

UT 2.4 GIT. Clonado. Gestión de confirmaciones y ramificaciones (I)

ENTORNOS DE DESARROLLO

1. Navegar a un commit determinado

- A medida que el desarrollo avanza, nos puede interesar retroceder en el tiempo y situarnos en un commit que ya no es el último.

```
$ git log
commit 5792f9a2f4db150188b58d0b37352883bd969dcf (HEAD -> master, origin/master)
Author: Roberto Hidalgo <robertohidalgo.informatica@iespaconolla.es>
Date: Thu Oct 24 21:56:07 2019 +0200

    Incluidos estilos de la página principal junto a su referencia desde index.html

commit 5f5bd92e2bcc0cc40852b8d0488ed7b560d43d3f
Author: Roberto Hidalgo <robertohidalgo.informatica@iespaconolla.es>
Date: Thu Oct 24 21:55:23 2019 +0200

    Incluida imagen a la web junto a su referencia desde index.html

commit 48f95bbf668168891856912d527d786e840805b8
Author: Roberto Hidalgo <robertohidalgo.informatica@iespaconolla.es>
Date: Thu Oct 24 21:54:15 2019 +0200

    Añado index.html sin estilos ni imagenes asociadas
```



id de commit 2

1. Navegar a un commit determinado

```
$ git checkout 5f5bd92e2bcc0cc40852b8d0488ed7b560d43d3f
```

id de commit 2

Note: switching to '5f5bd92e2bcc0cc40852b8d0488ed7b560d43d3f'.

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by switching back to a branch.

.....

1. Navegar a un commit determinado

- Para volver al último commit realizado basta con ejecutar:

```
$ git checkout master
```

```
Note: switching to '5f5bd92e2bcc0cc40852b8d0488ed7b560d43d3f'.
```

```
Previous HEAD position was 5f5bd92 Incluida imagen a la web junto a su  
referencia desde index.html
```

```
Switched to branch 'master'
```

```
Your branch is up to date with 'origin/master'.
```

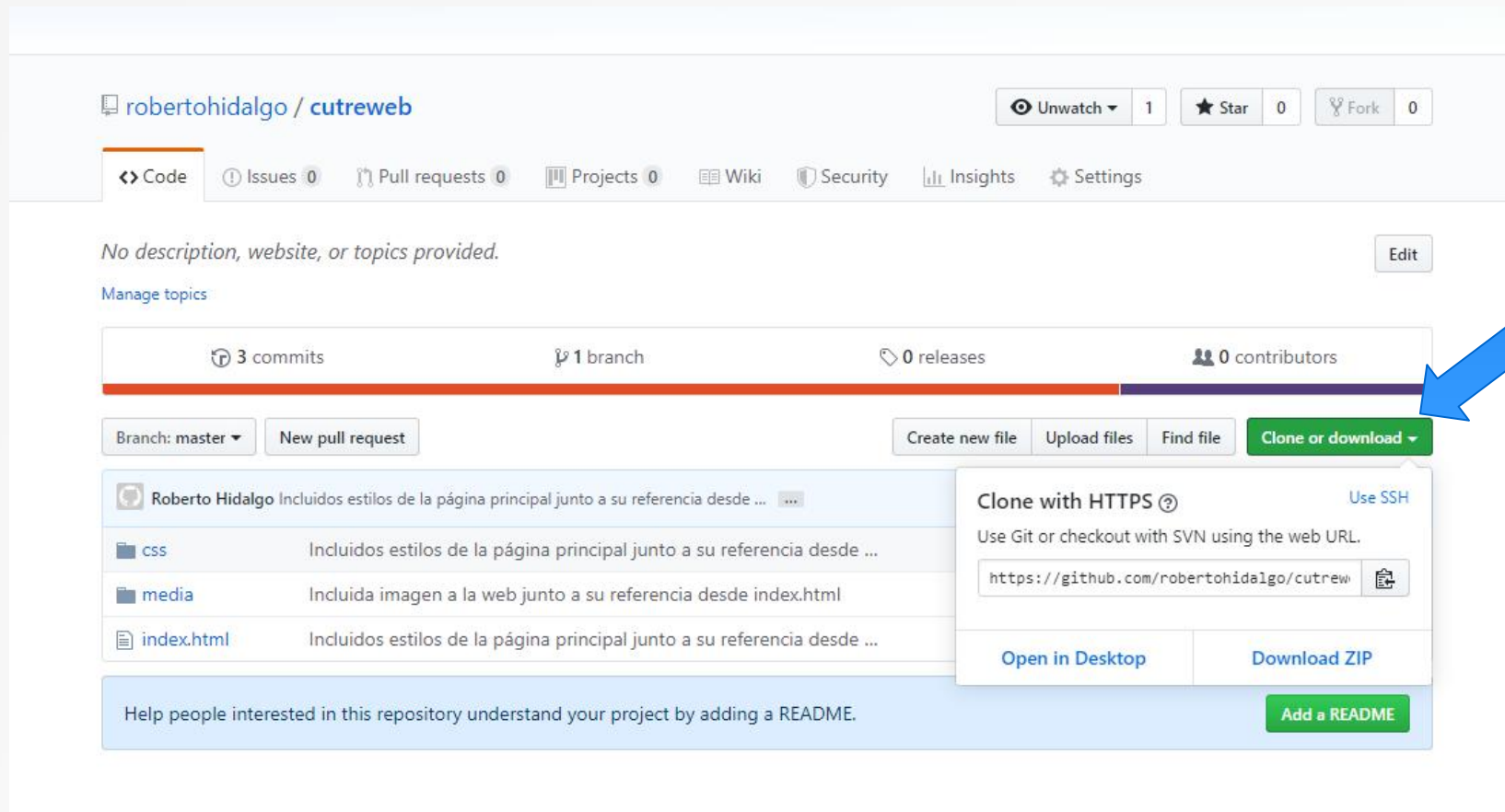
2. Clonar un repositorio

Como usuarios de git, podemos clonar (hacer una copia) de los ficheros que se encuentren en un remoto bien para colaborar en un proyecto concreto, bien para descargar ese proyecto en nuestro ordenador. En nuestro ejemplo, Jane quiere clonar el repositorio de Gregg.



2. Clonar un repositorio


- Para clonar un remoto, necesitas su URL. La podemos encontrar en GitHub



2. Clonar un repositorio


De esta manera clonaremos el repositorio remoto en nuestro repositorio local

```
$ git clone URL_repositorio
```



Crea una carpeta con el mismo nombre del repositorio remoto con todos los archivos dentro de ella

```
$ git clone URL_repositorio nombre_carpeta_local
```



Crea una carpeta con el nombre de la carpeta local que se haya dado en el comando con todos los archivos dentro de ella

2. Clonar un repositorio

- ¿Qué hace entonces clone?
 - Descarga el repositorio completo en la carpeta local
 - Añade el remoto 'origin', haciendo que apunte a esa URL
 - recuerda que se puede saber los remotos asociados ejecutando
`$ git remote -v`
 - Establece la rama inicial (normalmente master) con todos los commits que ésta posea y estableciendo HEAD al commit más reciente.

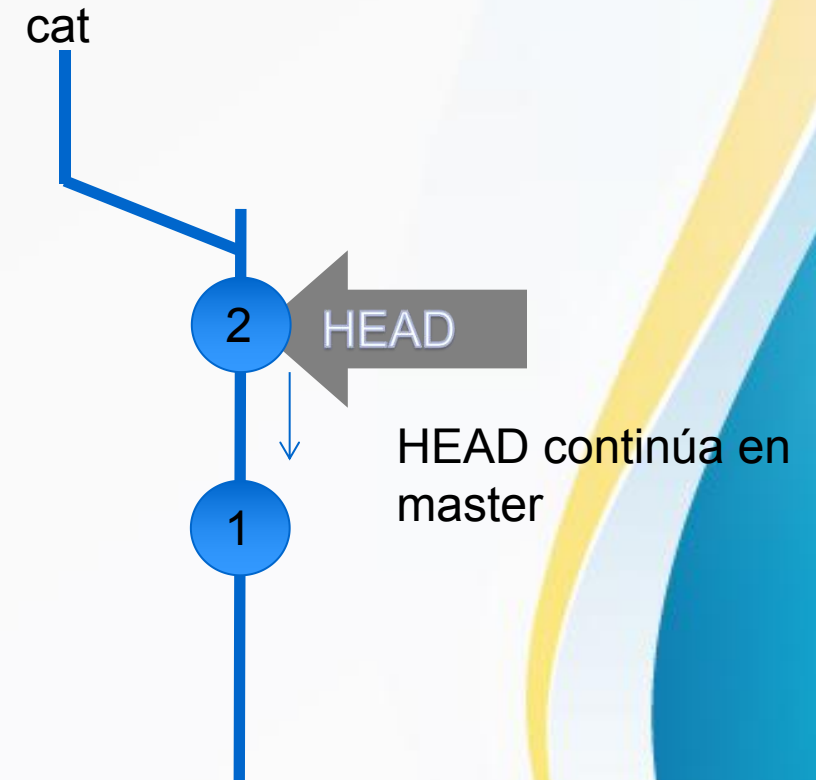


3. Ramificar

- Jane, ya tiene su repositorio creado y quiere implementar una nueva funcionalidad que le va a llevar algo de tiempo.
- Es hora de crear una rama.
- En el ejemplo, se crea una rama llamada cat

```
$ git branch cat
```

Nombre de la nueva rama



3. Ramificar

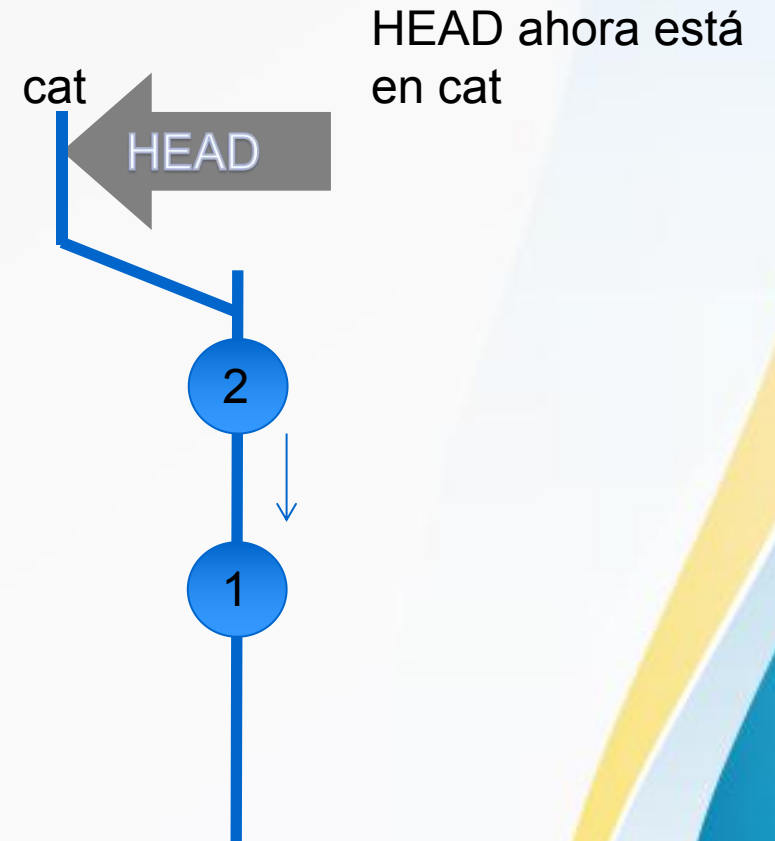
- Ejecutando `git branch`, nos da la información de las ramas disponibles y cuál es la seleccionada actualmente (sobre la que se sitúa HEAD)

```
$ git branch  
  cat  
* master
```

En este caso nos indica que la rama seleccionada es **master**

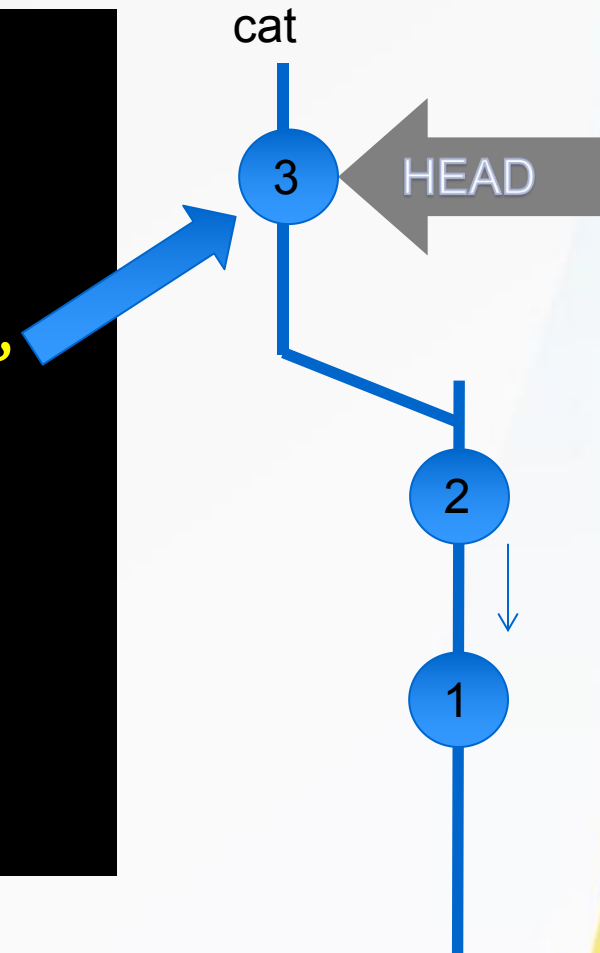
3. Ramificar. Cambiar de rama

```
$ git checkout cat  
Switched to branch 'cat'
```




3. Ramificar. Commit sobre nueva rama

```
$ echo "Schrödinger" > cat.txt  
  
$ git add cat.txt  
  
$ git commit -m "Añadido un gato cuántico"  
[cat ab48a3f] Create quantum cat.  
1 file changed, 1 insertion(+)  
create mode 100644 cat.txt
```



3. Ramificar.


- Ejecutando `ls` o `dir` se podrá ver que el fichero `cat.txt` se encuentra en nuestro directorio actual.



```
$ ls
README.txt cat.txt
```

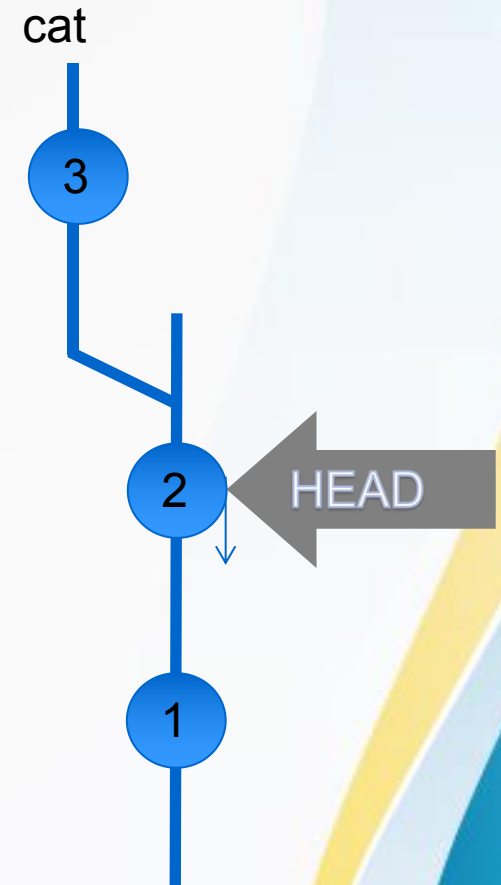
3. Ramificar. Volver a rama master

```
$ git checkout master  
Switched to branch 'master'  
$ ls  
README.txt
```



Si ejecutamos `git log`,
tampoco se lista el último
commit realizado en cat

`cat.txt` ya no
está!



4. Mezclar (merge) rama en master

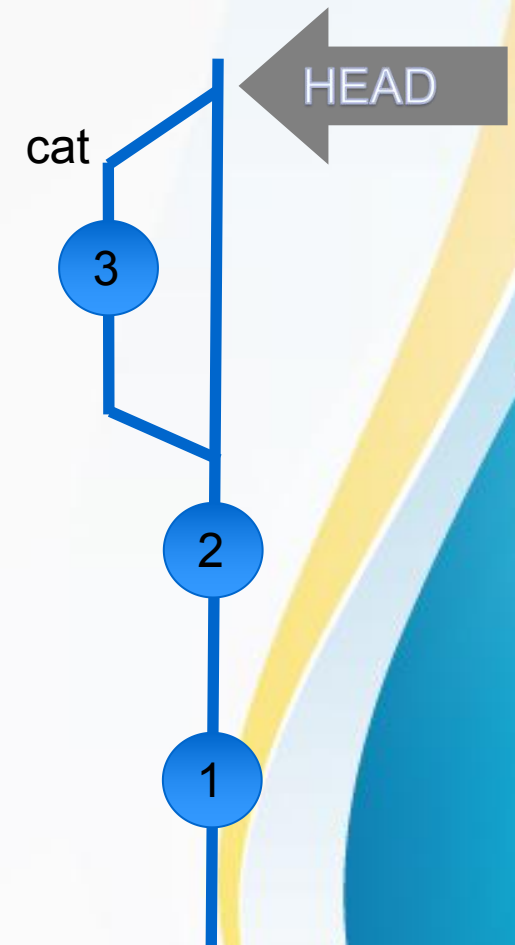
- Jane ya ha finalizado su nueva funcionalidad, ahora toca integrar esta funcionalidad en la rama master.

```
$ git checkout master
Switched to branch 'master'
$ ls
README.txt
```

```
$ git merge cat
Updating 1191ceb..ab48a3f
Fast-forward
 cat.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 cat.txt
```

No hay gato, tal y como esperábamos

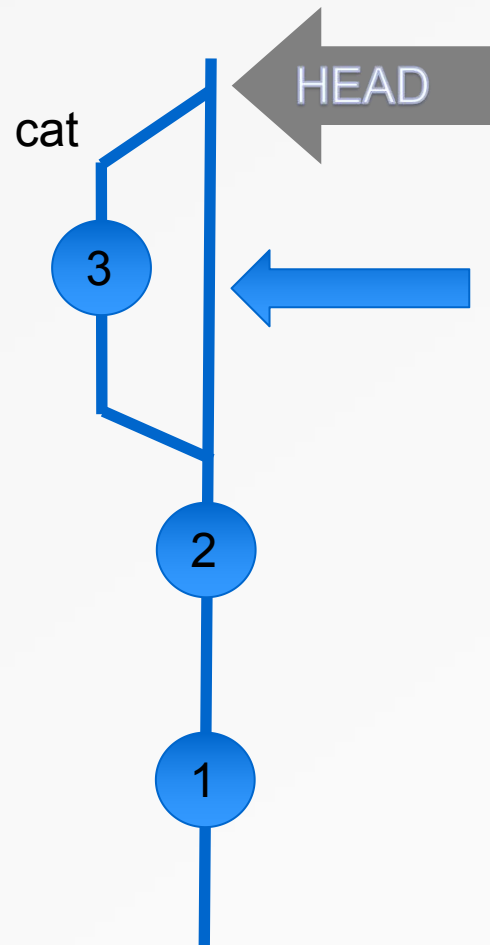
¿Qué significa "Fast - forward?"



4. Mezclar (merge) rama en master.

Fast-forward

- Las condiciones que se deben cumplir para que haya un **Fast-forward** son



No ha pasado nada desde la creación del branch “cat”.

Para GIT es muy fácil integrar la rama cat en master

4. Eliminación de ramas

```
$ git branch -d cat  
Deleted branch cat (was 957dbff).
```

Elimina la rama cat. Hay que estar muy seguro de que la rama ya no va a ser útil.

5. Creación y cambio de rama (juntos)

```
$ git checkout -b mouse
```

nombre de la
nueva rama



**Crea una rama llamada
mouse y situa HEAD en la
nueva rama**