Gestión de proyectos con Git y GitHub: Modulo 5

Temas:

Juan Quemada, DIT - UPM

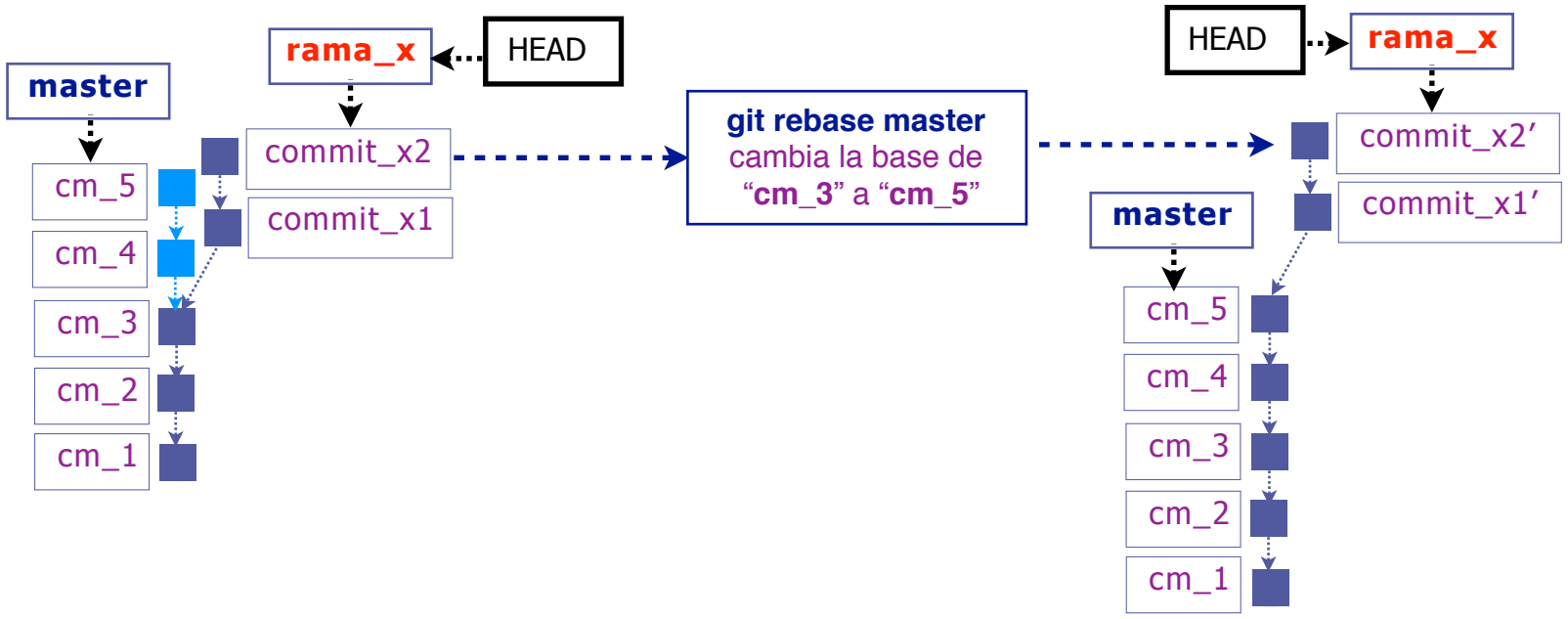
Git y GitHub

Integración de ramas con rebase

Juan Quemada, DIT - UPM

Cambiar la base de una rama

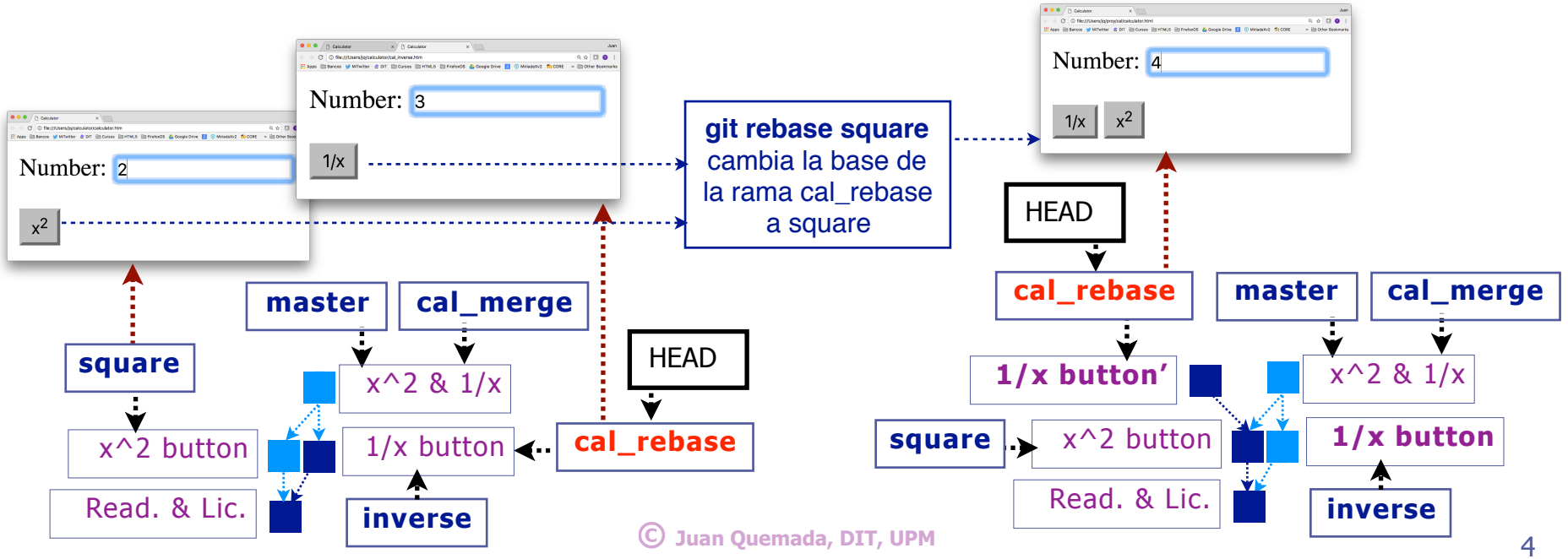
- Una rama se usa para hacer desarrollos separados de la rama master
  - Al finalizar el desarrollo de la rama, se puede integrar con master o con otra rama
- Cambiar la base de una rama permite también integrar desarrollos
  - Cambiar de base (**rebase)** integra los desarrollos linealmente (muy limpio)
    - Pero **elimina la historia** de ramas utilizadas e integradas para el desarrollo



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

Cambiar la base de la rama cal_rebase

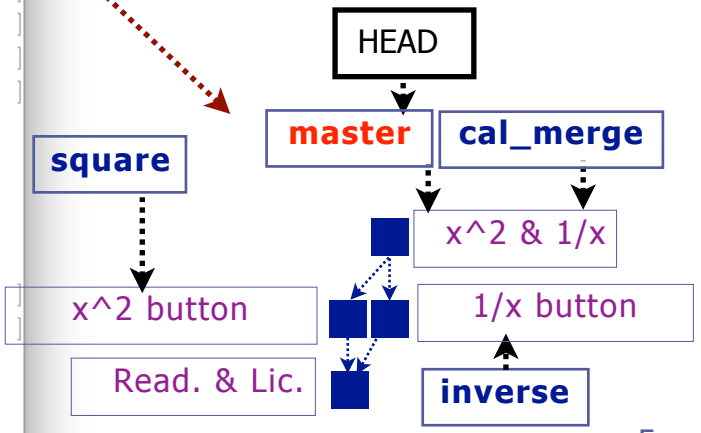
- **git rebase square**
  - Arranca un bucle que cambia la base de la rama cal_rebase (HEAD) a square
    - Como el rebase solo tiene que integrar un commit, aquí solo habra una iteración al bucle
  - El bucle integra el commit "1/x button" en "x^2 button", para su nueva base
    - Si hay conflictos, se aborta el proceso, marcando los conflictos, que deben resolverse con el editor
      - Una vez registrados los cambios a integrar en el índice y se continua con: **git rebase --continue**
  - Al finalizar el bucle la rama ha cambiado la base **"Read. & Lic."** por **"x^2 button"**
- **git rebase -i ...** permite modificación interactiva de la rama



Mostrar ramas

```
cal — -bash — 78×41
venus:cal jq$
venus:cal jq$ git branch -v
  cal_merge 1898ac7 Integrate x^2 & 1/x
  inverse   e868dc4 1/x button
* master    1898ac7 Integrate x^2 & 1/x
  square    b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git checkout -b cal_rebase inverse
Switched to a new branch 'cal_rebase'
venus:cal jq$
venus:cal jq$ git rebase -q square
Failed to merge in the changes.
Patch failed at 0001 1/x button
The copy of the patch that failed is found in:
   /Users/jq/proy/cal/.git/rebase-apply/patch

When you have resolved this problem, run "git rebase --continue".
If you prefer to skip this patch, run "git rebase --skip" instead.
To check out the original branch and stop rebasing, run "git rebase --abort".

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 add .
venus:cal jq$ git rebase --continue
venus:cal jq$
venus:cal jq$ git branch -v
  cal_merge  1898ac7 Integrate x^2 & 1/x
* cal_rebase 795c2da 1/x button
  inverse    e868dc4 1/x button
  master     1898ac7 Integrate x^2 & 1/x
  square     b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git log --oneline --graph
* 795c2da 1/x button
* b0e63ad x^2 button
* 1096247 Readme & License
venus:cal jq$
```

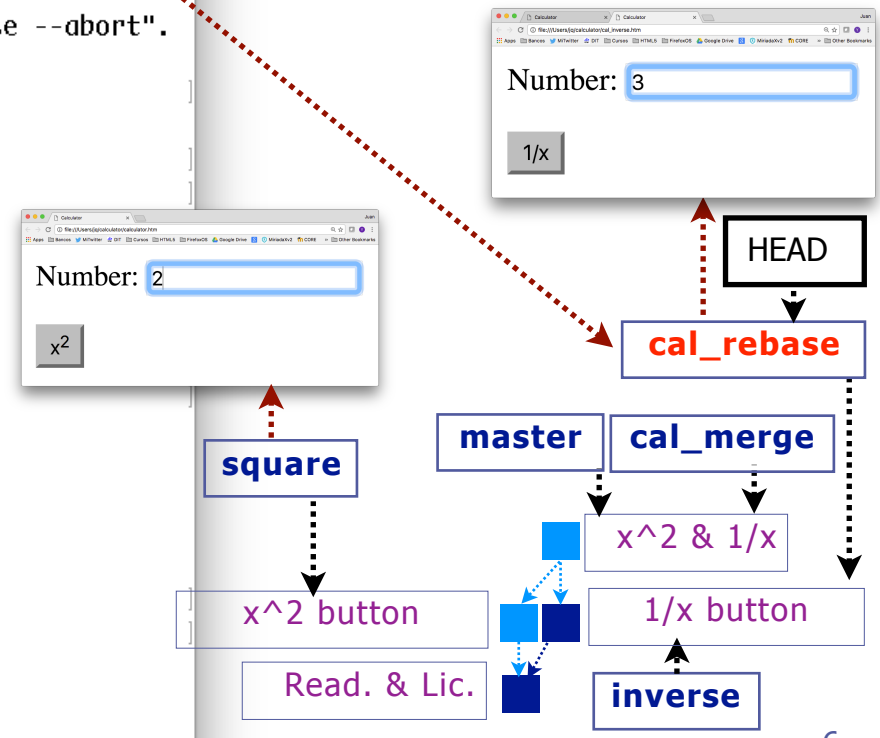
git branch muestra 4 ramas y marca **master** como activa.



Crear y restaurar cal_rebase

Crea la **rama cal_rebase** en inverse y realiza un checkout a cal_rebase.

```
cal — -bash — 78×41
venus:cal jq$
venus:cal jq$ git branch -v
  cal_merge 1898ac7 Integrate x^2 & 1/x
  inverse   e868dc4 1/x button
* master    1898ac7 Integrate x^2 & 1/x
  square    b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git checkout -b cal_rebase inverse
Switched to a new branch 'cal_rebase'
venus:cal jq$
venus:cal jq$ git rebase -q square
Failed to merge in the changes.
Patch failed at 0001 1/x button
The copy of the patch that failed is found in:
   /Users/jq/proy/cal/.git/rebase-apply/patch

When you have resolved this problem, run "git rebase --continue".
If you prefer to skip this patch, run "git rebase --skip" instead.
To check out the original branch and stop rebasing, run "git rebase --abort".

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 add .
venus:cal jq$ git rebase --continue
venus:cal jq$
venus:cal jq$ git branch -v
  cal_merge  1898ac7 Integrate x^2 & 1/x
* cal_rebase 795c2da 1/x button
  inverse    e868dc4 1/x button
  master     1898ac7 Integrate x^2 & 1/x
  square     b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git log --oneline --graph
* 795c2da 1/x button
* b0e63ad x^2 button
* 1096247 Readme & License
venus:cal jq$
```



Comenzar cambio de base

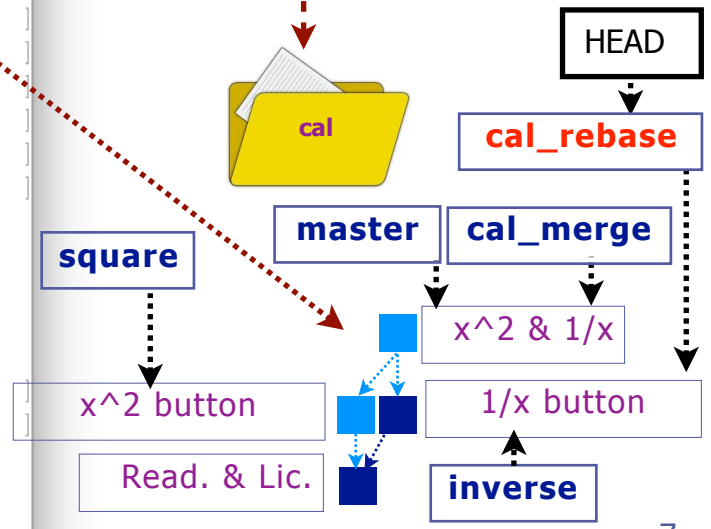
```
venus:cal jq
venus:cal jq
  cal_merge
  inverse    e868dc4 1/x button
* master     1898ac7 Integrate x^2 & 1/x
  square     b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git checkout -b cal_rebase inver
Switched to a new branch 'cal_rebase'
venus:cal jq$
venus:cal jq$ git rebase -q square
Failed to merge in the changes.
Patch failed at 0001 1/x button
The copy of the patch that failed is found in:
   /Users/jq/proy/cal/.git/rebase-apply/patch

When you have resolved this problem, run "git rebase --continue".
If you prefer to skip this patch, run "git rebase --skip" instead.
To check out the original branch and stop rebasing, run "git rebase --abort".

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 add .
venus:cal jq$ git rebase --continue
venus:cal jq$
venus:cal jq$ git branch -v
  cal_merge  1898ac7 Integrate x^2
* cal_rebase 795c2da 1/x button
  inverse    e868dc4 1/x button
  master     1898ac7 Integrate x^2
  square     b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git log --oneline --
* 795c2da 1/x button
* b0e63ad x^2 button
* 1096247 Readme & License
venus:cal jq$
```

git rebase -q square inicia el cambio de base de la rama **cal_rebase**, del commit **"Readme & License"** al commit **"x^2 button"**.
-q (quiet) limita estadísticas.

git status muestra el conflicto.

OJO! Hay **conflicto** en el fichero **calculator.html** y el cambio de base se interrumpe para integración manual con el editor.
Git da instrucciones sobre como continuar.

```
<!DOCTYPE html><html><head>
<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>
<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>
```



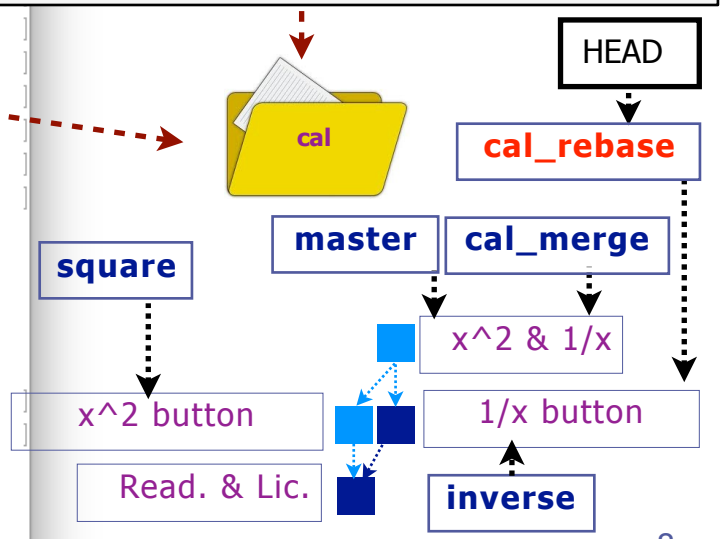
Resolver conflictos

```
venus:cal jq$
venus:cal jq$ git branch -v
  cal_merge 1898ac7 Integrate x^2 & 1/x
  inverse   e868dc4 1/x button
* master    1898ac7 Integrate x^2 & 1/x
  square    b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git checkout -b cal_rebase inverse
Switched to a new branch 'cal_rebase'
venus:cal jq$
venus:cal jq$ git rebase -q square
Failed to merge in the changes.
Patch failed at 0001 1/x button
The copy of the patch that failed is found in:
   /Users/jq/proy/cal/.git/rebase-apply/patch

When you have resolved this problem, run "git rebase --continue".
If you prefer to skip this patch, run "git rebase --skip" instead.
To check out the original branch and stop rebasing, run "git rebase --ab

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 add .
venus:cal jq$ git rebase --continue
venus:cal jq$
venus:cal jq$ git branch -v
  cal_merge  1898ac7 Integrate x^2 & 1/x
* cal_rebase 795c2da 1/x button
  inverse    e868dc4 1/x button
  master     1898ac7 Integrate x^2 & 1/x
  square     b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git log --oneline --graph
* 795c2da 1/x button
* b0e63ad x^2 button
* 1096247 Readme & License
venus:cal jq$
```

Number: 4

1/x x^2

```
<!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>
```



Los conflictos se resuelven con el editor, integrando **x^2** y **1/x.**

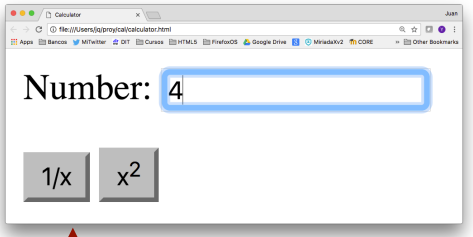
Finalizar rebase, mostrar ramas y grafo

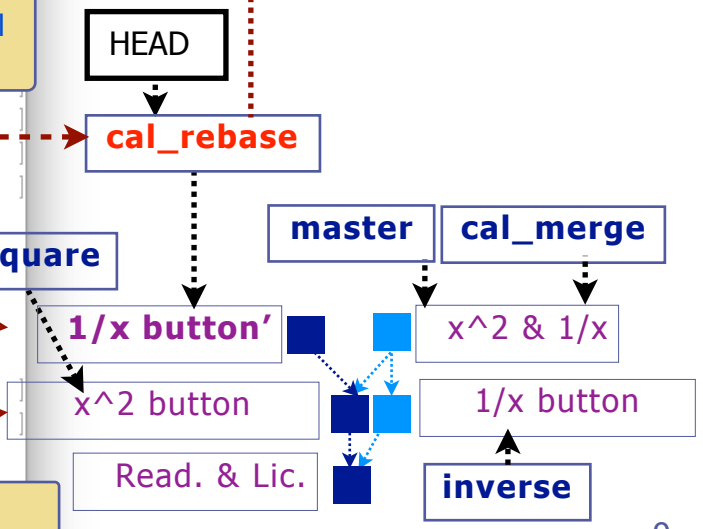
```
venus:cal jq$
venus:cal jq$ git branch -v
  cal_merge 1898ac7 Integrate
  inverse   e868dc4 1/x button
* master    1898ac7 Integrate x^2 & 1/x
  square    b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git checkout -b cal_rebase
Switched to a new branch 'cal_rebase'
venus:cal jq$
venus:cal jq$ git rebase -q square
Failed to merge in the changes.
Patch failed at 0001 1/x button
The copy of the patch that failed is found in:
   /Users/jq/proy/cal/.git/rebase-apply/patch

When you have resolved this problem, run "git rebase --continue".
If you prefer to skip this patch, run "git rebase --skip" instead.
To check out the original branch and stop rebasing, run "git rebase --abort".

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 add .
venus:cal jq$ git rebase --continue
venus:cal jq$
venus:cal jq$ git branch -v
  cal_merge  1898ac7 Integrate x^2 & 1/x
* cal_rebase 795c2da 1/x button
  inverse    e868dc4 1/x button
  master     1898ac7 Integrate x^2 & 1/x
  square     b0e63ad x^2 button
venus:cal jq$
venus:cal jq$ git log --oneline --graph
* 795c2da 1/x button
* b0e63ad x^2 button
* 1096247 Readme & License
venus:cal jq$
```

git add . añade las modificaciones al índice.
git rebase --continue finaliza la adaptación del primer commit a la nueva base y finaliza el bucle porque solo hay 1 commit.

git branch muestra 5 ramas y **cal_rebase** activa. Hacemos notar, que el commit generado por rebase tiene el mismo mensaje/título que el original, pero distinto identificador.

git log --oneline --graph muestra como rebase ha generado una historia lineal de commits en **cal_rebase**.





Conflictos vistos con diff

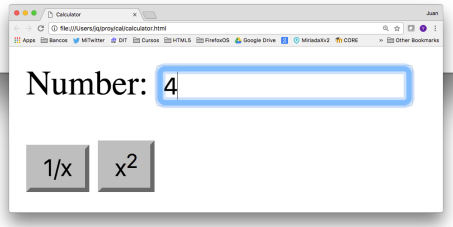
git diff muestra así los conflictos resueltos, **después** de integrarlos con un editor y antes de cerrar el commit.

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

Resolver conflictos con un **editor**.

```
cal — -bash — 58×34
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$
```

```
cal — -bash — 57×29
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$
```



Actualizar repositorio jquemada/cal en GitHub

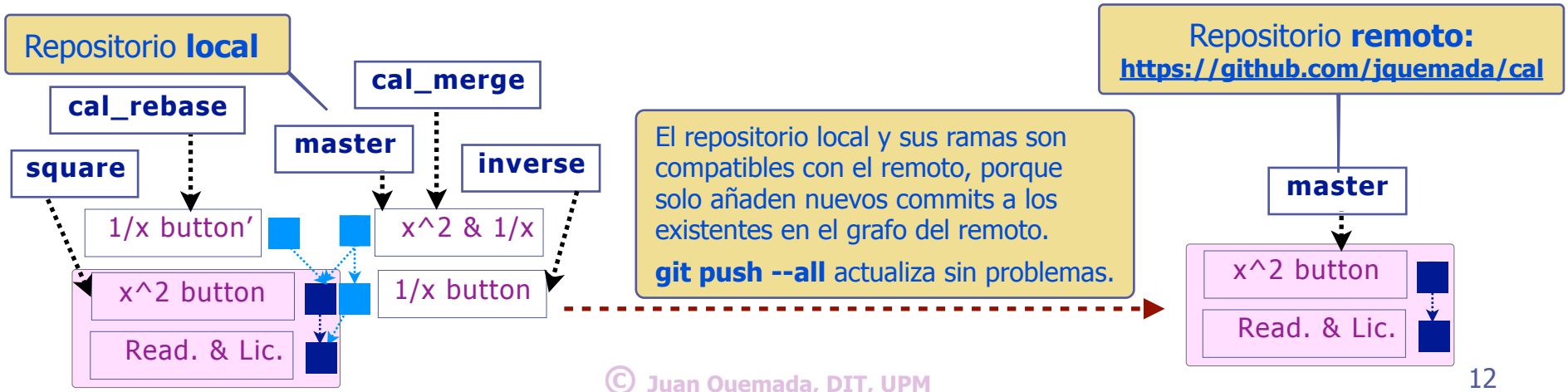
Actualizar un repositorio en GitHub con push

✦ git push …

- Actualiza el repositorio remoto con los nuevos commits de una rama local
  - **git push https://github.com/jquemada/cal master**
    - actualiza los nuevos commits de la rama master en el repositorio GitHub **jquemada/cal**
  - **git push https://github.com/jquemada/cal_2com master**
    - actualiza los nuevos commits de la rama master en el repositorio GitHub **jquemada/cal_2com**

✦ git push … necesita 2 condiciones para finalizar con éxito

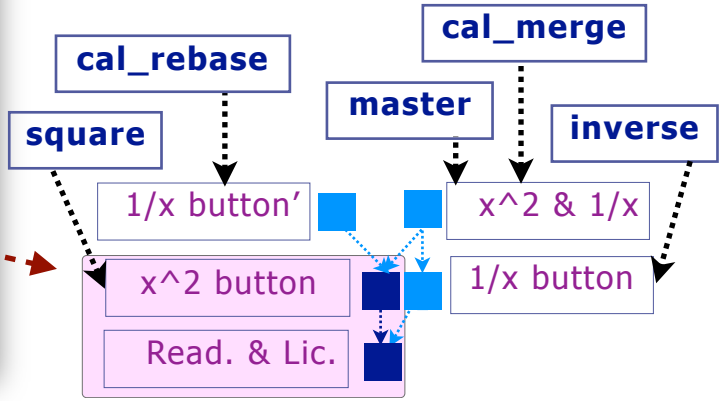
- Se debe tener credenciales de acceso al repositorio remoto
  - Por ejemplo, un repositorio en una cuenta u organización del usuario que lo actualiza
- La actualización de commits debe ser compatible con la rama actualizada en el remoto
  - Solo debe **añadir nuevos commits al final de la rama remota** o actualizar un **repositorio vacío**
    - **Peligroso!** La **opción -f** permite actualizar una rama incompatible, pero se pierden commits



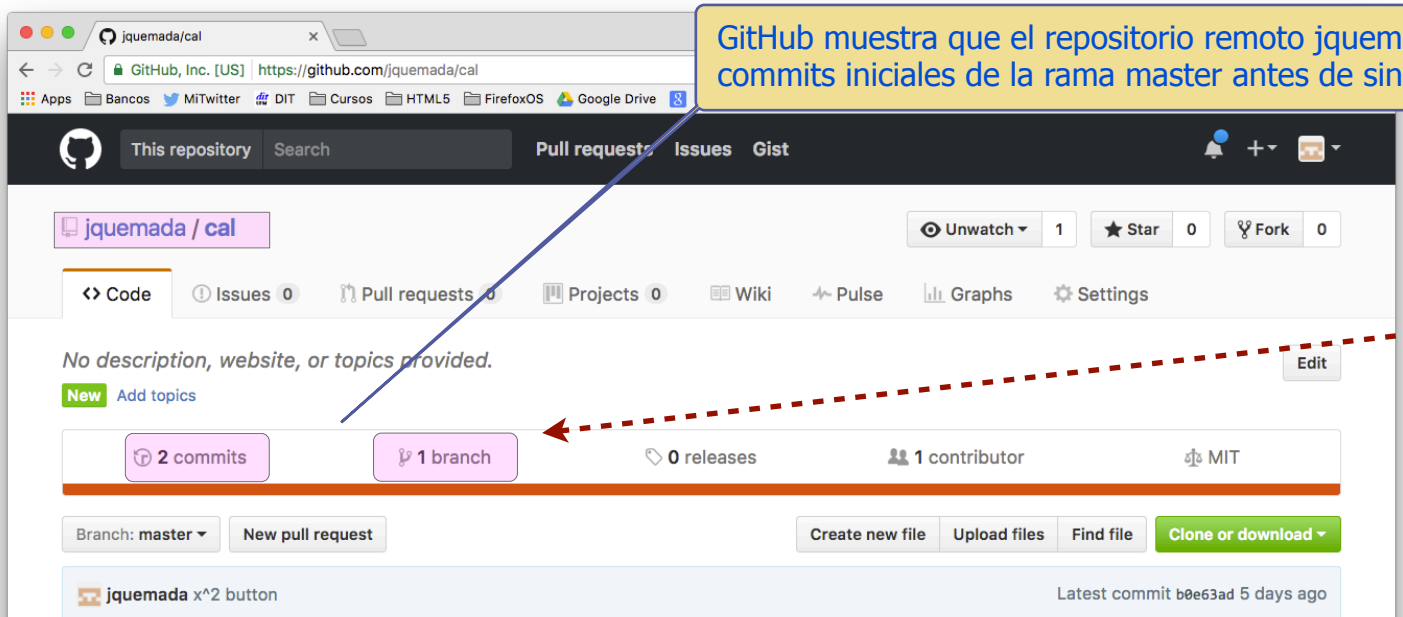
Historia del repositorio local

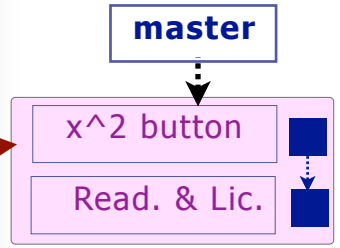
```
venus:cal jq$
venus:cal jq$
venus:cal jq$ git log --oneline --graph --all
* 795c2da 1/x button
| *   1898ac7 Integrate x^2 & 1/x
| |\
| |/
|/|
* | b0e63ad x^2 button
| * e868dc4 1/x button
|/
* 1096247 Readme & License
venus:cal jq$
venus:cal jq$ git push -q --all    https://github.com/jquemada/cal
venus:cal jq$
```

git log --oneline --graph --all muestra el grafo de commits del repositorio local.



GitHub muestra que el repositorio remoto jquemada/cal tiene solo los 2 commits iniciales de la rama master antes de sincronizarlo con el local.

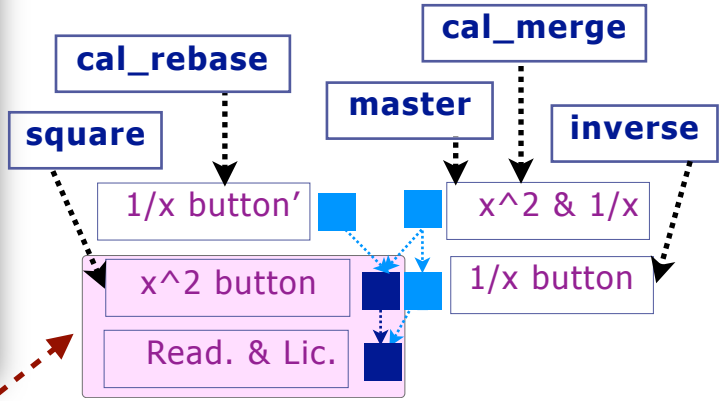


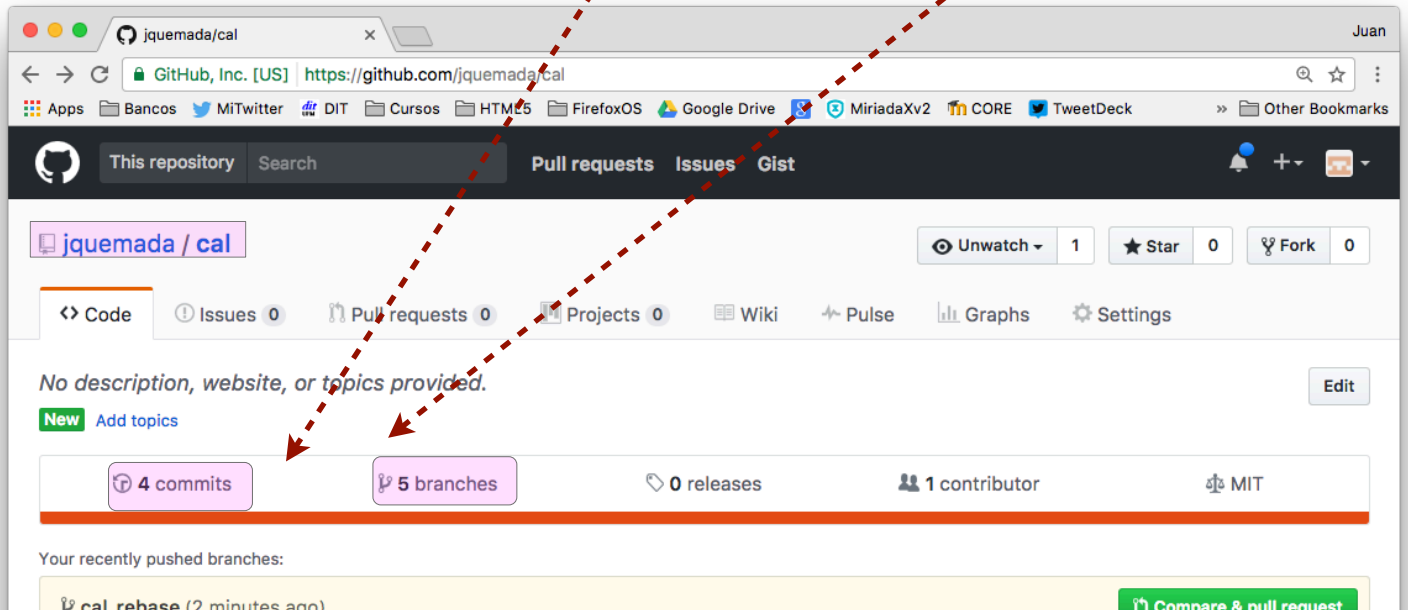


Historia del repositorio local

```
venus:cal jq$
venus:cal jq$
venus:cal jq$ git log --oneline --graph --all
* 795c2da 1/x button
| *   1898ac7 Integrate x^2 & 1/x
| |\
| |/
|/|
* | b0e63ad x^2 button
| * e868dc4 1/x button
|/
* 1096247 Readme & License
venus:cal jq$
venus:cal jq$ git push -q --all   https://github.com/jquemada/cal
venus:cal jq$
```

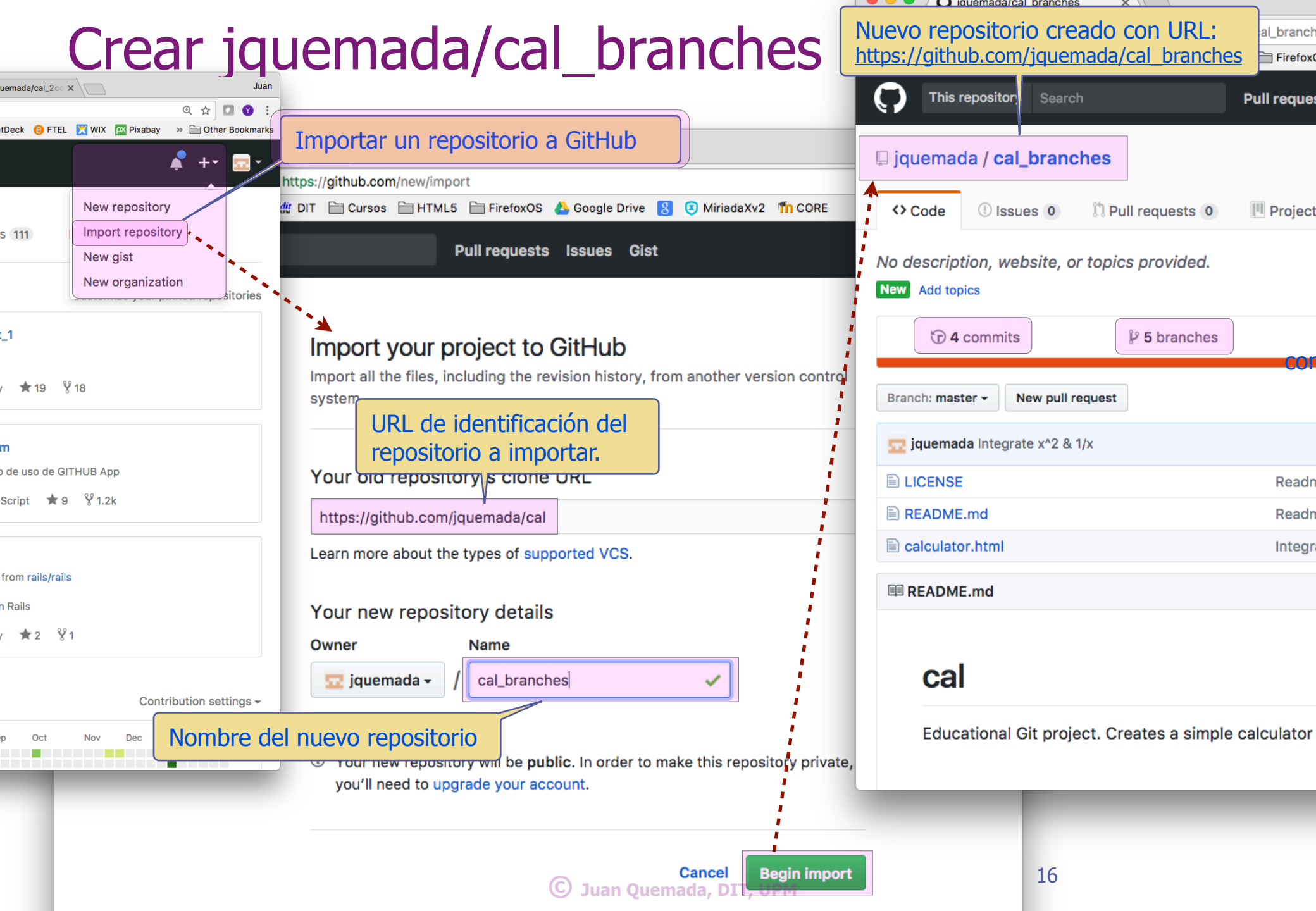
Al sincronizar todas las ramas con push (opcion **--all**) el repositorio remoto **jquemada/cal** actualiza todas las ramas del local.





Crear repositorio jquemada/cal_branches

Crear jquemada/cal_branches



Git y GitHub

Crear commit inicial en GitHub, clonar y actualizar: new_repository, .gitignore, clone, remote y push

Juan Quemada, DIT - UPM

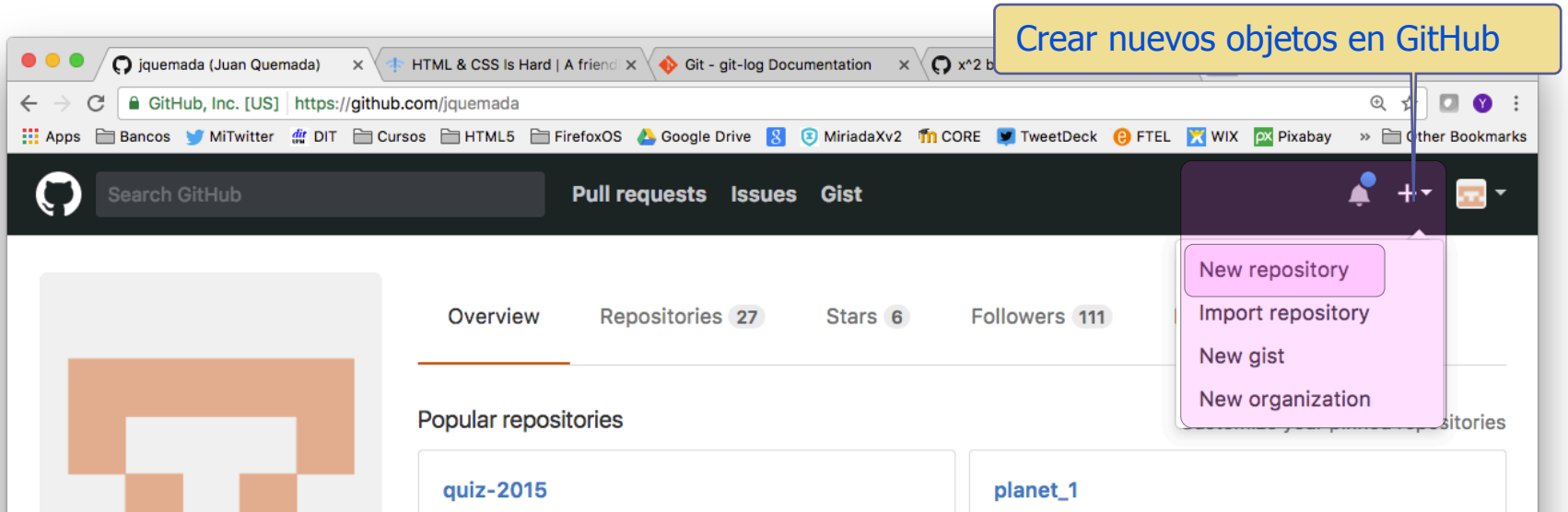
Funciones principales de GitHub

✦Las operaciones principales de un **usuario registrado** son

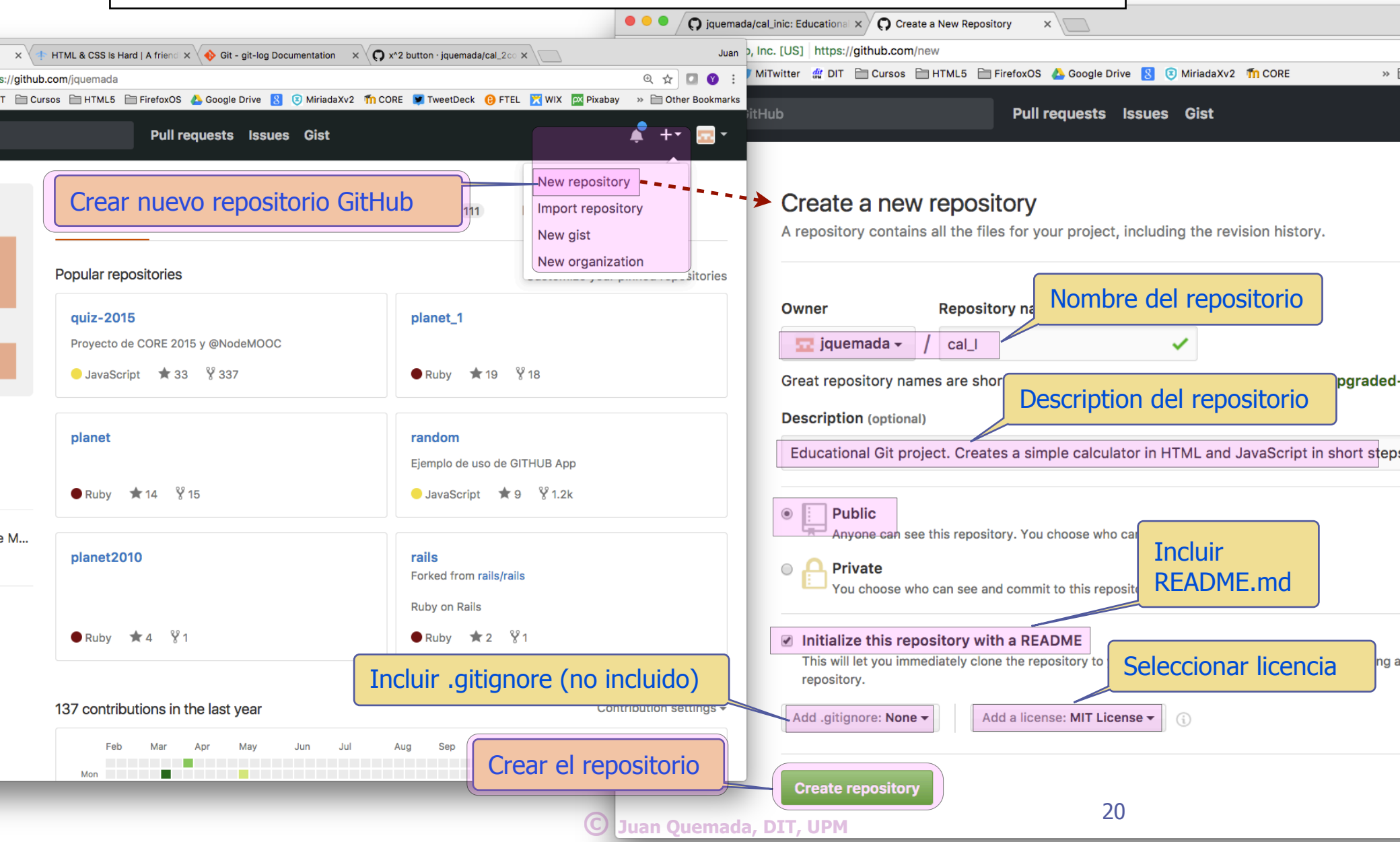
- **Crear repositorio remoto** inicial nuevo para albergar un proyecto
  - Utilizando el botón: **New repository**
    - Puede añadir los ficheros **README.md**, **LICENSE** o **.gitignore** al repositorio creado
- **Copia** un repositorio albergado en GitHub a otra cuenta (para contribuir)
  - Utilizando el botón: **Fork**
- **Importa** un repositorio identificado por su URL a GitHub, incluso en otro formato
  - Utilizando el botón: **Import repository**
    - Equivale a crear repositorio vacío (New_repository) e importar en él otro repositorio con un URL
- **Crear una organización** para albergar múltiples proyectos relacionados
  - Utilizando el botón: **New organisation**
    - Organización de asignatura CORE: https://github.com/CORE-UPM
- Y otras operaciones de compartición, gestión y mantenimiento

Commit inicial en GitHub

- **New repository** permite crear el repositorio inicial de un proyecto
  - Con un commit inicial típico que puede incluir hasta 3 ficheros
    - **README.md** fichero con descripción del proyecto o software
    - **LICENSE** fichero con la licencia de distribución del software
    - **.gitignore** fichero donde se indica que ficheros debe ignorar Git al generar versiones
- Es la forma habitual de arrancar un proyecto albergado en GitHub
  - Se crea el repositorio inicial con 1 commit en: https://github.com/jquemada/cal_I
    - Este repositorio se clona a continuación en uno local con el comando: **git clone …**
      - Se desarrolla un segundo commit en el repositorio local y se sube a GitHub con: **git push …**
  - Esto ilustra el desarrollo 2 commits iniciales similares a los de jquemada/cal
    - Siguiendo el procedimiento habitual de creación de un proyecto



Crear commit inicial en GitHub

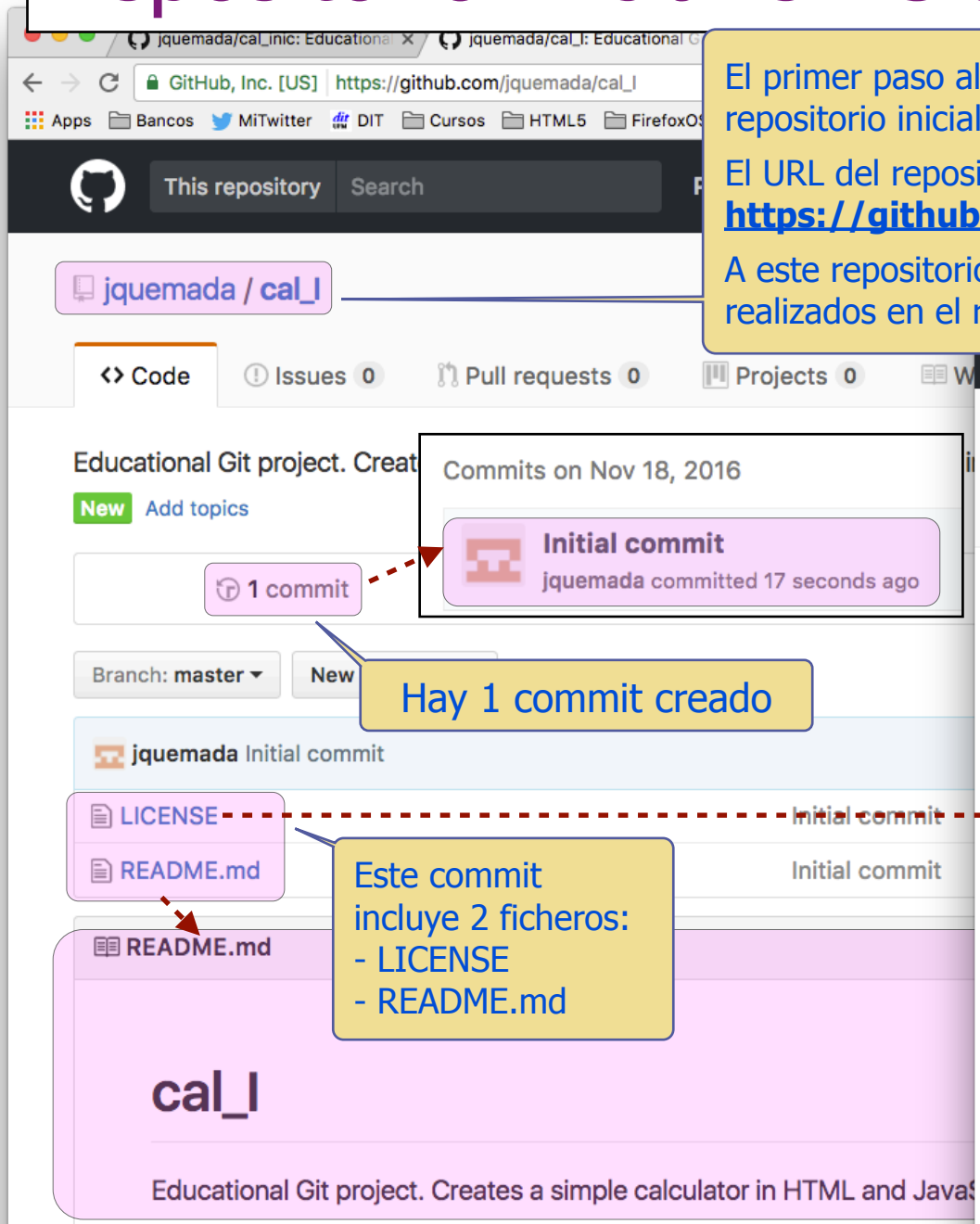


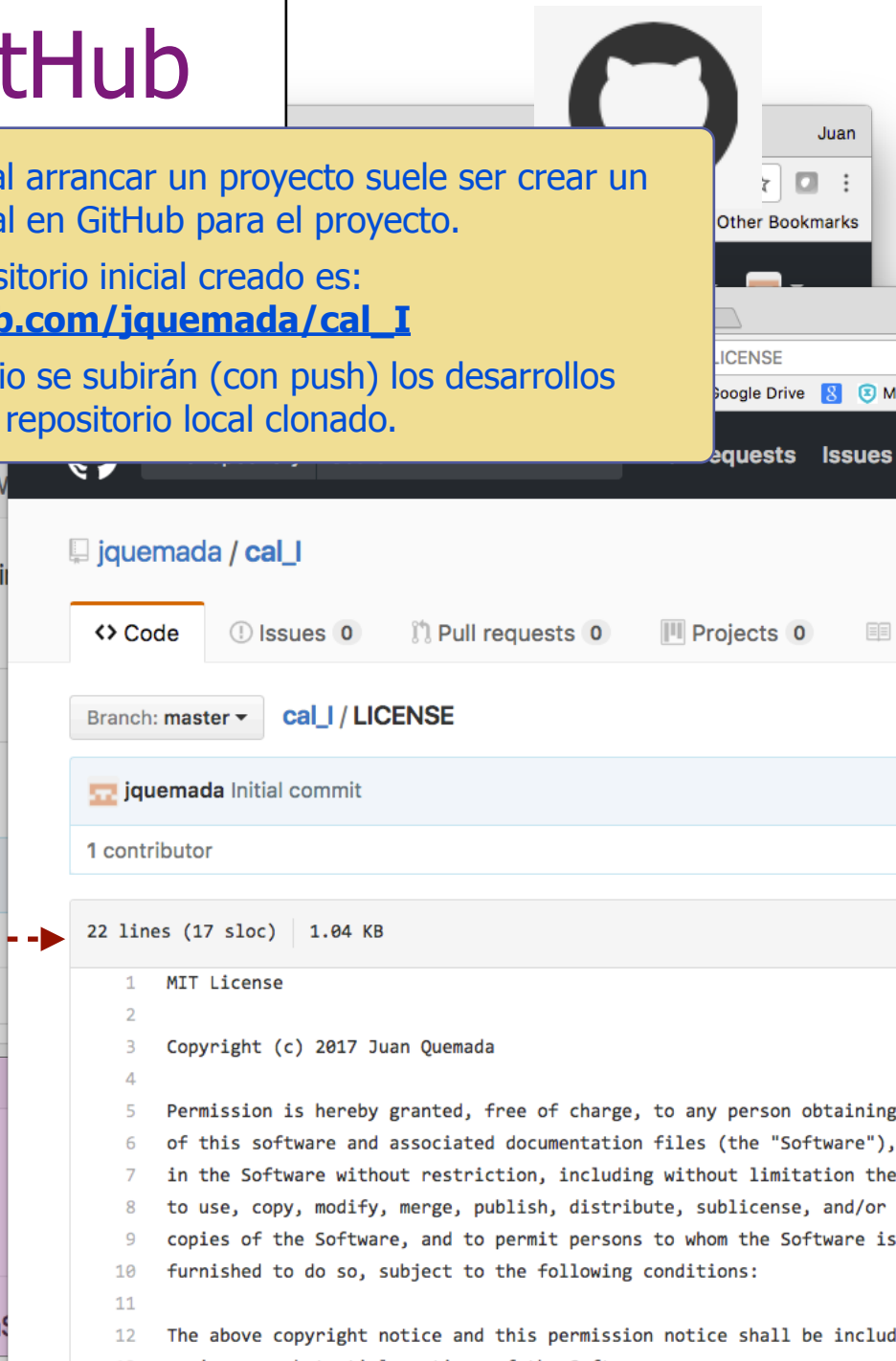
Repositorio inicial en GitHub

El primer paso al arrancar un proyecto suele ser crear un repositorio inicial en GitHub para el proyecto.

El URL del repositorio inicial creado es:
https://github.com/jquemada/cal_I

A este repositorio se subirán (con push) los desarrollos realizados en el repositorio local clonado.





.gitignore

✦**git ls-file --other --ignored --exclude-standard**

- lista todos los ficheros de este proyecto ignorados por Git

```
# .gitignore es un fichero que informa a Git de los ficheros que no debe gestionar.
#  - git status no los presentará como ficheros untracked.
#  - git add . no los añadirá al índice (staging area).

# .gitignore se crea en el directorio o subdirs de trabajo y afecta todo el árbol asociado.

# Su contenido: líneas con patrones de nombres.
#  - Puede usarse los comodines * y ?
#  - Patrones terminados en / indican directorios
#  - Un patrón que empiece con ! indica negación
#  - Se ignoran líneas en blanco y que comiencen con #
#  - [abc] indica cualquiera de los caracteres entre corchetes
#  - [a-z] indica cualquier carácter ASCII (rango desde a hasta z)

# Ejemplo
private.txt     # excluir los ficheros con nombre "private.txt"
*.class         # excluir los ficheros acabados en ".class"
*.[oa]          # excluir ficheros acabados en ".o" y ".a"
!lib.a          # no excluir el fichero "lib.a"
*~              # excluir ficheros acabados en "~"
testing/        # excluir directorio "testing"
```

Clonar un repositorio remoto

Referenciar un repositorio remoto

- Un repositorio remoto se identifica en Internet con un URL
  - Por ejemplo
    - https://github.com/jquemada/cal
    - https://github.com/jquemada/cal_I
  - **git remote …** permite asociar un nombre, denominado **remote**, a un URL
    - El nuevo nombre puede utilizarse, en vez del URL, en los comandos para identificar el repositorio
- **git remote [-v]**
  - Muestra los repositorios remotos definidos en un repositorio (**-v** modo verboso)
- **git remote add …**
  - Define un nuevo remote en el repositorio asociado a un URL
    - **git remote add cal_2com https://github.com/jquemada/cal_2com**
      - asocia el nombre **cal_2com** con el URL **https://github.com/jquemada/cal_2com**
- **git remote remove …**
  - Borra la definición de un remote en el repositorio
    - **git remote remove cal_2com**
      - Borra el nombre **cal_2com** del repositorio y ya no podrá ser utilizado en comandos

Clonar un repositorio remoto

- Un proyecto publicado en un servidor en Internet
  - Puede copiarse al ordenador local con **git clone …** para desarrollar en sus ramas
- **git clone …**
  - Crea un nuevo repositorio local, donde copia la rama master del rep. remoto clonado
    - Además asocia el nombre de remote **origin** al repositorio remoto origen de la clonación
  - Ejemplos de uso
    - **git clone https://github.com/jquemada/cal_I**
      - Copia el repositorio remoto en el directorio local **cal_I** (mismo nombre que el repositorio)
        - **origin** referencia el repositorio clonado: **https://github.com/jquemada/cal_I**
    - **git clone https://github.com/jquemada/cal_I mi_cal**
      - Copia el repositorio remoto en el directorio local **mi_cal**
        - **origin** referencia el repositorio clonado: **https://github.com/jquemada/cal_I**

Clonar repositorio remoto

git clone https://github.com/jquemada/cal_I clona el repositorio identificado por el URL **https://github.com/jquemada/cal**, en el directorio (repositorio) local **cal_I**.

```
venus:proy jq$
venus:proy jq$ git clone https://github.com/jquemada/cal_I
Cloning into 'cal_I'...
remote: Counting objects: 4, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), done.
Checking connectivity... done.
venus:proy jq$
venus:proy jq$ ls
cal     cal_I
venus:proy jq$ cd cal_I
venus:cal_I jq$
venus:cal_I jq$ git log --oneline
5410ac7 Initial commit
venus:cal_I jq$
venus:cal_I jq$ ls
LICENSE         README.md
venus:cal_I jq$
```

Muestra el repositorio clonado **cal_I**, además del ya existente **cal**.

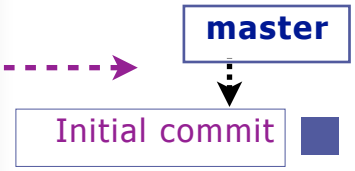
cal_I es un repositorio local diferente de **cal**.

Inspeccionar repositorio clonado

```
venus:proy jq$
venus:proy jq$ git clone https://github.com/jquemada/cal_I
Cloning into 'cal_I'...
remote: Counting objects: 4, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), done.
Checking connectivity... done.
venus:proy jq$
venus:proy jq$ ls
cal     cal_I
venus:proy jq$ cd cal_I
venus:cal_I jq$
venus:cal_I jq$ git log --oneline
5410ac7 Initial commit
venus:cal_I jq$
venus:cal_I jq$ ls
LICENSE         README.md
venus:cal_I jq$
```

cd cal_I entra en el directorio del nuevo repositorio clonado **cal_I**, para que los comandos git se asocien a este repositorio.

git los --oneline muestra que el repositorio contiene el commit generado en automáticamente en GitHub con el mensaje/título "Initial commit".



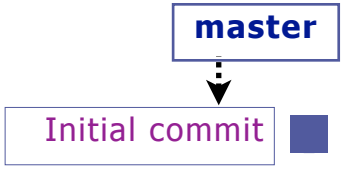
ls muestra que el directorio de trabajo del repositorio contiene los ficheros **README.md** y **LICENSE**.

Generar nuevo commit en el repositorio clonado

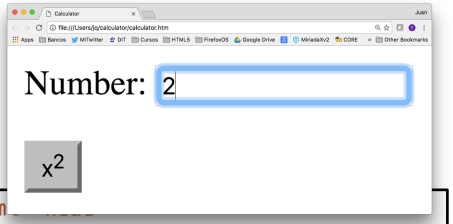
Directorio de trabajo limpio

```
cal_I — -bash — 53×19
venus:cal_I jq$
venus:cal_I jq$ git status -s
venus:cal_I jq$
venus:cal_I jq$ # Edit/create calculator.html
venus:cal_I jq$
venus:cal_I jq$ ls
LICENSE         README.md       calculator.html
venus:cal_I jq$
venus:cal_I jq$ git status -s
?? calculator.html
venus:cal_I jq$
venus:cal_I jq$ git add calculator.html
venus:cal_I jq$
venus:cal_I jq$ git commit -q   -m "x^2 button"
venus:cal_I jq$
venus:cal_I jq$ git log --oneline
9cba3e6 x^2 button
5410ac7 Initial commit
venus:cal_I jq$
```

git status -s muestra un directorio de trabajo limpio, sin cambios respecto al último commit.



Añadir fichero calculator.html



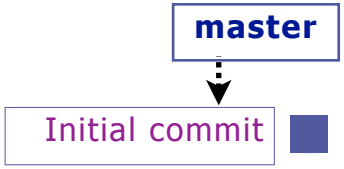
Se añade el fichero **calculator.html** al directorio de trabajo con un editor, copiando (cp), ..

```
venus:cal_I jq$
venus:cal_I jq$ git status -s
venus:cal_I jq$
venus:cal_I jq$ # Edit/create calculator.html
venus:cal_I jq$
venus:cal_I jq$ ls
LICENSE         README.md       calculator.html
venus:cal_I jq$
venus:cal_I jq$ git status -s
?? calculator.html
venus:cal_I jq$
venus:cal_I jq$ git add calculator.html
venus:cal_I jq$
venus:cal_I jq$ git commit -q   -m "x^2 button"
venus:cal_I jq$
venus:cal_I jq$ git log --oneline
9cba3e6 x^2 button
5410ac7 Initial commit
venus:cal_I jq$
```

calculator.html no está registrada.

```
<!DOCTYPE html><htm
<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>
```

calculator.html tiene la misma calculadora con el botón x^2, que se generó en el repositorio cal.

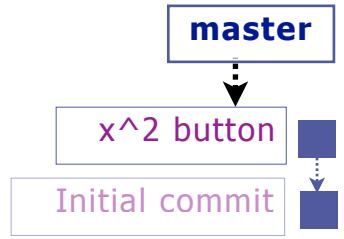


Crear nuevo commit

```
venus:cal_I jq$
venus:cal_I jq$ git status -s
venus:cal_I jq$
venus:cal_I jq$ # Edit/create calculator.html
venus:cal_I jq$
venus:cal_I jq$ ls
LICENSE         README.md       calculator.html
venus:cal_I jq$
venus:cal_I jq$ git status -s
?? calculator.html
venus:cal_I jq$
venus:cal_I jq$ git add calculator.html
venus:cal_I jq$
venus:cal_I jq$ git commit -q   -m "x^2 button"
venus:cal_I jq$
venus:cal_I jq$ git log --oneline
9cba3e6 x^2 button
5410ac7 Initial commit
venus:cal_I jq$
```

Registra fichero calculator.html en el índice.

Generar nuevo commit con mensaje/título **"x^2 button"**
Opción -q: modo quiet (sin mensajes, ni estadísticas).



Muestra los dos commits ya generados en la rama master (formato 1 línea).

Actualizar el repositorio origin

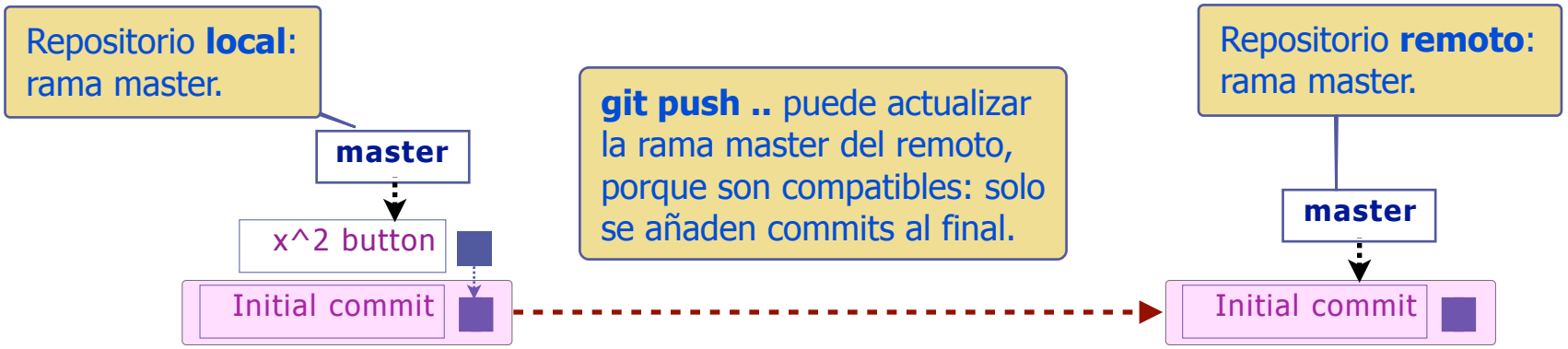
Actualizar un repositorio en GitHub con push

✦ git push …

- Actualiza el repositorio remoto con los nuevos commits de una rama local
  - **git push origin master**
    - actualiza master en el repositorio remoto **origin** (https://github.com/jquemada/cal)
  - **git push https://github.com/jquemada/cal master**
    - Es equivalente al comando: **git push origin master**

✦ git push … necesita 2 condiciones para finalizar con éxito

- Se debe tener credenciales de acceso al repositorio remoto
  - Por ejemplo, un repositorio en una cuenta u organización del usuario que lo actualiza
- La actualización de commits debe ser compatible con la rama actualizada en el remoto
  - Solo debe **añadir nuevos commits al final de la rama remota** o actualizar un **repositorio vacío**
    - **Peligroso!** La **opción -f** permite actualizar una rama incompatible, pero se pierden commits

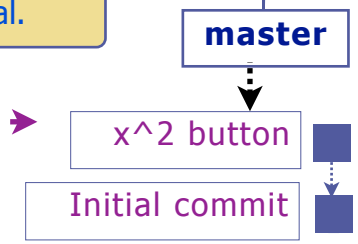


Mostrar historia y remote origin

```
venus:cal_I jq$
venus:cal_I jq$ git log --oneline
9cba3e6 x^2 button
5410ac7 Initial commit
venus:cal_I jq$
venus:cal_I jq$ git remote
origin
venus:cal_I jq$ git remote -v
origin  https://github.com/jquemada/cal_I (fetch)
origin  https://github.com/jquemada/cal_I (push)
venus:cal_I jq$
venus:cal_I jq$ git push -q   origin   master
venus:cal_I jq$
```

Muestra los dos commits del repositorio local.

Rama master del repositorio **local**.



git remote muestra el remote **origin** definido por **git clone ..**

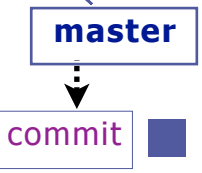
Mostrar URL asociado a origin

```
venus:cal_I jq$
venus:cal_I jq$ git log --oneline
9cba3e6 x^2 button
5410ac7 Initial commit
venus:cal_I jq$
venus:cal_I jq$ git remote
origin
venus:cal_I jq$ git remote -v
origin  https://github.com/jquemada/cal_I (fetch)
origin  https://github.com/jquemada/cal_I (push)
venus:cal_I jq$
venus:cal_I jq$ git push -q   origin   master
venus:cal_I jq$
```

git remote -v (verboso) muestra también el **URL** asociado al remote **origin** definido por **git clone ..**:

https://github.com/jquemada/cal_I

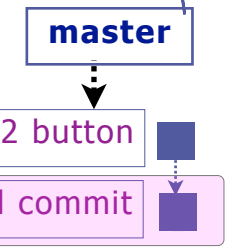
Rama master del repositorio **remoto**.



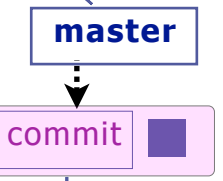
Actualizar rama master remota

```
venus:cal_I jq$
venus:cal_I jq$ git log --oneline
9cba3e6 x^2 button
5410ac7 Initial commit
venus:cal_I jq$
venus:cal_I jq$ git remote
origin
venus:cal_I jq$ git remote -v
origin  https://github.com/jquemada/cal_I (fetch)
origin  https://github.com/jquemada/cal_I (push)
venus:cal_I jq$
venus:cal_I jq$ git push -q origin master
venus:cal_I jq$
```

Rama master del repositorio **local**.

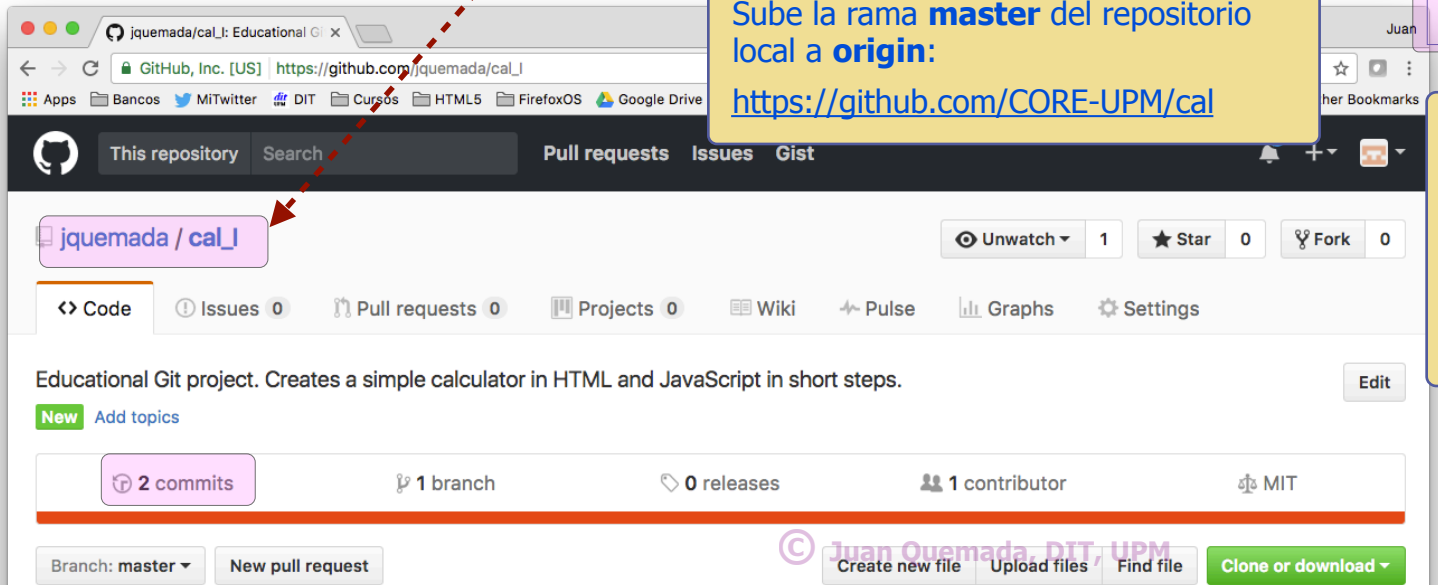


Rama master del repositorio **remoto**.



Sube la rama **master** del repositorio local a **origin**:

https://github.com/CORE-UPM/cal



La actualización solo añade nuevos commits al final de la rama master ya guardada en el repositorio remoto **origin:** es **compatible!**

Final del tema