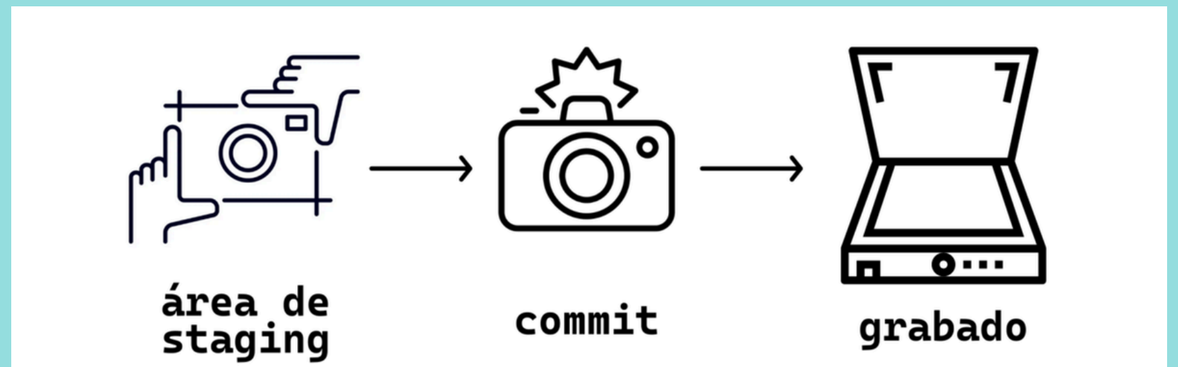
Los commits sirven para registrar los cambios que se han producido en el repositorio.

Es una de las piezas más importantes para entender cómo funciona Git.

¿Qué es un commit?

Piensa en los commits como si fuesen fotografías.

Cada fotografía muestra el estado de todos los archivos de tu repositorio en el momento en que se hizo y cada una va firmada con el autor, la fecha, localización y otra información útil.



El área de staging sería el encuadre, el commit la fotografía y, para guardarla, el repositorio escanearía la foto para replicarla

¿Qué es un commit?

Es como guardar partida en un videojuego, tienes tu punto de restauración si te equivocas.

¿Cómo hago un Commit?

El mensaje en un commit es muy pero que muy importante.

Si quieres **guardar los cambios que tienes en el área de staging**, puedes hacer un commit con el siguiente comando:

```
$ git commit
```

Esto creará un nuevo commit en tu repositorio y añadirá una referencia al commit en la rama en la que estás trabajando. Pero antes de hacerlo, tienes que indicar el mensaje del commit. Para ello te abrirá el editor que hayas configurado previamente.

Si quieres añadir directamente un mensaje sin abrir el editor, puedes usar el parámetro `-m` o `--message`:

```
$ git commit -m "Add new search feature"
```

El mensaje especificado se usará como título del commit para describir los cambios realizados. Si quieres añadir más información a tu commit, más parecido a lo que podías hacer al abrir el editor del mensaje de commit sin parámetro, puedes volver a utilizar el parámetro `-m` tantas veces como quieras.

```
$ git commit -m "Add new search feature" -m "This new feature is awesom\
e. Make the search to work faster and better!"
```

> En el capítulo de *Buenas prácticas*, verás diferentes reglas que puedes seguir para escribir los mejores mensajes de commit posible.

Ahora, es importante que entiendas que **estos cambios se van a grabar en tu repositorio local**. A partir de aquí, para deshacer estos cambios, tendrás que revertirlos grabando un nuevo commit en el historial de cambios del repositorio.

¿Qué es el HEAD?

Entiende HEAD como un “estás aquí” de un mapa. Sólo puedes estar en un sólo lugar y ese lugar es el HEAD (en mayúsculas).

HEAD es el puntero que referencia el punto actual del historial de cambios del repositorio en el que estás trabajando.