



Gestión de proyectos con Git y GitHub: Modulo 4

Temas:

1. Ramas y grafo de commits: branch, checkout, diff, log, reset y show 2
2. Integración de ramas: merge, commit y checkout y fast-forward 29

Juan Quemada, DIT - UPM

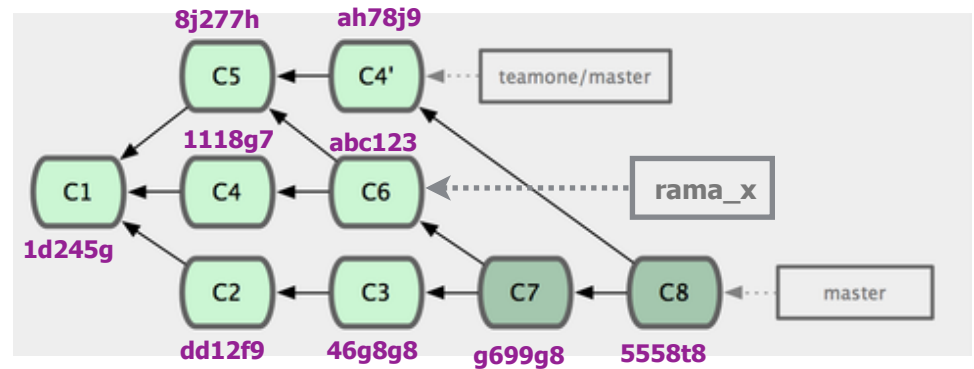


Git y GitHub

Ramas y grafo de commits: branch, checkout, diff, log, reset y show

Juan Quemada, DIT - UPM

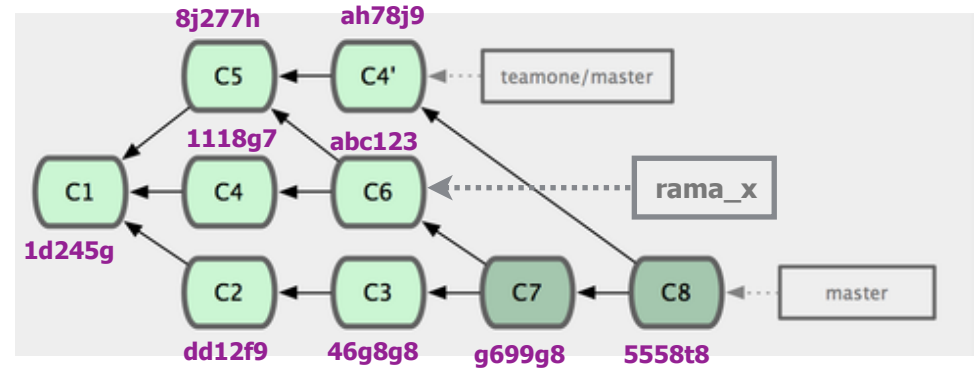
Ramas



*S. Chacon, B. Straub: <https://git-scm.com/book/es/v1>

- ◆ La **rama master** es la rama principal de un repositorio
 - Es una rama predefinida que se crea con el primer commit
- ◆ Las **ramas** soportan desarrollos separados de master
 - Una rama puede comenzar en cualquier commit del repositorio
 - Los nuevos **commits** de una rama **se deben añadir al final** (la rama siempre crece)
 - Una flecha indica el commit o los commits anteriores en el desarrollo
- ◆ **git branch**
 - **git branch rama_y 46g8g8** crea la rama_y con base en C3 (46g8g8)
 - **git branch [-v]** muestra las ramas existentes
- ◆ Una rama es como un puntero a su último commit, por ejemplo
 - **master** asociada al **commit 5558t8**
 - **rama_x** asociada al **commit abc123**

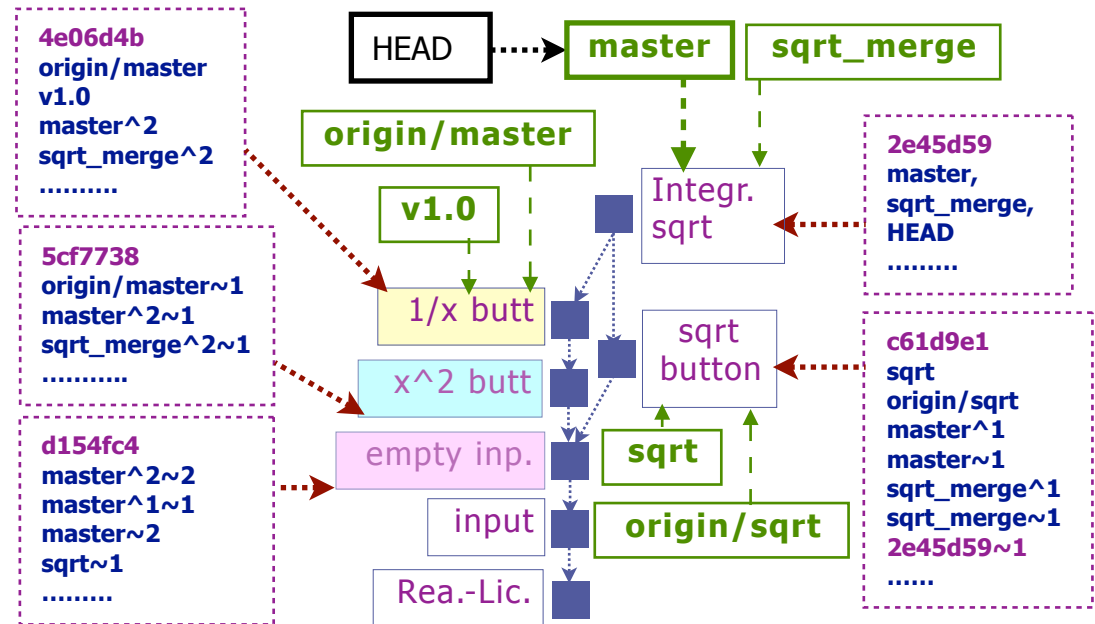
Grafo de commits



*S. Chacon, B. Straub: <https://git-scm.com/book/es/v1>

- ◆ El **grafo de commits** de un repositorio
 - Grafo con la relación de generación de todos los commits de un repositorio
- ◆ Las **flechas de salida** de un commit indican su relación con los anteriores
 - **Una flecha:** commit **generado por modificación** del commit anterior
 - **Dos flechas:** commit **generado por integración** de una rama o commit con otro
- ◆ **Historia de un commit**
 - Secuencia ordenada (por fechas) de commits utilizados para generar dicho commit
- ◆ **Padres y ancestros de un commit**
 - **<commit>^n:** representa el número (n) de **padre** de un commit de integración
 - Por ejemplo: **master^1=g699g8** o **master^2=ah78j9**
 - **<commit>~n:** **ancestro n** de su historia
 - Por ejemplo: **46g8g8~1=dd12f9** o **master~2=46g8g8**

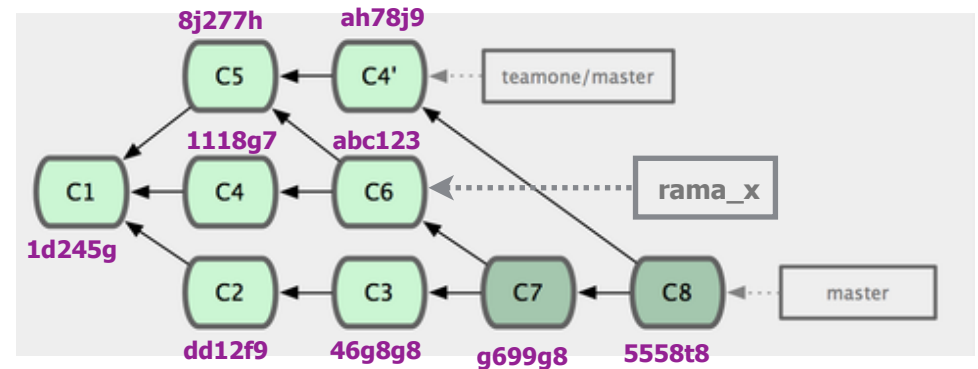
Referencias a commits del grafo del repo.



◆ Las formas mas habituales de identificar los commits del grafo son:

- **HEAD:** puntero o referencia al commit actual en el directorio de trabajo
- **Rama:** puntero al último commit de una rama de desarrollo
 - **Rama local:** rama copiada en el repositorio local (**master**, **sqrt**, ..)
 - **Rama remota:** rama de un repositorio remote (**origin/master**, **origin/sqrt**, **remote/origin/sqrt**, ..)
- **Identificador** corto o largo: identificador único de un commit (**d154fc4**, **973751d2**, ...)
- **Ancestro:** **commit~n** -> sigue la línea del 1er padre, p.e. **master~2**, **sqrt~1**, ..
- **Padre:** **commit^n** -> El primer **padre** (**c^1**) es también el primer ancestro (**c~1**)
- **Tag o etiqueta:** **v1.0**, **v0.1**, -> Se añaden con: **git tag -a v1.0**

Comandos log, diff y show



*S. Chacon, B. Straub: <https://git-scm.com/book/es/v1>

♦ git log ## muestra la historia

- **git log -3** muestra los 3 últimos commits de la historia del commit actual (HEAD)
- **git log --oneline --graph rama_x** muestra el grafo de integración de commits de **rama_x** (formato 1 lín.)
- **git log --oneline master~2** historia del 2o ancestro de **master** (formato 1 línea)
- **git log --oneline --all --graph** muestra el **grafo completo** de commits del repositorio
- **git log --oneline --all -8** muestra los **8 últimos commits de todo el repositorio** ordenados por fechas
- **git log --follow file1 -20** muestra historia de cambios del fichero **file1** (commits donde file1 ha cambiado)

♦ git diff ## muestra diferencias de código entre commits, wd, ...

- **git diff rama_x master** muestra los cambios en ficheros entre **rama_x** y **master**
- **git diff master~2** muestra los cambios en ficheros entre **master~2** y **HEAD**
- **git diff rama_x master -- LICENSE** muestra cambios en el fichero LICENSE entre **rama_x** y **master**
- **git diff rama_x master LICENSE** similar al anterior si no hay ambigüedad en los nombres

♦ git show ## muestra elementos de un commit

- **git show rama_x** muestra metadatos del último commit de rama_x y las diferencias con commit anterior
- **git show** muestra metadatos de HEAD y las diferencias con commit anterior
- **git show HEAD~2:package.json** muestra el código del fichero package.json del commit HEAD~2

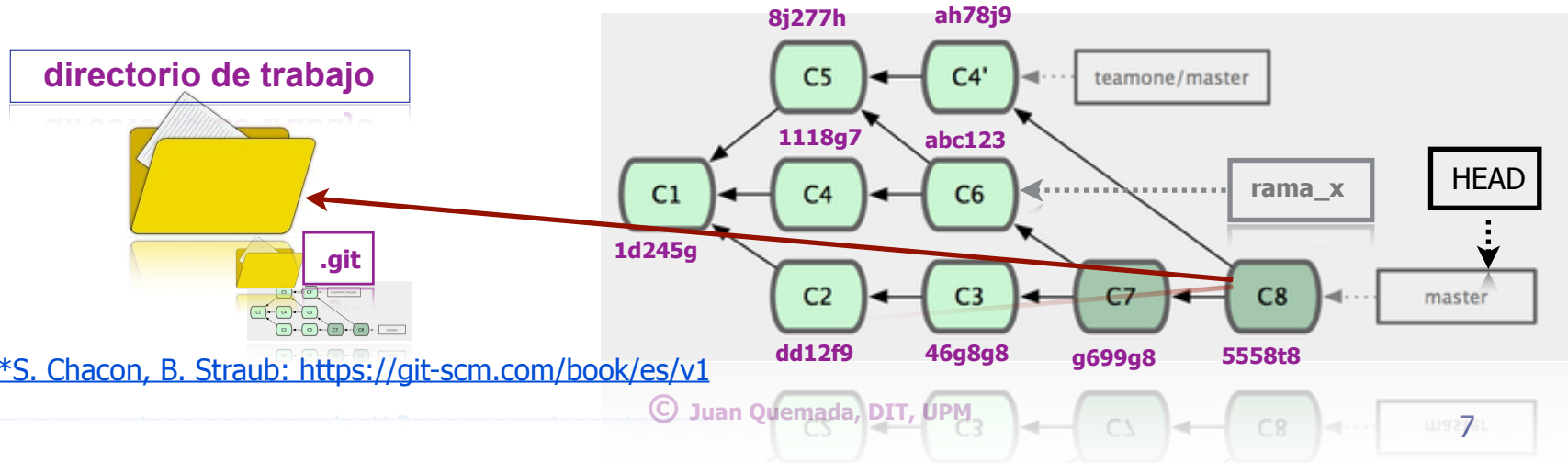
Restaurar commits en el directorio de trabajo

♦ `git checkout <rama>` (o `<commit>`) restaura **<rama>** en el **directorio de trabajo**

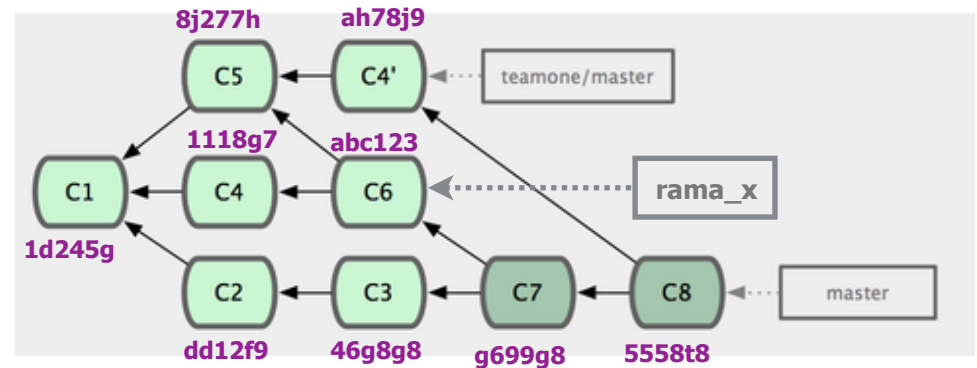
- El **directorio de trabajo** debe estar **limpio** (sin cambios respecto al commit anterior)
 - ¡OJO! La opción `-f` restaura aunque existan cambios en el directorio de trabajo, que **se perderán**
- Ejemplos
 - `git checkout rama_x` restaura `rama_x` (abc123) en el directorio de trabajo y actualiza HEAD
 - `git checkout dd12f9~2` restaura el segundo ancestro de 5558t8 en el dir. de trabajo y actualiza HEAD
 - `git checkout -b rama_y master~2` crea `rama_y` en segundo ancestro de `master` y la restaura en direc. de trabajo

♦ **HEAD** es un puntero al commit que está restaurado en el directorio de trabajo

- **HEAD** se actualiza automáticamente al hacer el checkout



Comando reset



*S. Chacon, B. Straub: <https://git-scm.com/book/es/v1>

♦ `git reset <commit>`

- Cambia el puntero de rama y HEAD a <commit>. Deja las diferencias entre HEAD y <commit> en el directorio de trabajo. Es decir, restaura el contenido de commit, añadiendo las diferencias con <commit>
 - **Peligro!** Se pueden perder commits del grafo, si los eliminados no están guardados en otra rama.
- Ejemplos de uso
 - `git reset master~2` mueve HEAD y el puntero de rama al 2º ancestro, dejando las diferencias en el dir. de trabajo
 - **Nota:** Los cambios introducidos en los 2 commits que desaparecen de la rama quedan en los ficheros modified

♦ `git reset --hard <commit>`

- Cambia el puntero de rama y HEAD a <commit>.
 - **Muy peligroso!** Los commits y su código se pueden perder, si los eliminados no están guardados en otra rama
- Ejemplos de uso
 - `git reset --hard master~2` mueve HEAD y el puntero de rama al 2º ancestro
 - **Muy peligroso!** Los últimos 2 commits de la rama desaparecerían si no estuviesen en otra rama

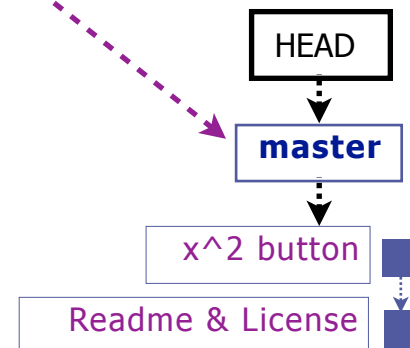
♦ El comando `git reset` permite compactar o eliminar commits del grafo

- normalmente se utiliza para eliminar, compactar o rehacer la última parte de la historia de una rama
 - Por ejemplo, si HEAD estuviese en master, `git reset --hard master~1` eliminaría el commit C8 (5558t8)
 - Como los cambios han quedado en el directorio de trabajo, estos se pueden integrar en un commit solo de master

Mostrar ramas

Muestra las ramas existentes después de haber creado los dos primeros commits. Solo existe la rama master, que Git creó automáticamente con el primer commit. El asterisco (*) indica que HEAD está en la rama master, que es la rama activa.

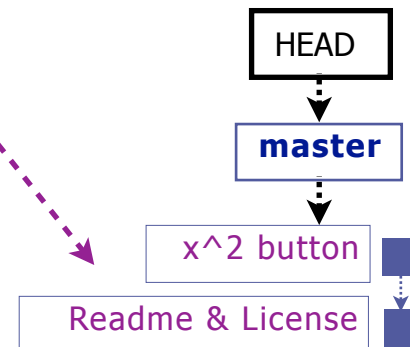
```
venus:cal jq$
venus:cal jq$ git branch
* master
venus:cal jq$
venus:cal jq$ git log --oneline
b0e63ad x^2 button
1096247 Readme & License
venus:cal jq$
venus:cal jq$ git branch inverse 1096247
venus:cal jq$
venus:cal jq$ git branch
    inverse
* master
venus:cal jq$ git branch -v
    inverse 1096247 Readme & License
* master b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git checkout inverse
Switched to branch 'inverse'
venus:cal jq$
venus:cal jq$ git branch
* inverse
  master
venus:cal jq$
```



Mostrar historia

```
venus:cal jq$  
venus:cal jq$ git branch  
* master  
venus:cal jq$  
venus:cal jq$ git log --oneline  
b0e63ad x^2 button  
1096247 Readme & License  
venus:cal jq$  
venus:cal jq$ git branch inverse 1096247  
venus:cal jq$  
venus:cal jq$ git branch  
inverse  
* master  
venus:cal jq$ git branch -v  
inverse 1096247 Readme & License  
* master b0e63ad x^2 button  
venus:cal jq$  
venus:cal jq$ git checkout inverse  
Switched to branch 'inverse'  
venus:cal jq$  
venus:cal jq$ git branch  
* inverse  
master  
venus:cal jq$
```

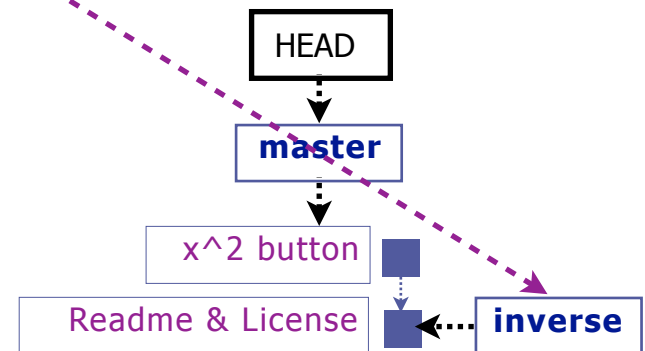
git log --oneline es equivalente a **git log --oneline HEAD** que muestra la historia de commits hasta **HEAD**.



Crear la rama inverse

```
venus:cal jq$  
venus:cal jq$ git branch  
* master  
venus:cal jq$  
venus:cal jq$ git log --oneline  
b0e63ad x^2 button  
1096247 Readme & License  
venus:cal jq$  
venus:cal jq$ git branch inverse 1096247  
venus:cal jq$  
venus:cal jq$ git branch  
inverse  
* master  
venus:cal jq$ git branch -v  
inverse 1096247 Readme & License  
* master b0e63ad x^2 button  
venus:cal jq$  
venus:cal jq$ git checkout inverse  
Switched to branch 'inverse'  
venus:cal jq$  
venus:cal jq$ git branch  
* inverse  
master  
venus:cal jq$
```

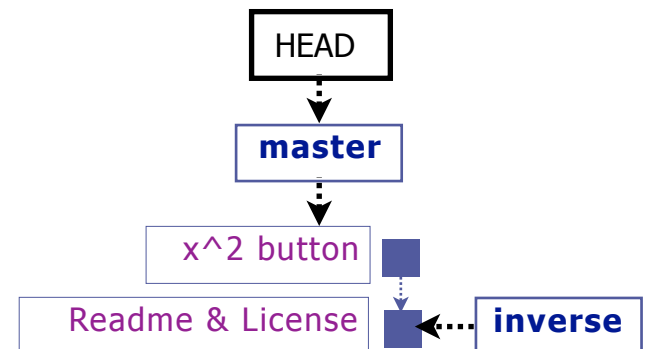
Crea la rama de nombre inverse con base en el commit **1096247** (equivalente a **HEAD~1**).



Mostrar ramas

```
venus:cal jq$  
venus:cal jq$ git branch  
* master  
venus:cal jq$  
venus:cal jq$ git log --oneline  
b0e63ad x^2 button  
1096247 Readme & License  
venus:cal jq$  
venus:cal jq$ git branch inverse 1096247  
venus:cal jq$  
venus:cal jq$ git branch  
inverse  
* master  
venus:cal jq$ git branch -v  
inverse 1096247 Readme & License  
* master b0e63ad x^2 button  
venus:cal jq$  
venus:cal jq$ git checkout inverse  
Switched to branch 'inverse'  
venus:cal jq$  
venus:cal jq$ git branch  
* inverse  
master  
venus:cal jq$
```

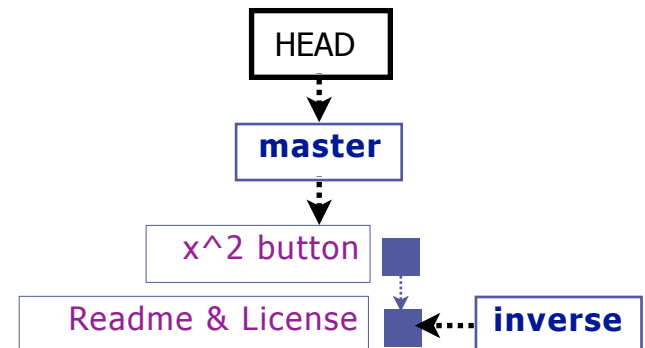
Muestra que ahora hay dos ramas, aunque la rama activa (*) sigue siendo master.



Mostrar ramas en modo verboso

```
venus:cal jq$  
venus:cal jq$ git branch  
* master  
venus:cal jq$  
venus:cal jq$ git log --oneline  
b0e63ad x^2 button  
1096247 Readme & License  
venus:cal jq$  
venus:cal jq$ git branch inverse 1096247  
venus:cal jq$  
venus:cal jq$ git branch  
inverse  
* master  
venus:cal jq$ git branch -v  
inverse 1096247 Readme & License  
* master b0e63ad x^2 button  
venus:cal jq$  
venus:cal jq$ git checkout inverse  
Switched to branch 'inverse'  
venus:cal jq$  
venus:cal jq$ git branch  
* inverse  
master  
venus:cal jq$
```

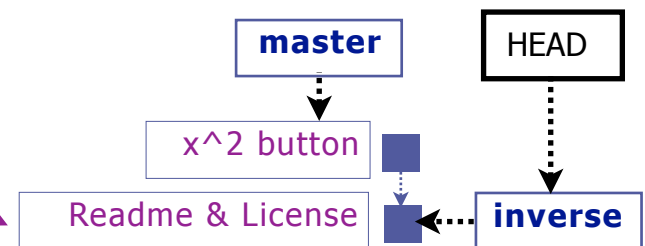
Muestra las dos ramas en modo verboso (-v), dando no solo el nombre de la rama, sino el id corto y el mensaje/título de su último commit.



Restaurar rama inverse

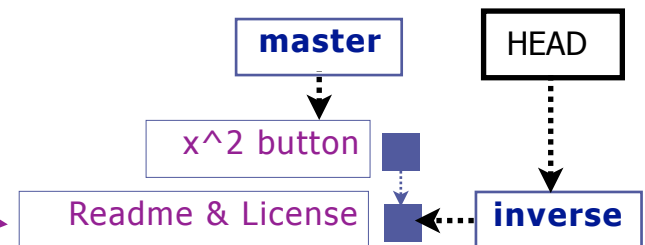
```
venus:cal jq$  
venus:cal jq$ git branch  
* master  
venus:cal jq$  
venus:cal jq$ git log --oneline  
b0e63ad x^2 button  
1096247 Readme & License  
venus:cal jq$  
venus:cal jq$ git branch inverse 1096247  
venus:cal jq$  
venus:cal jq$ git branch  
inverse  
* master  
venus:cal jq$ git branch -v  
inverse 1096247 Readme & License  
* master b0e63ad x^2 button  
venus:cal jq$  
venus:cal jq$ git checkout inverse  
Switched to branch 'inverse'  
venus:cal jq$  
venus:cal jq$ git branch  
* inverse  
master  
venus:cal jq$
```

Restaura la rama inverse en el directorio de trabajo, para poder trabajar en ella. HEAD referencia ahora rama inverse (el commit 1096247).



Mostrar ramas

```
venus:cal jq$  
venus:cal jq$ git branch  
* master  
venus:cal jq$  
venus:cal jq$ git log --oneline  
b0e63ad x^2 button  
1096247 Readme & License  
venus:cal jq$  
venus:cal jq$ git branch inverse 1096247  
venus:cal jq$  
venus:cal jq$ git branch  
    inverse  
* master  
venus:cal jq$ git branch -v  
    inverse 1096247 Readme & License  
* master b0e63ad x^2 button  
venus:cal jq$  
venus:cal jq$ git checkout inverse  
Switched to branch 'inverse'  
venus:cal jq$  
venus:cal jq$ git branch  
* inverse  
    master  
venus:cal jq$
```

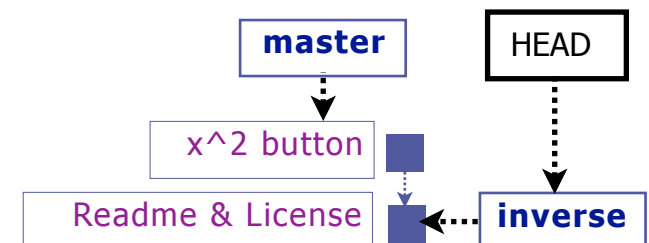


Mostrar ramas en modo verboso

```
venus:cal jq$  
venus:cal jq$ git branch -v  
* inverse 1096247 Readme & License  
  master b0e63ad x^2 button  
venus:cal jq$  
venus:cal jq$ git log --oneline  
1096247 Readme & License  
venus:cal jq$  
venus:cal jq$ ls  
LICENSE      README.md  
venus:cal jq$  
venus:cal jq$ # Edit calculator.html  
venus:cal jq$  
venus:cal jq$ ls  
LICENSE      README.md      calculator.html  
venus:cal jq$  
venus:cal jq$ git add .  
venus:cal jq$ git commit -m "1/x button"  
[inverse e868dc4] 1/x button  
 1 file changed, 17 insertions(+)  
 create mode 100644 calculator.html  
venus:cal jq$  
venus:cal jq$ git log --oneline --all --graph  
* e868dc4 1/x button  
| * b0e63ad x^2 button  
|/  
* 1096247 Readme & License  
venus:cal jq$
```

git branch -v muestra las dos ramas, incluyendo el id de commit asociado.

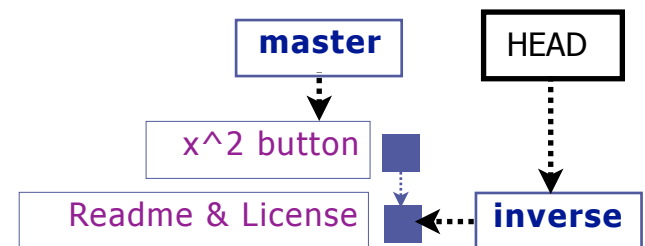
El asterisco indica que inverse es la rama activa.



Mostrar historia

```
venus:cal jq$
venus:cal jq$ git branch -v
* inverse 1096247 Readme & License
  master b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git log --oneline
1096247 Readme & License
venus:cal jq$
venus:cal jq$ ls
LICENSE      README.md
venus:cal jq$
venus:cal jq$ # Edit calculator.html
venus:cal jq$
venus:cal jq$ ls
LICENSE      README.md      calculator.html
venus:cal jq$
venus:cal jq$ git add .
venus:cal jq$ git commit -m "1/x button"
[inverse e868dc4] 1/x button
1 file changed, 17 insertions(+)
create mode 100644 calculator.html
venus:cal jq$
venus:cal jq$ git log --oneline --all --graph
* e868dc4 1/x button
| * b0e63ad x^2 button
|/
* 1096247 Readme & License
venus:cal jq$
```

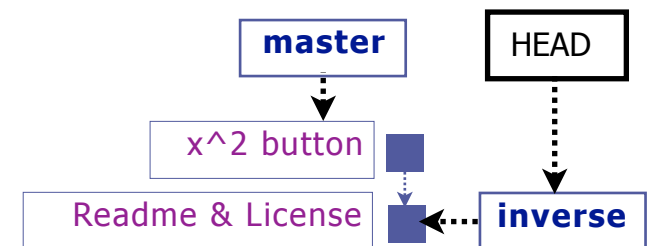
historia tiene ahora 1 commit.



Mostrar ficheros del directorio de trabajo

```
venus:cal jq$  
venus:cal jq$ git branch -v  
* inverse 1096247 Readme & License  
  master b0e63ad x^2 button  
venus:cal jq$  
venus:cal jq$ git log --oneline  
1096247 Readme & License  
venus:cal jq$  
venus:cal jq$ ls  
LICENSE README.md  
venus:cal jq$  
venus:cal jq$ # Edit calculator.html  
venus:cal jq$  
venus:cal jq$ ls  
LICENSE      README.md      calculator.html  
venus:cal jq$  
venus:cal jq$ git add .  
venus:cal jq$ git commit -m "1/x button"  
[inverse e868dc4] 1/x button  
1 file changed, 17 insertions(+)  
create mode 100644 calculator.html  
venus:cal jq$  
venus:cal jq$ git log --oneline --all --graph  
* e868dc4 1/x button  
| * b0e63ad x^2 button  
|/  
* 1096247 Readme & License  
venus:cal jq$
```

calculator.html no aparece en el directorio de trabajo, porque el commit 1096247 no lo incluyó.



Crear fichero: calculator.html

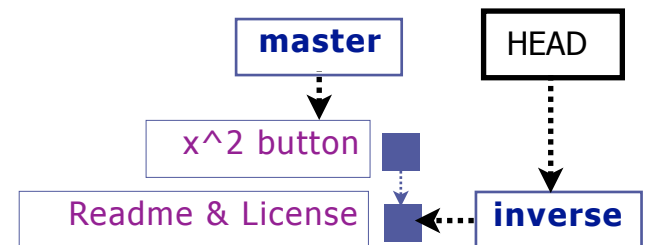
```
venus:cal jq$
venus:cal jq$ git branch -v
* inverse 1096247 Readme & License
  master b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git log --oneline
1096247 Readme & License
venus:cal jq$
venus:cal jq$ ls
LICENSE      README.md
venus:cal jq$
venus:cal jq$ # Edit calculator.html
venus:cal jq$
venus:cal jq$ ls
LICENSE      README.md      calculator.html
venus:cal jq$
venus:cal jq$ git add .
venus:cal jq$ git commit -m "1/x button"
[inverse e868dc4] 1/x button
1 file changed, 17 insertions(+)
create mode 100644 calculator.html
venus:cal jq$
venus:cal jq$ git log --oneline --all --graph
* e868dc4 1/x button
| * b0e63ad x^2 button
|/
* 1096247 Readme & License
venus:cal jq$
```

```
<!DOCTYPE html><html><head>
<title>Calculator</title><meta charset="utf-8">
<script type="text/javascript">

function inverse() {
  var num = document.getElementById("n1");
  num.value = 1/num.value;
}
</script>
</head>
<body>
  Number:
  <input type="text" id="n1" onclick="empty()"><p>

  <button onclick="inverse()"> 1/x </button>
</body>
</html>
```

Editamos la nueva calculadora en el dir. de trabajo.



Mostrar fich. del dir. de trabajo

```
venus:cal jq$
venus:cal jq$ git branch -v
* inverse 1096247 Readme & License
  master b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git log --oneline
1096247 Readme & License
venus:cal jq$
venus:cal jq$ ls
LICENSE      README.md
venus:cal jq$
venus:cal jq$ # Edit calculator.html
venus:cal jq$
venus:cal jq$ ls
LICENSE      README.md      calculator.html
venus:cal jq$
venus:cal jq$ git add .
venus:cal jq$ git commit -m "1/x button"
[inverse e868dc4] 1/x button
1 file changed, 17 insertions(+)
create mode 100644 calculator.html
venus:cal jq$
venus:cal jq$ git log --oneline --all --graph
* e868dc4 1/x button
| * b0e63ad x^2 button
|/
* 1096247 Readme & License
venus:cal jq$
```

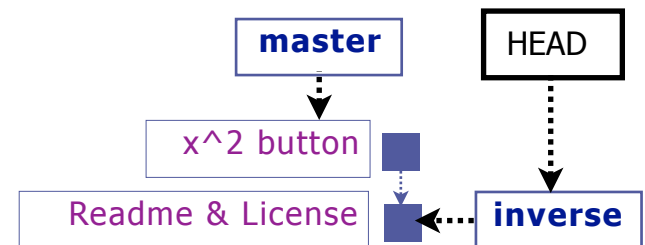
calculator.html

```
<!DOCTYPE html><html><head>
<title>Calculator</title><meta charset="utf-8">
<script type="text/javascript">

function inverse() {
  var num = document.getElementById("n1");
  num.value = 1/num.value;
}
</script>
</head>
<body>
  Number:
  <input type="text" id="n1" onclick="empty()"><p>

  <button onclick="inverse()"> 1/x </button>
</body>
</html>
```

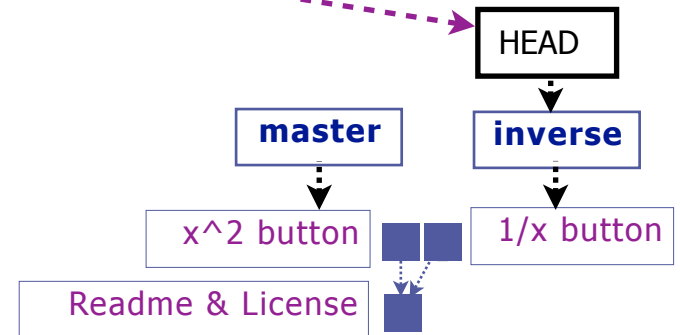
Después de crear **calculator.html** con el editor ls lo muestra, pero ahora tiene otro programa: el botón 1/x



Registrar en índice y crear commit

```
venus:cal jq$  
venus:cal jq$ git branch -v  
* inverse 1096247 Readme & License  
  master b0e63ad x^2 button  
venus:cal jq$  
venus:cal jq$ git log --oneline  
1096247 Readme & License  
venus:cal jq$  
venus:cal jq$ ls  
LICENSE      README.md  
venus:cal jq$  
venus:cal jq$ # Edit calculator.html  
venus:cal jq$  
venus:cal jq$ ls  
LICENSE      README.md      calculator.html  
venus:cal jq$  
venus:cal jq$ git add .  
venus:cal jq$ git commit -m "1/x button"  
[inverse e868dc4] 1/x button  
1 file changed, 17 insertions(+)  
create mode 100644 calculator.html  
venus:cal jq$  
venus:cal jq$ git log --oneline --all --graph  
* e868dc4 1/x button  
| * b0e63ad x^2 button  
|/  
* 1096247 Readme & License  
venus:cal jq$
```

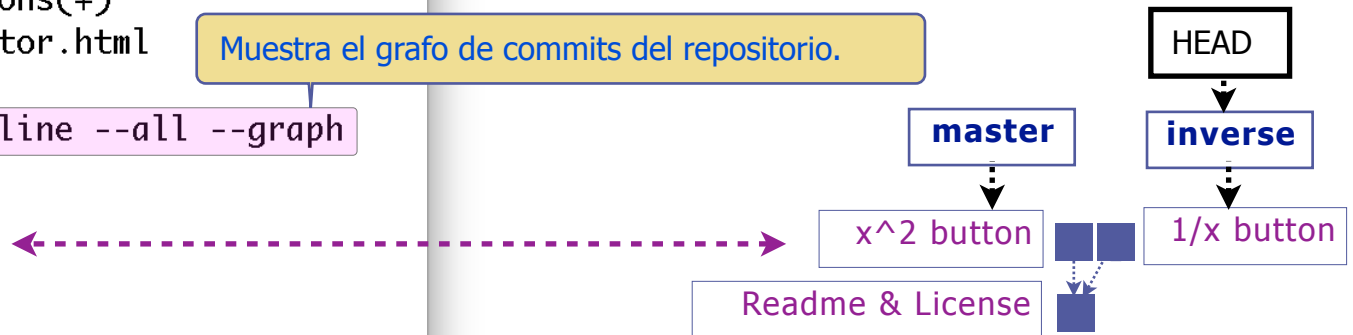
Registrar todos los cambios en el índice y generar el nuevo commit "1/x button".



Mostrar grafo de commits

```
venus:cal jq$  
venus:cal jq$ git branch -v  
* inverse 1096247 Readme & License  
  master b0e63ad x^2 button  
venus:cal jq$  
venus:cal jq$ git log --oneline  
1096247 Readme & License  
venus:cal jq$  
venus:cal jq$ ls  
LICENSE      README.md  
venus:cal jq$  
venus:cal jq$ # Edit calculator.html  
venus:cal jq$  
venus:cal jq$ ls  
LICENSE      README.md      calculator.html  
venus:cal jq$  
venus:cal jq$ git add .  
venus:cal jq$ git commit -m "1/x button"  
[inverse e868dc4] 1/x button  
1 file changed, 17 insertions(+)  
create mode 100644 calculator.html  
venus:cal jq$  
venus:cal jq$ git log --oneline --all --graph  
* e868dc4 1/x button  
| * b0e63ad x^2 button  
|/  
* 1096247 Readme & License  
venus:cal jq$
```

Muestra el grafo de commits del repositorio.



Ejemplo de show

```
venus:cal jq$  
venus:cal jq$ git show master  
commit b0e63ad0a57510d65814ac8ff98c7eb16bf359c1  
Author: Juan Quemada <jquemada@dit.upm.es>  
Date: Tue Feb 14 19:23:10 2017 +0100
```

Identificador largo del commit

Autor del commit

Fecha del commit

Mensaje o título del commit

x^2 button

calculator.html tiene los cambios indicados

Todo el código está en **verde**, porque el fichero es nuevo.
Todo se ha añadido en este commit.

```
diff --git a/calculator.html b/calculator.html  
new file mode 100644  
index 00000000..839b37e  
--- /dev/null  
+++ b/calculator.html  
@@ -0,0 +1,17 @@  
+<!DOCTYPE html><html><head>  
+<title>Calculator</title><meta charset="utf-8">  
+<script type="text/javascript">  
+  
+function square() {  
+  var num = document.getElementById("n1");  
+  num.value = num.value * num.value;  
+}  
+</script>  
+</head>  
+<body>  
+  Number:  
+  <input type="text" id="n1"><p>  
+  
+  <button onclick="square()"> x<sup>2</sup> </button>  
+</body>  
+</html>  
venus:cal jq$
```



```
<!DOCTYPE html><html><head>  
<title>Calculator</title>  
<script type="text/javascript">  
  
function square() {  
  var num = document.getElementById("n1");  
  num.value = num.value * num.value;  
}  
</script>  
</head>  
<body>  
  Number:  
  <input type="text" id="n1"><p>  
  
  <button onclick="square()"> x<sup>2</sup> </button>  
</body>  
</html>
```

Ejemplo de diff

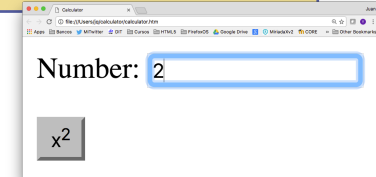
Muestra las diferencias entre los últimos commits de las ramas inverse y master.

```
venus:cal jq$
venus:cal jq$ git diff inverse master
diff --git a/calculator.html b/calculator.html
index fc74320..839b37e 100644
--- a/calculator.html
+++ b/calculator.html
@@ -2,9 +2,9 @@
<title>Calculator</title><meta charset="utf-8">
<script type="text/javascript">

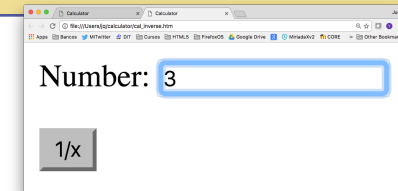
-function inverse() {
+function square() {
    var num = document.getElementById("n1");
-   num.value = 1/num.value;
+   num.value = num.value * num.value;
}
</script>
</head>
@@ -12,6 +12,6 @@ function inverse() {
    Number:
    <input type="text" id="n1"><p>

-   <button onclick="inverse()"> 1/x </button>
+   <button onclick="square()"> x<sup>2</sup> </button>
</body>
</html>
venus:cal jq$
```

b/calculator.html es del commit de master



a/calculator.html es del commit de inverse



Negro: código que delimita lo añadido o eliminado

Rojo: código eliminado del primer commit.

Verde: código añadido en el segundo commit.



Final del tema

Cuestionario

◆ Indicar que respuesta define correctamente el siguiente término:

Rama

- => 1. Commit a partir del cual se generó el commit indicado por modificación o integración
2. Grafo que muestra la relación de generación de todos los commits de todas las ramas de un repositorio
3. Secuencia de commits ordenada por fechas que siguiendo la relación de generación lleva a ese commit
4. Commit generado unicamente por modificación del anterior
5. Commit anterior en el tiempo que pertenece a la rama en que se genero dicho commit
6. Secuencia de commits ordenada por fechas que soporta un desarrollo
7. Comando que muestra el grafo de commits
8. Comando que muestra la historia de un commit
9. Comando que muestra los metadatos de un commit y las diferencias con el commit anterior
10. Comando que cambia el puntero de rama y HEAD al commit indicado dejando los cambios introducidos en los commits borrados en el directorio de trabajo
11. Comando que cambia el puntero de rama y HEAD al commit borrando commits
12. Comando que muestra las diferencias entre ficheros de dos commits diferentes

Grafo de commits => (las mismas respuestas)

Historia de un commit => (las mismas respuestas)

Commit con una flecha de salida => (las mismas respuestas)

Ancestro de un commit => (las mismas respuestas)

Padre de un commit => (las mismas respuestas)

git log --oneline <commit> => (las mismas respuestas)

git log --oneline --graph --all => (las mismas respuestas)

git show <commit> => (las mismas respuestas)

git reset <commit> => (las mismas respuestas)

git reset --hard <commit> => (las mismas respuestas)

git diff ... => (las mismas respuestas)

Cuestionario

◆ Indicar que respuesta define correctamente el siguiente término:

Rama (6)

- => 1. Commit a partir del cual se generó el commit indicado por modificación o integración
2. Grafo que muestra la relación de generación de todos los commits de todas las ramas de un repositorio
3. Secuencia de commits ordenada por fechas que siguiendo la relación de generación lleva a ese commit
4. Commit generado unicamente por modificación del anterior
5. Commit anterior en el tiempo que pertenece a la rama en que se genero dicho commit
6. Secuencia de commits ordenada por fechas que soporta un desarrollo
7. Comando que muestra el grafo de commits
8. Comando que muestra la historia de un commit
9. Comando que muestra los metadatos de un commit y las diferencias con el commit anterior
10. Comando que cambia el puntero de rama y HEAD al commit indicado dejando los cambios introducidos en los commits borrados en el directorio de trabajo
11. Comando que cambia el puntero de rama y HEAD al commit borrando commits
12. Comando que muestra las diferencias entre ficheros de dos commits diferentes

Grafo de commits (2)

=> (las mismas respuestas)

Historia de un commit (3)

=> (las mismas respuestas)

Commit con una flecha de salida (4)

=> (las mismas respuestas)

Ancestro de un commit (5)

=> (las mismas respuestas)

Padre de un commit (1)

=> (las mismas respuestas)

git log --oneline <commit> (8)

=> (las mismas respuestas)

git log --oneline --graph --all (7)

=> (las mismas respuestas)

git show <commit> (9)

=> (las mismas respuestas)

git reset <commit> (10)

=> (las mismas respuestas)

git reset --hard <commit> (11)

=> (las mismas respuestas)

git diff ... (12)

=> (las mismas respuestas)

Ejercicio opcional

- ◆ Clona en tu ordenador el repo **<tu _cuenta>/cal_2com** copiado con Fork en el tema anterior
 - **git clone** https://github.com/<tu_cuenta>/cal_2com
 - **git clone ..** copia el repositorio remoto en un repositorio local, con directorio de trabajo, de nombre **cal_2com**
 - Mostrar con **ls** el directorio que se acaba de crear y con **ls -a cal_2com** su contenido, incluyendo .git
- ◆ Entrar en el directorio de trabajo **cal_2com** del repositorio clonado
 - **cd cal_2com** para que los directorios de trabajo sean el mismo
 - Mostrar estado del directorio con **git status ..**, historia y grafo con **git log ..**, contenido con **ls ..**, etc.
- ◆ Crear la rama **mi_rama** en el commit **"Readme & License"**
 - Crearla con **git branch mi_rama <commit>** identificando el commit por id o por master~1
 - Restaurar la rama con **git checkout mi_rama** para poder empezar a hacer cambios en ella
- ◆ Copiar **calculadora.html** del commit **"x^2 button"** al directorio de trabajo
 - Copiarlo del commit x^2 button con "git show .." o descargarlo de GitHub o editarlo desde cero, etc.
- ◆ Cambiar el HTML y JavaScript de x^2 para que calcule el cubo y muestre x^3
 - Línea 5: **function square()** ... por **function cube()** ...
 - Línea 7: **... num.value * num.value;** por **... Math.pow(num.value, 3);**
 - Línea 14: **...onclick="square()"> x<sup>2<sup>...** por **...onclick="cube()"> x<sup>3<sup>...**
- ◆ Registrar con **git add .** y crear commit con **git commit -m "x^3 button"**
 - Mostrar la historia de la rama master **git log --oneline**
- ◆ Subir **mi_rama** al repositorio copiado con Fork con
 - **git push** https://github.com/<tu_cuenta>/cal_2com **mi_rama**
 - Inspeccionar con el navegador el repositorio actualizado y la nueva rama creada



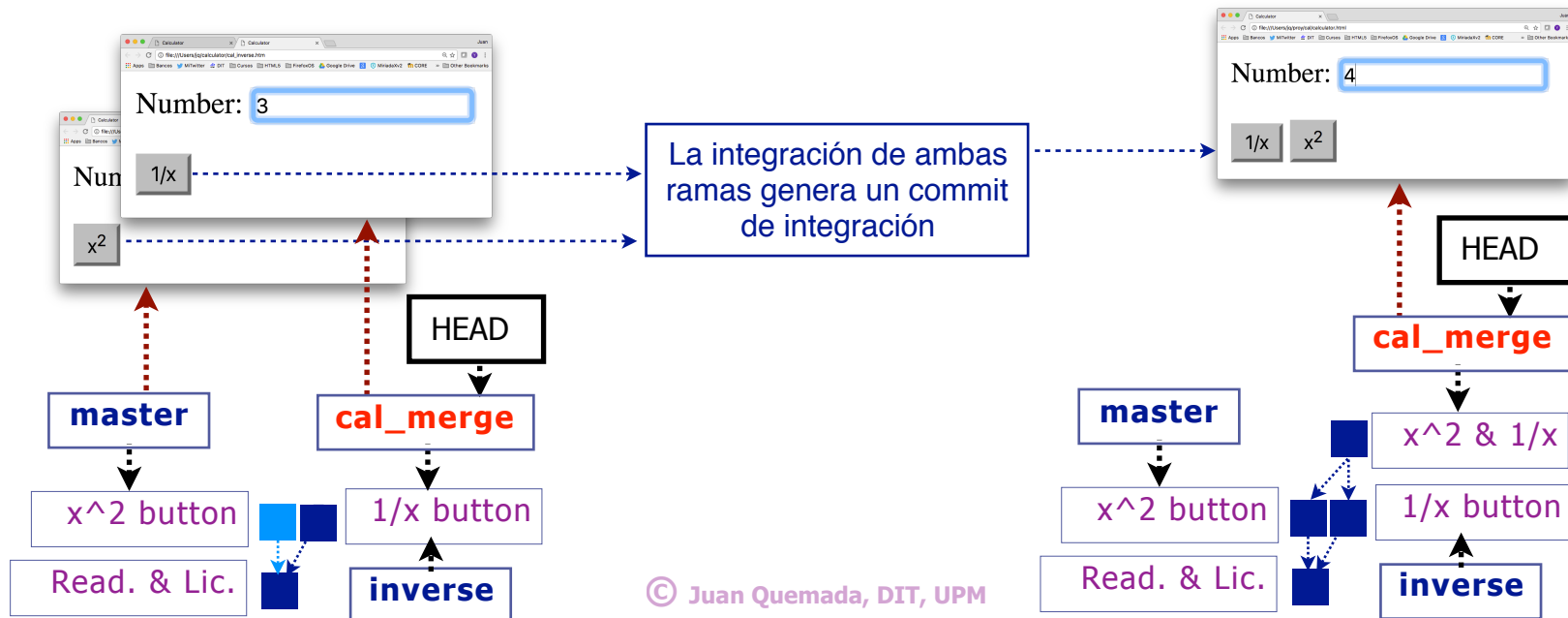
Git y GitHub

Integración de ramas: merge, commit, checkout y fast-forward

Juan Quemada, DIT - UPM

Integración de ramas con merge

- ♦ La figura ilustra la integración de la rama **master** en la rama **cal_merge** (inverse)
 - La integración se realiza con "**git merge master**" (estando en la rama **cal_merge**)
 - La integración es un proceso que puede tener conflictos que la interrumpen y hay que resolver
 - Una integración genera un **commit de integración** con **dos padres**, uno en cada rama
 - El commit de integración es una nueva calculadora con los dos botones integrados



Integración de commits: git merge

◆ **git merge -m "msg" <branch>**

- Integra <branch> en HEAD e incluye **"msg"** como descripción del commit de integración

◆ Git utiliza **estrategias de integración complejas** en función de los ancestros comunes, integrando fichero a fichero, aproximadamente así:

- Ficheros iguales en ambos commits: incluye el fichero común (**sin** conflicto)
- Fichero esta solo en uno de los commits: incluye el fichero (**sin** conflicto)
- Ficheros con diferencias en contextos disjuntos: integra ambos ficheros (**sin** conflicto)
- Ficheros con diferencias en el mismo contexto: une ambos ficheros y **genera conflicto**

◆ Si no hay conflictos, se **genera un commit de integración** de tipo **auto-merge**

- Además se debe comprobar que el **software resultante funciona bien!**

◆ Si hay algún **conflicto**, se **aborta** la integración y se generan los **"conflictos"**

- Los **conflictos deben resolverse** (con el editor y los cambios **registrarse en el índice**.
 - La integración debe completarse entonces con **git merge --continue** o con **git commit ...**
- **git merge --abort** deshace la integración y vuelve al estado anterior a invocar merge

Conflictos: estado de los ficheros

- ◆ Si hay conflictos no se genera el commit de integración
 - **Ficheros sin conflictos:** quedan en estado **staged**
 - **Ficheros con conflictos:** quedan en estado **unmerged**
 - Los conflictos quedan **marcados** en el código y deben resolverse con el editor
 - Una vez resueltos los conflictos, deben registrarse en el índice y cerrar el merge
 - El merge debe cerrarse con **git merge --continue** (o con **git commit ...**)

```
<title>Calculator</title><meta charset="utf-8">  
<script type="text/javascript">
```

```
++<<<<<<< HEAD  
+function inverse() {  
+  var num = document.getElementById("n1");  
+  num.value = 1/num.value;  
++=====  
+ function square() {  
+  var num = document.getElementById("n1");  
+  num.value = num.value * num.value;  
++>>>>>>> master  
}  
</script>  
</head>
```

El contexto (en negro) son las líneas idénticas de ambos ficheros que delimitan las líneas con conflicto.

La primera parte (HEAD) son las líneas de código conflictivas de la rama activa.

La segunda parte (master) son las líneas de código conflictivas de la rama a integrar (master).

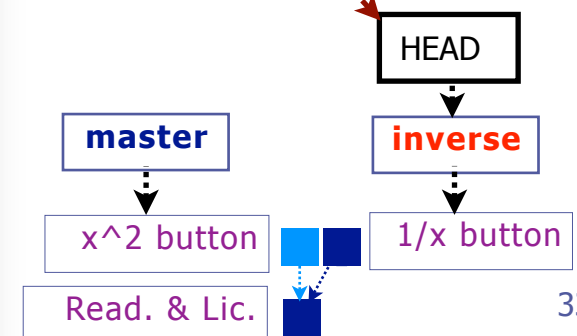
El contexto (en negro) son las líneas idénticas de ambos ficheros que delimitan las líneas con conflicto.

Conflicto queda marcado así en los ficheros fuente.

Mostrar ramas en modo verboso

```
venus:cal jq$
venus:cal jq$ git branch -v
* inverse e868dc4 1/x button
[ master b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git checkout -b cal_merge
Switched to a new branch 'cal_merge'
venus:cal jq$
venus:cal jq$ git branch -v
* cal_merge e868dc4 1/x button
[ inverse e868dc4 1/x button
[ master b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git merge master
Auto-merging calculator.html
CONFLICT (add/add): Merge conflict in calculator.html
Automatic merge failed; fix conflicts and then commit the result.
venus:cal jq$
venus:cal jq$ git status -s
AA calculator.html
venus:cal jq$
venus:cal jq$ # Fix conflicts with editor
venus:cal jq$ # -> in calculator.html
venus:cal jq$
venus:cal jq$ git commit -a -m "Integrate x^2 & 1/x"
[cal_merge 1898ac7] Integrate x^2 & 1/x
venus:cal jq$
venus:cal jq$ git log --oneline --graph
* 1898ac7 Integrate x^2 & 1/x
| \
| * b0e63ad x^2 button
| * | e868dc4 1/x button
|/
* 1096247 Readme & License
venus:cal jq$
```

Muestra ramas.



Cal — -Dash — 0 / x 30

100



Integrar master en inverse

Hay **conflicto** en el fichero **calculator.html** y la integración falla. Git pasa a un estado intermedio y no acabará la integración hasta el siguiente **git commit ...**
git merge --abort aborta la integración si por alguna causa no hubiese que finalizarla.

Intenta integrar automáticamente la rama **master** en **inverse**.

```
venus:cal jq$  
venus:cal jq$ git merge master  
Auto-merging calculator.html
```

```
CONFLICT (add/add): Merge conflict in calculator.html  
Automatic merge failed; fix conflicts and then commit the result.
```

```
venus:cal jq$  
venus:cal jq$ git status -s  
AA calculator.html  
venus:cal jq$  
venus:cal jq$ # Fix conflicts with editor  
venus:cal jq$ # -> in calculator.html  
venus:cal jq$  
venus:cal jq$ git commit -a -m "Integrate x^2 & 1/x"  
[cal_merge 1898ac7] Integrate x^2 & 1/x  
venus:cal jq$  
venus:cal jq$ git log --oneline --graph  
* 1898ac7 Integrate x^2 & 1/x  
| \  
| * b0e63ad x^2 button  
| * e868dc4 1/x button  
|/  
* 1096247 Readme & License  
venus:cal jq$
```

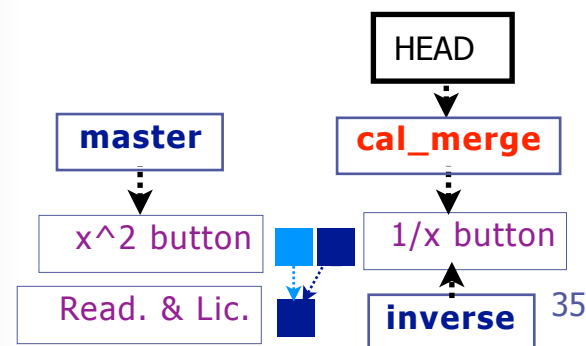
Muestra el fichero **calculator.html** en el estado AA, indicando que tiene conflictos.

```
<!DOCTYPE html><html><head>  
<title>Calculator</title><meta charset="utf-8">  
<script type="text/javascript">
```

```
<=====  
function inverse() {  
  var num = document.getElementById("n1");  
  num.value = 1/num.value;  
  =====  
function square() {  
  var num = document.getElementById("n1");  
  num.value = num.value * num.value;  
>>>>>> master  
}  
</script>  
</head>  
<body>  
  Number:  
  <input type="text" id="n1"><p>
```

```
<=====  
HEAD  
<button onclick="inverse()"> 1/x </button>  
=====  
<button onclick="square()"> x<sup>2</sup> </button>  
>>>>>> master  
</body>  
</html>
```

Conflictos en **calculator.html** en dir. de trab.



Resolver conflictos

```
venus:cal jq$
venus:cal jq$ git branch -v
* inverse e868dc4 1/x button
  master b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git checkout -b cal_merge
Switched to a new branch 'cal_merge'
venus:cal jq$
venus:cal jq$ git branch -v
* cal_merge e868dc4 1/x button
  inverse e868dc4 1/x button
  master b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git merge master
Auto-merging calculator.html
CONFLICT (add/add): Merge conflict in calculator.html
Automatic merge failed; fix conflicts and then commit the result.
venus:cal jq$
venus:cal jq$ git status -s
AA calculator.html
venus:cal jq$
venus:cal jq$ # Fix conflicts with editor
venus:cal jq$ # -> in calculator.html
venus:cal jq$
venus:cal jq$ git commit -a -m "Integrate x^2 & 1/x"
[cal_merge 1898ac7] Integrate x^2 & 1/x
venus:cal jq$
venus:cal jq$ git log --oneline --graph
* 1898ac7 Integrate x^2 & 1/x
| \
| * b0e63ad x^2 button
| * | e868dc4 1/x button
|/
* 1096247 Readme & License
venus:cal jq$
```

Number: 4

1/x

x²

```
<!DOCTYPE html><html><head>
<title>Calculator</title><meta charset="utf-8">
<script type="text/javascript">

function inverse() {
  var num = document.getElementById("n1");
  num.value = 1/num.value;
}

function square() {
  var num = document.getElementById("n1");
  num.value = num.value * num.value;
}

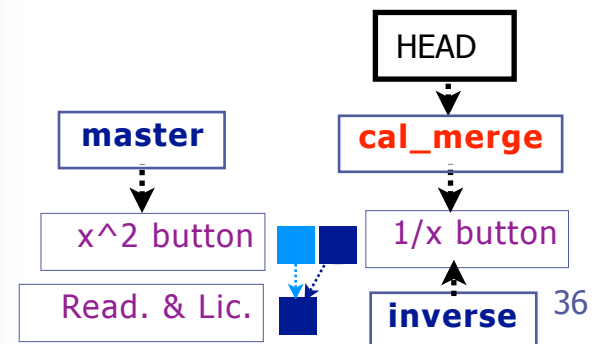
</script>
</head>
<body>
  Number:
  <input type="text" id="n1"><p>

  <button onclick="inverse()"> 1/x </button>
  <button onclick="square()"> x<sup>2</sup> </button>
</body>
</html>
```

Conflictos
resueltos en
calculator.html



Los conflictos se
resuelven con el
editor, integrando
x² y 1/x.



Crear commit de integración

```
venus:cal jq$
venus:cal jq$ git branch -v
* inverse e868dc4 1/x button
[ master b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git checkout -b cal_merge
Switched to a new branch 'cal_merge'
```

```
venus:cal jq$
venus:cal jq$ git branch -v
* cal_merge e868dc4 1/x button
[ inverse e868dc4 1/x button
[ master b0e63ad x^2 button
```

```
venus:cal jq$
venus:cal jq$ git merge master
Auto-merging calculator.html
CONFLICT (add/add): Merge conflict in calculator.html
Automatic merge failed; fix conflicts and then commit the result.
```

```
venus:cal jq$
venus:cal jq$ git status -s
AA calculator.html
venus:cal jq$
venus:cal jq$ # Fix conflicts with editor
venus:cal jq$ # -> in calculator.html
```

```
venus:cal jq$ git commit -a -m "Integrate x^2 & 1/x"
[cal_merge 1898ac7] Integrate x^2 & 1/x
```

```
venus:cal jq$
venus:cal jq$ git log --oneline --graph
```

```
* 1898ac7 Integrate x^2 & 1/x
```

```
| \
| * b0e63ad x^2 button
| * e868dc4 1/x button
| /
```

```
* 1096247 Readme & License
venus:cal jq$
```

Number: 4

1/x

x²

```
<!DOCTYPE html><html><head>
<title>Calculator</title><meta charset="utf-8">
<script type="text/javascript">

function inverse() {
  var num = document.getElementById("n1");
  num.value = 1/num.value;
}

function square() {
  var num = document.getElementById("n1");
  num.value = num.value * num.value;
}

</script>
</head>
<body>
  Number:
  <input type="text" id="n1"><p>

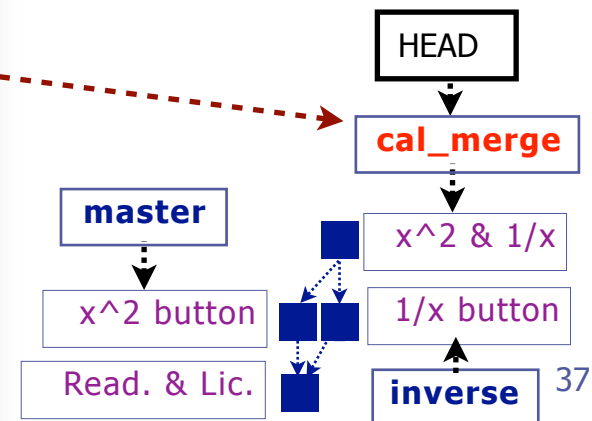
  <button onclick="inverse()"> 1/x </button>
  <button onclick="square()"> x<sup>2</sup> </button>
</body>
</html>
```

git commit -a -m "..." finaliza la integración.
opción -a añade ficheros modificados al índice y cierra commit.

También se podría haber utilizado
git commit --continue

git log --oneline --graph muestra el grafo de integración de la **rama cal_merge**. El commit de integración junta las 2 líneas del desarrollo. Une las 2 ramas en cal_merge.

© Juan Quemada, DIT, UPM



Conflictos vistos con diff

git diff muestra así los conflictos resueltos, **después** de integrarlos con un editor y antes de cerrar el commit.

```
venus:cal jq$
venus:cal jq$ git diff
diff --cc calculator.html
index fc74320,839b37e..0000000
--- a/calculator.html
+++ b/calculator.html
@@@ -2,9 -2,9 +2,15 @@@
<title>Calculator</title><meta charset="utf-8">
<script type="text/javascript">

++<<<<<<< HEAD
+function inverse() {
+  var num = document.getElementById("n1");
+  num.value = 1/num.value;
++=====
+ function square() {
+   var num = document.getElementById("n1");
+   num.value = num.value * num.value;
++>>>>>>> master
+
+ }
+ </script>
+ </head>
@@@ -12,6 -12,6 +18,10 @@@
Number:
<input type="text" id="n1"><p>

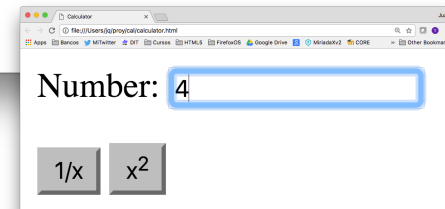
++<<<<<<< HEAD
+ <button onclick="inverse()"> 1/x </button>
++=====
+ <button onclick="square()"> x<sup>2</sup> </button>
++>>>>>>> master
+
+ </body>
+ </html>
venus:cal jq$
```

git diff muestra así los conflictos **antes** de resolverlos.

Resolver conflictos con un **editor**.

```
venus:cal jq$ git diff
diff --cc calculator.html
index fc74320,839b37e..0000000
--- a/calculator.html
+++ b/calculator.html
@@@ -2,16 -2,16 +2,22 @@@
<title>Calculator</title><meta charset='utf-8">
<script type="text/javascript">

+function inverse() {
+  var num = document.getElementById("n1");
+  num.value = 1/num.value;
+}
++
+ function square() {
+   var num = document.getElementById("n1");
+   num.value = num.value * num.value;
+ }
+
+ </script>
+ </head>
+ <body>
+   Number:
+   <input type="text" id="n1"><p>
+
+ <button onclick="inverse()"> 1/x </button>
+ <button onclick="square()"> x<sup>2</sup> </button>
+
+ </body>
+ </html>
venus:cal jq$
```

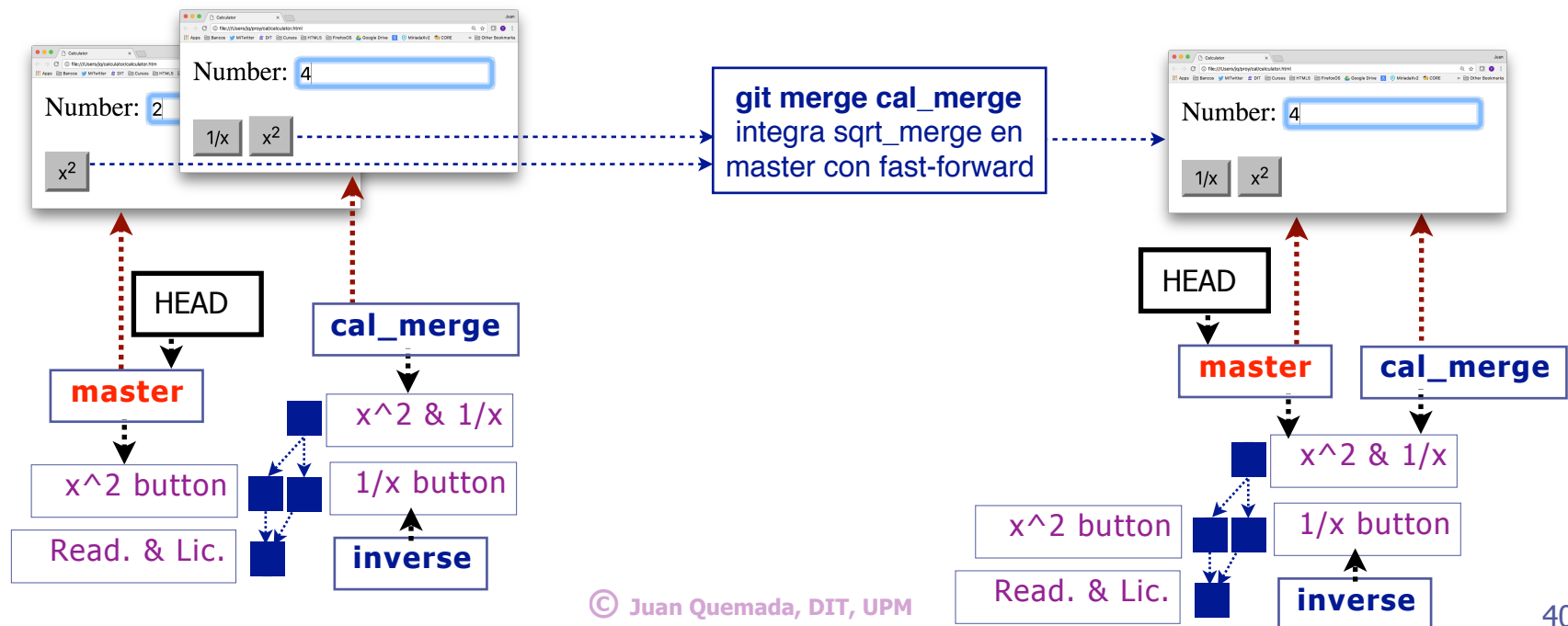




Integración ff (fast-forward)

Integración ff (fast-forward)

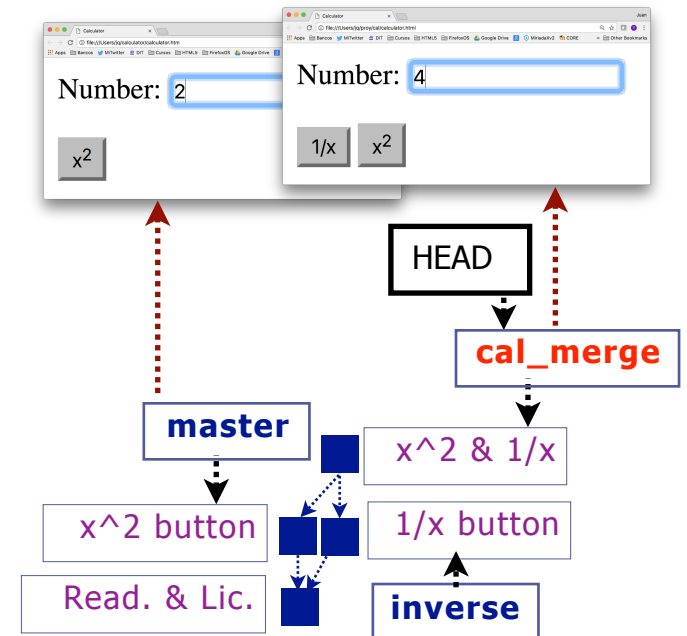
- ♦ **git merge** ... detecta si la integración pedida se ha realizado ya en otra rama
 - Si es así, Git está configurado para realizar una integración ff (Fast-Forward)
 - Simplemente avanzará el puntero de rama hasta el commit que contiene la integración solicitada
 - No se generará un nuevo commit, se reutilizará el existente
- ♦ Al integrar la rama **cal_merge** en la rama **master**
 - Se reutiliza con **fast-forward** la integración realizada en la rama **cal_merge**
 - Porque **master** se ha integrado ya en **cal_merge**, tal y como muestra el grafo de commits



Mostrar ramas

git branch muestra las 3 ramas existentes, marcando la rama **cal_merge** como activa.

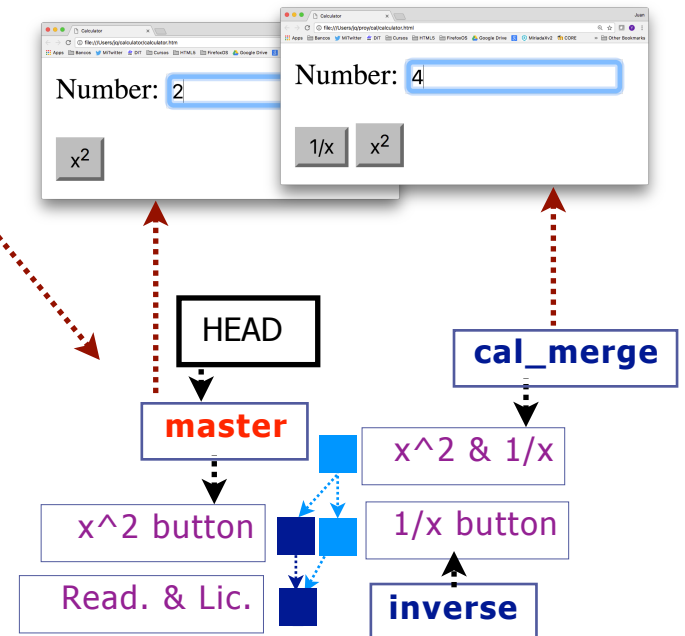
```
venus:cal jq$
venus:cal jq$ git branch -v
* cal_merge 1898ac7 Integrate x^2 & 1/x
  inverse   e868dc4 1/x button
  master    b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git checkout master
Switched to branch 'master'
venus:cal jq$
venus:cal jq$ git branch square
venus:cal jq$
venus:cal jq$ git branch -v
  cal_merge 1898ac7 Integrate x^2 & 1/x
  inverse   e868dc4 1/x button
* master    b0e63ad x^2 button
  square    b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git merge cal_merge
Updating b0e63ad..1898ac7
Fast-forward
 calculator.html | 6 ++++++
 1 file changed, 6 insertions(+)
venus:cal jq$
venus:cal jq$ git log --oneline --graph
* 1898ac7 Integrate x^2 & 1/x
| \
| * b0e63ad x^2 button
| * | e868dc4 1/x button
|/
* 1096247 Readme & License
venus:cal jq$
```



Restaurar rama master

```
venus:cal jq$
venus:cal jq$ git branch -v
* cal_merge 1898ac7 Integrate x^2 & 1/x
  inverse   e868dc4 1/x button
  master    b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git checkout master
Switched to branch 'master'
venus:cal jq$
venus:cal jq$ git branch square
venus:cal jq$
venus:cal jq$ git branch -v
  cal_merge 1898ac7 Integrate x^2 & 1/x
  inverse    e868dc4 1/x button
* master     b0e63ad x^2 button
  square     b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git merge cal_merge
Updating b0e63ad..1898ac7
Fast-forward
 calculator.html | 6 ++++++
 1 file changed, 6 insertions(+)
venus:cal jq$
venus:cal jq$ git log --oneline --graph
* 1898ac7 Integrate x^2 & 1/x
| \
| * b0e63ad x^2 button
| * | e868dc4 1/x button
| /
* 1096247 Readme & License
venus:cal jq$
```

git checkout master restaura la rama master para trabajar en ella.

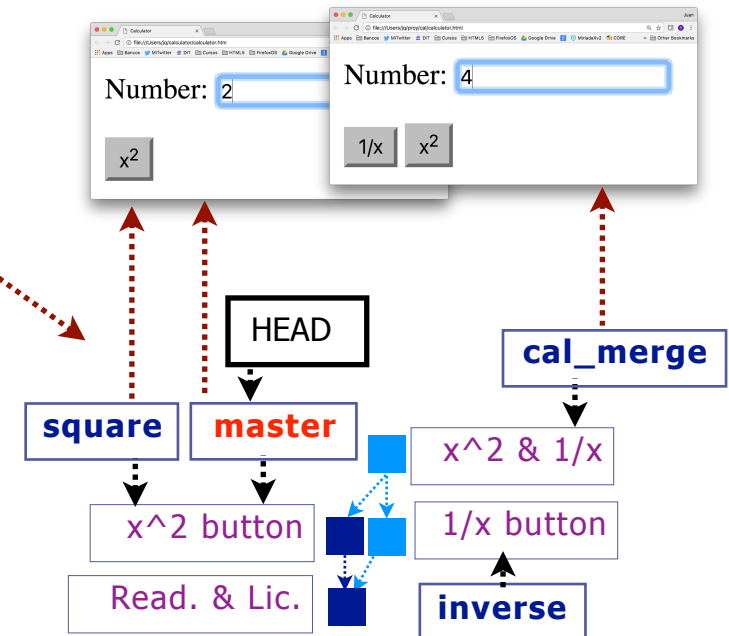


Crear rama square y mostrarla

```
venus:cal jq$
venus:cal jq$ git branch -
* cal_merge 1898ac7 Integr
  inverse    e868dc4 1/x button
  master     b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git checkout master
Switched to branch 'master'
venus:cal jq$
venus:cal jq$ git branch square
venus:cal jq$
venus:cal jq$ git branch -v
  cal_merge 1898ac7 Integrate x^2 & 1/x
  inverse    e868dc4 1/x button
* master     b0e63ad x^2 button
  square     b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git merge cal_merge
Updating b0e63ad..1898ac7
Fast-forward
 calculator.html | 6 ++++++
 1 file changed, 6 insertions(+)
venus:cal jq$
venus:cal jq$ git log --oneline --graph
* 1898ac7 Integrate x^2 & 1/x
| \
| * b0e63ad x^2 button
| * | e868dc4 1/x button
| /
* 1096247 Readme & License
venus:cal jq$
```

git branch square crea una rama **square** con base en master (b0e63ad), que permitirá acceder a este commit cuando master avance.

git branch muestra ahora las 4 ramas y **master** como activa.



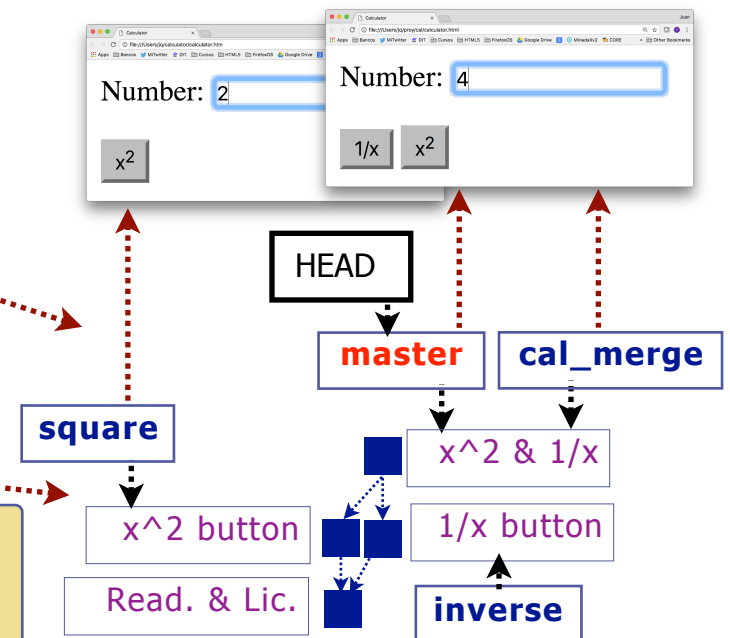
Integrar en rama master

```
venus:cal jq$
venus:cal jq$ git branch -v
* cal_merge 1898ac7 Integrate x^2 & 1/x
  inverse   e868dc4 1/x button
  master    b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git checkout master
Switched to branch 'master'
venus:cal jq$
venus:cal jq$ git branch square
venus:cal jq$
venus:cal jq$ git branch -v
  cal_merge 1898ac7 Integrate x^2 & 1/x
  inverse    e868dc4 1/x button
* master     b0e63ad x^2 button
  square     b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git merge cal_merge
Updating b0e63ad..1898ac7
Fast-forward
 calculator.html | 6 ++++++
 1 file changed, 6 insertions(+)
venus:cal jq$
venus:cal jq$ git log --oneline --graph
* 1898ac7 Integrate x^2 & 1/x
| \
| * b0e63ad x^2 button
| * | e868dc4 1/x button
|/
* 1096247 Readme & License
venus:cal jq$
```

git merge cal_merge integra la rama **cal_merge** en la rama activa (**master**).

La integración es **ff** o **fast-forward**, porque el commit a integrar está ya integrado con el último de master, tal y como indica el grafo.

Ambas ramas comparten ahora el mismo commit con el mismo identificador.



git log --oneline --graph muestra ahora el mismo grafo de commits que **cal_merge**. Ambas ramas están en el mismo commit. **cal_merge** se podría borrar ahora sin perder información.



Final del tema