



# Gestión de proyectos con Git y GitHub: Modulo 6

## Temas:

1. Clonar con Fork en GitHub: Fork, clone y push ..... 2
2. Ramas locales, remotas, tracking y refsspecs: branch, checkout, clone, fetch y pull ..... 31

Juan Quemada, DIT - UPM



# Git y GitHub

Clonar con Fork en GitHub: `fork_repository`, clone y push

Juan Quemada, DIT - UPM

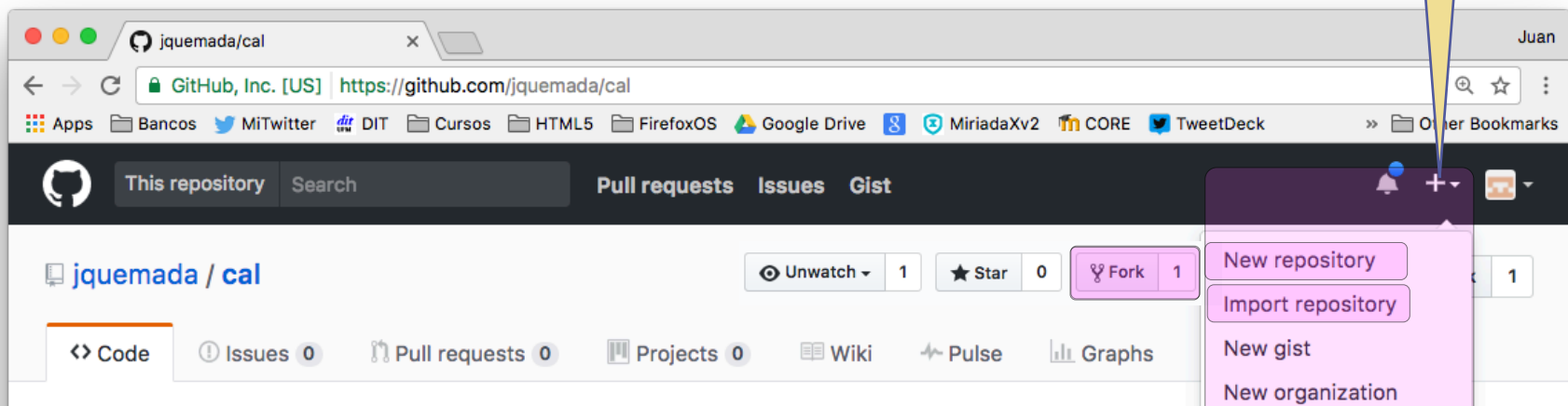
# Funciones principales de GitHub



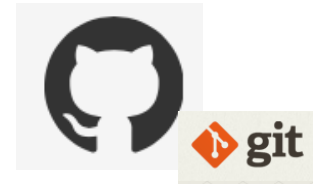
◆ Las operaciones principales de un **usuario registrado** son

- **Crear repositorio remoto** nuevo y vacío para albergar un proyecto
  - Utilizando el botón: **New repository**
- **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**
    - Se puede importar de otro servidor en Internet o de GitHub, incluso cambiando el formato
- **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

Crear nuevos  
objetos en  
GitHub



# Fork de un repositorio en GitHub



- ◆ Un equipo suele tener un **repositorio** remoto **de referencia**
  - Donde se consolidan las últimas versiones de un proyecto
    - Es la referencia que utilizan todos los miembros del equipo y sus usuarios
  - El repositorio remoto suele administrarse por unas pocas personas
- ◆ Otros desarrolladores lo copian con **Fork** a su cuenta en GitHub
  - Los desarrollos se realizan en la copia en GitHub clonada con Fork
    - GitHub permite contribuir los desarrollos al repositorio de referencia con **pull request**
- ◆ El proyecto **jquemada/cal** se clonó con **Fork** a **CORE-UPM/cal**
  - Cuando el repositorio tenía solo los 2 primeros commits de la calculadora
    - **CORE-UPM/cal** no ha evolucionado y sigue teniendo solo los **2 commits iniciales**
      - En cambio, **jquemada/cal** ha evolucionando desde que se clonó con nuevas ramas y commits
- ◆ **CORE-UPM/cal** debe ser clonado en local para realizar desarrollos
  - Los desarrollos se añaden como nuevos commits en el repositorio local
    - Una vez finalizados se pueden subir a CORE-UPM/cal y contribuir a jquemada/cal con **pull\_request**

# Fork: clonar un proyecto de GitHub



Repositorio cal del usuario jquemada en:  
<https://github.com/jquemada/cal>

Copia el repositorio a otra cuenta u organización del usuario en GitHub pulsando el **botón Fork**.

Where should we fork this repository?



@ging



@CORE-UPM...

Can't find what you're looking for?

You already have a fork of this repository:

[jquemada/cal\\_2com](#)

Copia de **jquemada/cal** en **CORE-UPM**:  
<https://github.com/CORE-UPM/cal>

En el momento del **Fork** el repositorio **jquemada/cal** tiene estos **2 commits**. A partir de este momento cada repositorio evolucionara por separado a partir de estos 2 commits.

Commits on Feb 12, 2017



x^2 button

jquemada committed 3 hours ago

Commits on Feb 9, 2017



Readme & License

jquemada committed 3 days ago

2 commits

# Clonar un repositorio remoto

# Pasar a directorio de proyectos

Estando en el directorio **cal** (repositorio) pasamos al directorio **proy**, donde están los repositorios de proyectos para que el nuevo repositorio se clone en el directorio adecuado.

```
venus:cal jq$  
venus:cal jq$ cd ..  
venus:proy jq$  
venus:proy jq$ git clone https://github.com/CORE-UPM/cal cal_2  
Cloning into 'cal_2'...  
remote: Counting objects: 10, done.  
remote: Compressing objects: 100% (7/7), done.  
remote: Total 10 (delta 3), reused 10 (delta 3), pack-reused 0  
Unpacking objects: 100% (10/10), done.  
Checking connectivity... done.  
venus:proy jq$  
venus:proy jq$ cd cal_2  
venus:cal_2 jq$  
venus:cal_2 jq$ git log --oneline  
b0e63ad x^2 button  
1096247 Readme & License  
venus:cal_2 jq$  
venus:cal_2 jq$ git status -s  
venus:cal_2 jq$  
venus:cal_2 jq$ ls  
LICENSE      README.md      calculator.html  
venus:cal_2 jq$
```

# Clonar repositorio CORE-UPM/cal de GitHub

**git clone https://github.com/jquemada/cal cal\_2**  
copia el repositorio remoto en GitHub, identificado por el URL <https://github.com/jquemada/cal>, en el directorio (repositorio) local **cal\_2**.

```
venus:cal jq$  
venus:cal jq$ cd ..  
venus:proy jq$  
venus:proy jq$ git clone https://github.com/CORE-UPM/cal cal_2  
Cloning into 'cal_2'...  
remote: Counting objects: 10, done.  
remote: Compressing objects: 100% (7/7), done.  
remote: Total 10 (delta 3), reused 10 (delta 3), pack-reused 0  
Unpacking objects: 100% (10/10), done.  
Checking connectivity... done.  
venus:proy jq$  
venus:proy jq$ cd cal_2  
venus:cal_2 jq$  
venus:cal_2 jq$ git log --oneline  
b0e63ad x^2 button  
1096247 Readme & License  
venus:cal_2 jq$  
venus:cal_2 jq$ git status -s  
venus:cal_2 jq$  
venus:cal_2 jq$ ls  
LICENSE      README.md      calculator.html  
venus:cal_2 jq$
```



# Entrar en el nuevo repositorio clonado

```
venus:cal jq$  
venus:cal jq$ cd ..  
venus:proy jq$  
venus:proy jq$ git clone https://github.com/CORE-UPM/cal cal_2  
Cloning into 'cal_2'...  
remote: Counting objects: 10, done.  
remote: Compressing objects: 100% (7/7), done.  
remote: Total 10 (delta 3), reused 10 (delta 3), pack-reused 0  
Unpacking objects: 100% (10/10), done.  
Checking connectivity... done.  
venus:proy jq$  
venus:proy jq$ cd cal_2  
venus:cal_2 jq$  
venus:cal_2 jq$ git log --oneline  
b0e63ad x^2 button  
1096247 Readme & License  
venus:cal_2 jq$  
venus:cal_2 jq$ git status -s  
venus:cal_2 jq$  
venus:cal_2 jq$ ls  
LICENSE      README.md      calculator.html  
venus:cal_2 jq$
```

Se cambia el directorio de trabajo del terminal de comandos para que coincida con el directorio de trabajo Git.

# Ver historia del repositorio clonado

```
venus:cal jq$  
venus:cal jq$ cd ..  
venus:proy jq$  
venus:proy jq$ git clone https://github.com/CORE-UPM/cal cal_2  
Cloning into 'cal_2'...  
remote: Counting objects: 10, done.  
remote: Compressing objects: 100% (7/7), done.  
remote: Total 10 (delta 3), reused 10 (delta 3), pack-reused 0  
Unpacking objects: 100% (10/10), done.  
Checking connectivity... done.  
venus:proy jq$  
venus:proy jq$ cd cal_2  
venus:cal_2 jq$  
venus:cal_2 jq$ git log --online  
b0e63ad x^2 button  
1096247 Readme & License  
venus:cal_2 jq$  
venus:cal_2 jq$ git status -s  
venus:cal_2 jq$ ls  
LICENSE      README.md      calculator.html  
venus:cal_2 jq$
```

Muestra los 2 commits del repositorio remoto clonado.

master

x^2 button

Readme & License

# Mostrar estado del directorio de trabajo

```
venus:cal jq$  
venus:cal jq$ cd ..  
venus:proy jq$  
venus:proy jq$ git clone https://github.com/CORE-UPM/cal cal_2  
Cloning into 'cal_2'...  
remote: Counting objects: 10, done.  
remote: Compressing objects: 100% (7/7), done.  
remote: Total 10 (delta 3), reused 10 (delta 3), pack-reused 0  
Unpacking objects: 100% (10/10), done.  
Checking connectivity... done.  
venus:proy jq$  
venus:proy jq$ cd cal_2  
venus:cal_2 jq$  
venus:cal_2 jq$ git log --oneline  
b0e63ad x^2 button  
1096247 Readme & License  
venus:cal_2 jq$  
venus:cal_2 jq$ git status -s  
venus:cal_2 jq$  
venus:cal_2 jq$ ls  
LICENSE      README.md      calculator.html  
venus:cal_2 jq$
```

Muestra un directorio de trabajo limpio, sin cambios respecto al último commit.

master

x^2 button

Readme & License

# Mostrar ficheros del directorio de trabajo

```
venus:cal jq$  
venus:cal jq$ cd ..  
venus:proy jq$  
venus:proy jq$ git clone https://github.com/CORE-UPM/cal cal_2  
Cloning into 'cal_2'...  
remote: Counting objects: 10, done.  
remote: Compressing objects: 100% (7/7), done.  
remote: Total 10 (delta 3), reused 10 (delta 3), pack-reused 0  
Unpacking objects: 100% (10/10), done.  
Checking connectivity... done.  
venus:proy jq$  
venus:proy jq$ cd cal_2  
venus:cal_2 jq$  
venus:cal_2 jq$ git log --oneline  
b0e63ad x^2 button  
1096247 Readme & License  
venus:cal_2 jq$  
venus:cal_2 jq$ git status -s  
venus:cal_2 jq$ ls  
LICENSE      README.md    calculator.html  
venus:cal_2 jq$
```

**git clone ..** ha restaurado los 3 ficheros del último commit en el directorio de trabajo.

master

x^2 button

Readme & License

Generar nuevo commit en el repositorio clonado

# Ficheros del directorio de trabajo

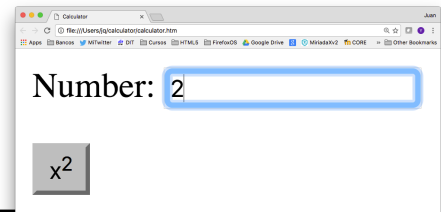
```
venus:cal_2 jq$  
venus:cal_2 jq$ ls  
LICENSE      README.md  
venus:cal_2 jq$  
venus:cal_2 jq$ # Edit calculator.html  
venus:cal_2 jq$  
venus:cal_2 jq$ git status -s  
M calculator.html  
venus:cal_2 jq$  
venus:cal_2 jq$ git diff  
diff --git a/calculator.html b/calculator.html  
index 839b37e..db62028 100644  
--- a/calculator.html  
+++ b/calculator.html  
@@ -8,6 +8,9 @@ function square() {  
  }  
  </script>  
  </head>  
+  
+<h3>My Calculator</h3>  
+  
  <body>  
    Number:  
    <input type="text" id="n1"><p>  
venus:cal_2 jq$  
venus:cal_2 jq$ git commit -a -q -m "Añadir título"  
venus:cal_2 jq$  
venus:cal_2 jq$ git log --oneline  
5c70f9c Añadir título  
b0e63ad x^2 button  
1096247 Readme & License  
venus:cal_2 jq$
```

ls muestra los ficheros del directorio de trabajo.

calculator.html

calculator.html tiene la calculadora generada en el commit anterior.

```
<!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>
```



# Añadir título a calculadora.html

```
venus:cal_2 jq$
venus:cal_2 jq$ ls
LICENSE      README.md    calculator.html
venus:cal_2 jq$
venus:cal_2 jq$ # Edit calculator.html
venus:cal_2 jq$
venus:cal_2 jq$ git status -s
  M calculator.html
venus:cal_2 jq$
venus:cal_2 jq$ git diff
diff --git a/calculator.html b/calculator.html
index 839b37e..db62028 100644
--- a/calculator.html
+++ b/calculator.html
@@ -8,6 +8,9 @@ function square() {
 }
</script>
</head>
+
+<h3>My Calculator</h3>
+
<body>
  Number:
  <input type="text" id="n1"><p>
venus:cal_2 jq$
venus:cal_2 jq$ git commit -a -q -m "Añadir título"
venus:cal_2 jq$
venus:cal_2 jq$ git log --oneline
5c70f9c Añadir título
b0e63ad x^2 button
1096247 Readme & License
venus:cal_2 jq$
```

calculator.html

calculator.html

Se edita el fichero y se  
añade el título HTML  
(**<h3>My Calculator  
</h3>**) a la calculadora  
(resaltado en verde).

## My Calculator

Number:

x<sup>2</sup>

```
<!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>
<h3>My Calculator</h3>
<body>
  Number:
  <input type="text" id="n1"><p>

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

# Inspeccionar estado

```
venus:cal_2 jq$  
venus:cal_2 jq$ ls  
LICENSE      README.md    calculator.html  
venus:cal_2 jq$  
venus:cal_2 jq$ # Edit calculator.html  
venus:cal_2 jq$  
venus:cal_2 jq$ git status -s  
M calculator.html  
venus:cal_2 jq$  
venus:cal_2 jq$ git diff  
diff --git a/calculator.html b/calculator.html  
index 839b37e..db62028 100644  
--- a/calculator.html  
+++ b/calculator.html  
@@ -8,6 +8,9 @@ function square() {  
  }  
  </script>  
  </head>  
+  
+<h3>My Calculator</h3>  
+  
  <body>  
    Number:  
    <input type="text" id="n1"><p>  
venus:cal_2 jq$  
venus:cal_2 jq$ git commit -a -q -m "Añadir título"  
venus:cal_2 jq$  
venus:cal_2 jq$ git log --oneline  
5c70f9c Añadir título  
b0e63ad x^2 button  
1096247 Readme & License  
venus:cal_2 jq$
```

**calculator.html** está en estado modified, no se ha registrado todavía en el índice.

## My Calculator

Number:

$x^2$

```
<!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>  
  
<h3>My Calculator</h3>  
  
<body>  
  Number:  
  <input type="text" id="n1"><p>  
  
  <button onclick="square()"> x<sup>2</sup> </button>  
</body>  
</html>
```



# Inspeccionar diferencias

```
venus:cal_2 jq$
venus:cal_2 jq$ ls
LICENSE      README.md    calculator
venus:cal_2 jq$
venus:cal_2 jq$ # Edit calculator.html
venus:cal_2 jq$
venus:cal_2 jq$ git status -s
  M calculator.html
venus:cal_2 jq$
venus:cal_2 jq$ git diff
diff --git a/calculator.html b/calculator.html
index 839b37e..db62028 100644
--- a/calculator.html
+++ b/calculator.html
@@ -8,6 +8,9 @@ function square() {
 }
</script>
</head>
+
+<h3>My Calculator</h3>
+
<body>
  Number:
  <input type="text" id="n1"><p>
venus:cal_2 jq$
venus:cal_2 jq$ git commit -a -q -m "Añadir título"
venus:cal_2 jq$
venus:cal_2 jq$ git log --oneline
5c70f9c Añadir título
b0e63ad x^2 button
1096247 Readme & License
venus:cal_2 jq$
```

**git diff** muestra las diferencias en los ficheros **modified** del directorio de trabajo. Solo se ha modificado **calculator.html**.

git diff

```
<!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>

<h3>My Calculator</h3>

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

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

## My Calculator

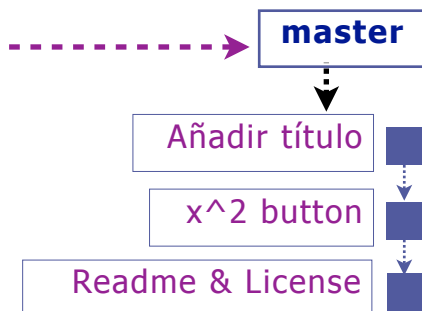
Number: 45

x<sup>2</sup>

# Cerrar commit

```
venus:cal_2 jq$
venus:cal_2 jq$ ls
LICENSE      README.md    calculator.html
venus:cal_2 jq$
venus:cal_2 jq$ # Edit calculator.html
venus:cal_2 jq$
venus:cal_2 jq$ git status -s
  M calculator.html
venus:cal_2 jq$
venus:cal_2 jq$ git diff
diff --git a/calculator.html b/calculator.html
index 839b37e..db62028 100644
--- a/calculator.html
+++ b/calculator.html
@@ -8,6 +8,9 @@ function square() {
 }
</script>
</head>
+
+<h3>My Calculator</h3>
+
<body>
  Number:
  <input type="text" id="n1"><p>
venus:cal_2 jq$ git commit -a -q -m "Añadir título"
venus:cal_2 jq$ git log --oneline
5c70f9c Añadir título
b0e63ad x^2 button
1096247 Readme & License
venus:cal_2 jq$
```

Generar nuevo commit con mensaje/título **"Añadir título"**.  
**Opción -a**: añadir al índice todos los cambios antes del commit.  
**Opción -q**: modo quiet (sin mensajes, ni estadísticas).



# Mostrar historia de master

```
venus:cal_2 jq$  
venus:cal_2 jq$ ls  
LICENSE      README.md      calculator.html  
venus:cal_2 jq$  
venus:cal_2 jq$ # Edit calculator.html  
venus:cal_2 jq$  
venus:cal_2 jq$ git status -s  
  M calculator.html  
venus:cal_2 jq$  
venus:cal_2 jq$ git diff  
diff --git a/calculator.html b/calculator.html  
index 839b37e..db62028 100644  
--- a/calculator.html  
+++ b/calculator.html  
@@ -8,6 +8,9 @@ function square() {  
  }  
  </script>  
  </head>  
+  
+<h3>My Calculator</h3>  
+  
  <body>  
    Number:  
    <input type="text" id="n1"><p>  
venus:cal_2 jq$  
venus:cal_2 jq$ git commit -a -q -m "Añadir título"  
venus:cal_2 jq$  
venus:cal_2 jq$ git log --oneline  
5c70f9c Añadir título  
b0e63ad x^2 button  
1096247 Readme & License  
venus:cal_2 jq$
```

Muestra los tres commits (formato 1 línea).

master

Añadir título

x^2 button

Readme & License

Actualizar el repositorio origin

# Mostrar historia

```
venus:cal_2 jq$  
venus:cal_2 jq$ git log --online  
5c70f9c Añadir título  
b0e63ad x^2 button  
1096247 Readme & License  
venus:cal_2 jq$  
venus:cal_2 jq$ git remote  
origin  
venus:cal_2 jq$ git remote -v  
origin https://github.com/CORE-UPM/cal (fetch)  
origin https://github.com/CORE-UPM/cal (push)  
venus:cal_2 jq$  
venus:cal_2 jq$ git push -q origin master  
venus:cal_2 jq$
```

Muestra los tres commits (formato 1 línea).

Rama master del repositorio **local**.

**master**

Añadir título

x^2 button

Readme & License

# Mostrar remote origin

```
venus:cal_2 jq$  
venus:cal_2 jq$ git log --oneline  
5c70f9c Añadir título  
b0e63ad x^2 button  
1096247 Readme & License  
venus:cal_2 jq$  
venus:cal_2 jq$ git remote  
origin  
venus:cal_2 jq$ git remote -v  
origin https://github.com/CORE-UPM/cal (fetch)  
origin https://github.com/CORE-UPM/cal (push)  
venus:cal_2 jq$  
venus:cal_2 jq$ git push -q origin master  
venus:cal_2 jq$
```

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

# Mostrar URL asociado a origen

```
venus:cal_2 jq$  
venus:cal_2 jq$ git log --oneline  
5c70f9c Añadir título  
b0e63ad x^2 button  
1096247 Readme & License  
venus:cal_2 jq$  
venus:cal_2 jq$ git remote  
origin  
venus:cal_2 jq$ git remote -v  
origin https://github.com/CORE-UPM/cal (fetch)  
origin https://github.com/CORE-UPM/cal (push)  
venus:cal_2 jq$  
venus:cal_2 jq$ git push -q origin master  
venus:cal_2 jq$
```

**git remote -v** (verboso) muestra también el **URL** asociado al remote **origin** definido por **git clone ..**:  
<https://github.com/jquemada/cal>

Rama master del repositorio **remoto**.

**master**

x^2 button

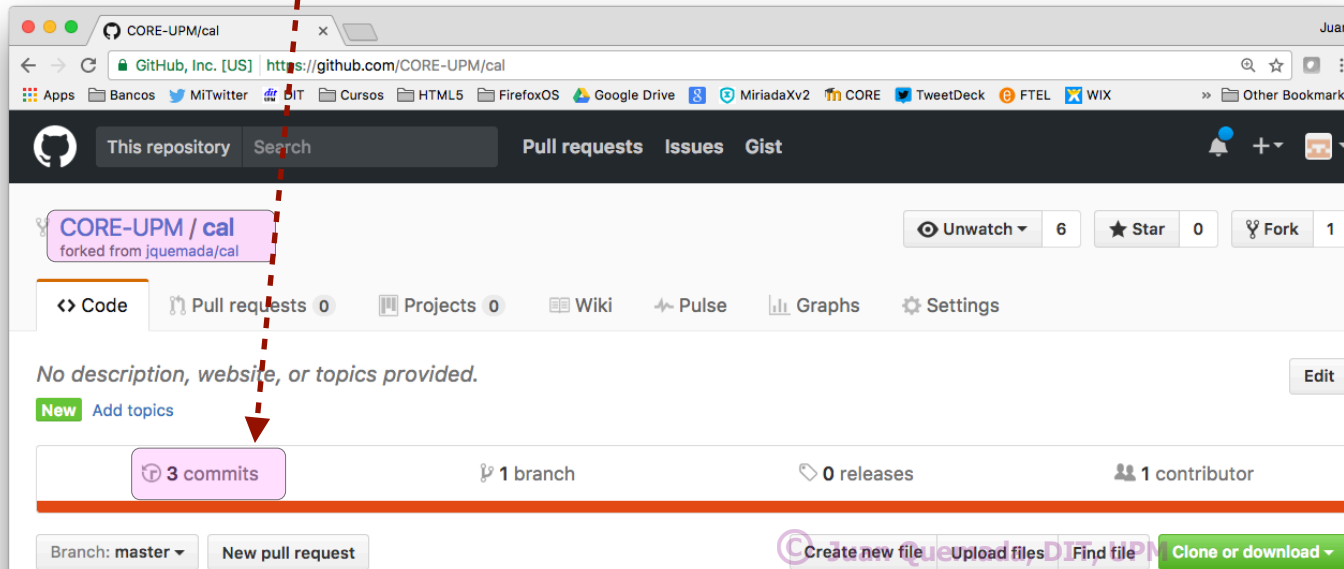
Readme & License

# Sincronizar origin con push

```
venus:cal_2 jq$  
venus:cal_2 jq$ git log --oneline  
5c70f9c Añadir título  
b0e63ad x^2 button  
1096247 Readme & License  
venus:cal_2 jq$  
venus:cal_2 jq$ git remote  
origin  
venus:cal_2 jq$ git remote -v  
origin https://github.com/CORE-UPM/cal (fetch)  
origin https://github.com/CORE-UPM/cal (push)  
venus:cal_2 jq$  
venus:cal_2 jq$ git push -q origin master  
venus:cal_2 jq$
```

El remote **origin** es equivalente a utilizar el URL:  
<https://github.com/jquemada/cal>

Sube la rama **master** del repositorio local al repositorio **origin**:  
<https://github.com/CORE-UPM/cal>



Rama master del repositorio **local**.

**master**

Añadir título

x^2 button

Readme & License

Rama master del repositorio **remoto**.

**master**

x^2 button

Readme & License

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



Crear repositorio CORE-UPM/cal\_3com

# Crear CORE-UPM/cal\_3com

Importar un repositorio a GitHub

<https://github.com/new/import>

Import your project to GitHub

Import all the files, including the revision history, from another version control system.

Your old repository's clone URL

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

Learn more about the types of [supported VCS](#).

Your new repository details

Owner	Name
CORE-UPM	cal_3com

be public. In order to make this repository private you'll need to [upgrade your account](#).

Nuevo repositorio creado con URL:  
[https://github.com/jquemada/cal\\_2com](https://github.com/jquemada/cal_2com)

This repository Search Pull requests

CORE-UPM / cal\_3com

<> Code 0 Issues 0 Pull requests 0 Projects 0

No description, website, or topics provided.

New Add topics

3 commits 1 branch

Branch: master New pull request

jquemada Añadir título

LICENSE	Readme &
README.md	Readme &
calculator.html	Añadir título

README.md

## cal

Educational Git project. Creates a simple calculator in H

Cancel

Begin import



# Final del tema

# Cuestionario

◆ Indicar que respuesta define correctamente el siguiente término:

## **git push origin master**

- => 1. Botón de los repositorios de GitHub que permite clonar ese repositorio en otra cuenta en GitHub  
2. Comando que actualiza la rama master del repositorio remoto origen del local con las últimas versiones  
3. Identificador simbólico asociado al URL del repositorio remoto del que se ha clonado el repositorio local  
4. Comando que clona un repositorio remoto identificado por su URL  
5. Comando que muestra los repositorios remotos definidos en el local, junto con el URL y función asociados

## **Botón**

### **Botón de Fork**

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

### **git clone <repositorio>**

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

### **Repositorio remoto origin**

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

### **git remote -v**

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

# Cuestionario

◆ Indicar que respuesta define correctamente el siguiente término:

**git push origin master (2)**

- => 1. Botón de los repositorios de GitHub que permite clonar ese repositorio en otra cuenta en GitHub  
2. Comando que actualiza la rama master del repositorio remoto origen del local con las últimas versiones  
3. Identificador simbólico asociado al URL del repositorio remoto del que se ha clonado el repositorio local  
4. Comando que clona un repositorio remoto identificado por su URL  
5. Comando que muestra los repositorios remotos definidos en el local, junto con el URL y función asociados

**Botón de Fork (1)**

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

**git clone <repositorio> (4)**

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

**Repositorio remoto origin (3)**

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

**git remote -v (5)**

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

# Ejercicio opcional

- ◆ Copia con Fork el repositorio **jquemada/cal\_branches** de GitHub a tu cuenta
  - Inspeccionar ambos repositorios en GitHub (original y copiado) con el navegador Web
    - Comprobar que ambos tienen las mismas ramas con los mismos commits y los mismos identificadores
- ◆ Crear una nueva organización o cuenta que denominamos **<cuanta\_2>**
  - Creen una de las dos siguiendo las instrucciones que les da GitHub
- ◆ Copiar con Fork el repositorio **<tu\_cuenta>/cal\_branches** a **<cuanta\_2>**
  - Inspeccionar ambos repositorios en GitHub (original y copiado) con el navegador Web
    - Comprobar que también tiene las mismas ramas con los mismos commits y los mismos identificadores



# Git y GitHub

Ramas locales, remotas, tracking y refsspecs:  
branch, checkout, clone, fetch y pull

Juan Quemada, DIT - UPM

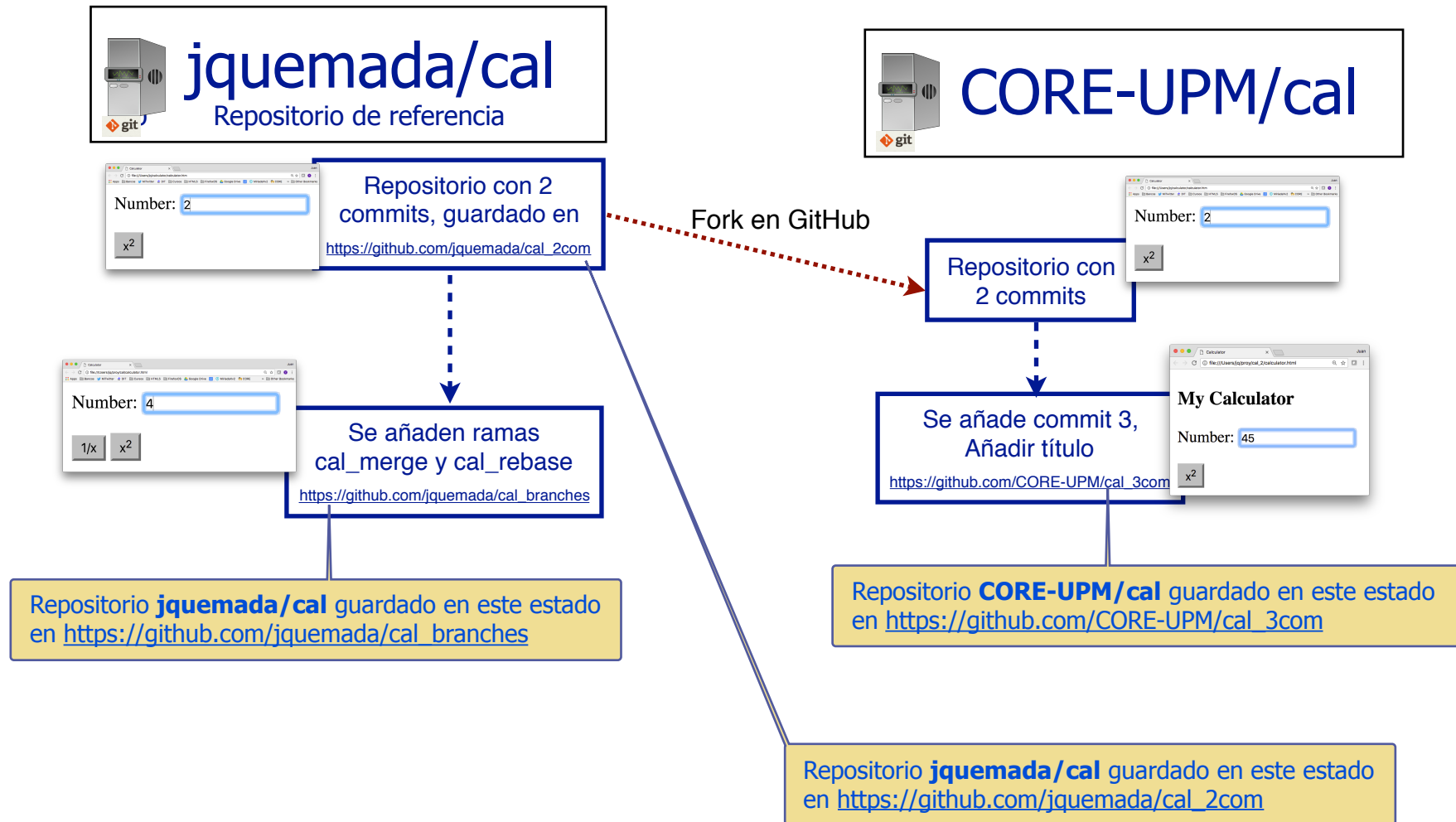
# Rama local, remota y tracking

- ◆ **Rama local:** rama donde se realiza el desarrollo de nuevos commits
  - Las ramas locales almacenan sus commits en el repositorio de commits en el directorio `.git`
- ◆ **Rama remota:** rama de un **remote** definido en el repositorio local
  - Son copias de las ramas del remote mostradas en el grafo de commits del repositorio local
    - Su estado se actualiza en las operaciones de sincronización: **clone**, **fetch**, **pull** o **push**
  - **remote:** nombre simbólico asociado al URL de un repositorio definido con **git remote ...**
- ◆ **Nombre de rama remota:** va precedido por `<remote>` o por `remotes/<remote>`
  - **origin/master** o **remotes/origin/master** identifica la rama **master** del **remote origin**
  - **cal/square** o **remotes/cal/square** identifica la rama **square** del **remote cal**
- ◆ **Rama tracking:** rama local asociada a una remota
  - La rama tracking simplifica las operaciones de sincronización con la remota
    - Una rama de desarrollo local suele ser tracking de la remota equivalente, por ej. **master** de **origin/master**





# Repositorios remotos utilizados



# Crear, enlazar o listar ramas remotas

## ◆ git clone ...

- Copiar un repositorio identificado por <URL> a un directorio local:
  - Define el repositorio remoto como repositorio remote origin, copia solo la rama master definiéndola como rama tracking de origin/master y actualiza todas las ramas remotas de origin
    - **git clone <https://github.com/jquemada/cal>** copia el repositorio remoto al directorio cal
    - **git clone <https://github.com/jquemada/cal> cal\_2** copia el repositorio remoto al directorio cal\_2

## ◆ git branch ...

- Muestra ramas locales, remotas o tracking con opciones **-a** (--all), **-r** (--remote) o **-vv** (--verbose)
  - **git branch -r** muestra solo las **ramas remotas** de un repositorio local
  - **git branch -a** muestra las **ramas locales** y **remotas** de un repositorio local
  - **git branch -vv** muestra **ramas locales**, **remotas** y **tracking** con sus ramas remotas asociadas y su estado

## ◆ git branch ...

- Crea una rama tracking asociada con una rama remota
  - **git branch -b square r1/square** crea rama tracking local **square** y la asocia como tracking a **r1/square**
  - **git branch --track r1/square** transforma la rama local **square**, ya existente, en tracking de **r1/square**
  - **git branch --track s2 r1/s1** transforma la rama local **s2**, ya existente, en tracking de **r1/s1**

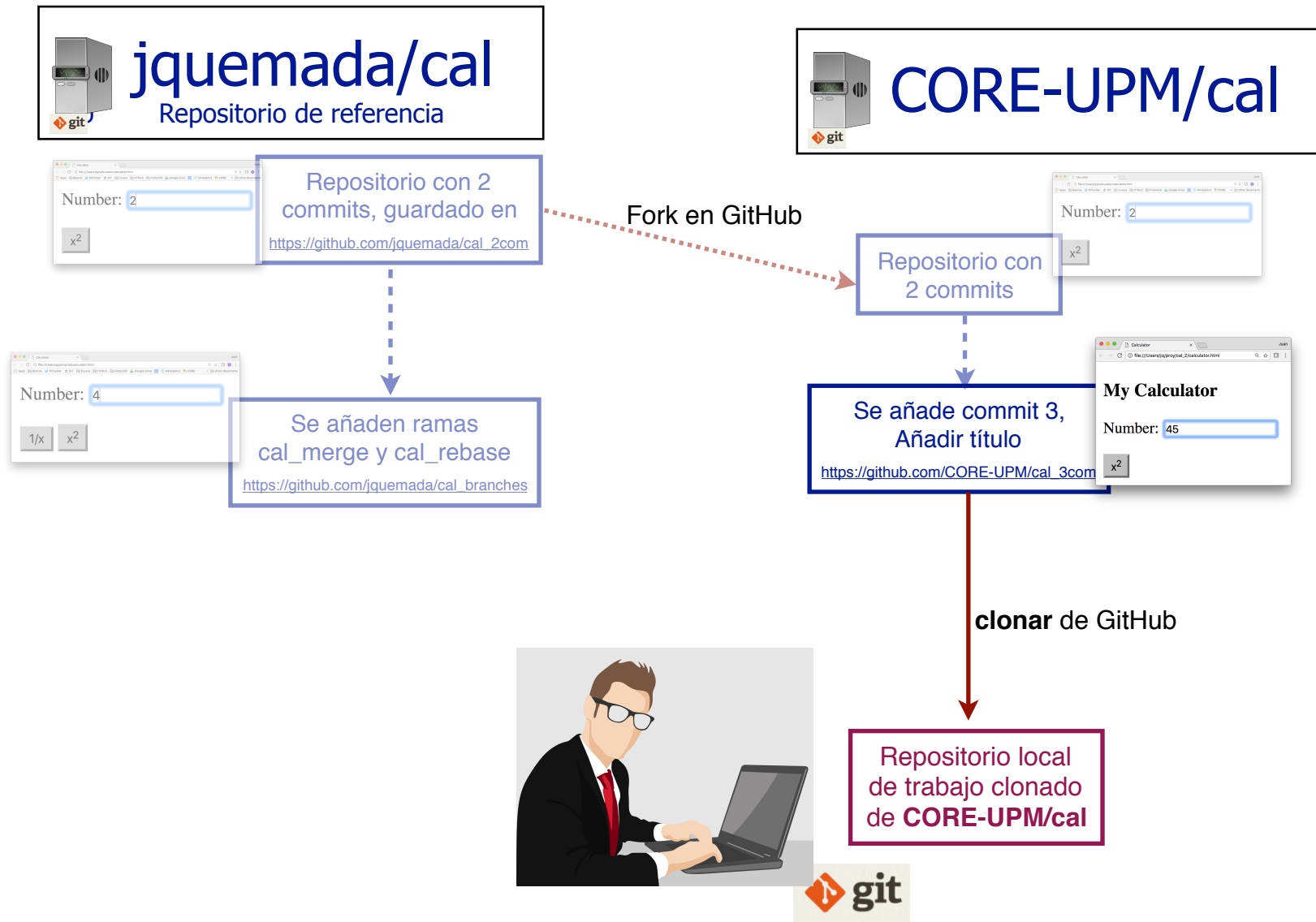
## ◆ git fetch ...

- Crea si no existen o actualiza las ramas remotas de un remote creado con **git remote ....**
  - **git fetch cal** crea localmente las **ramas remotas** del remote **cal** o actualiza su estado si existen
  - **git fetch** actualiza el estado de todas las ramas tracking
  - **git fetch --all** crea o actualiza el estado de todas las ramas de todos los remotes definidos
  - **git fetch -p origin** la opción -p (--prune) actualiza las ramas de origin eliminando las que ya no existen



# Clonar repositorio

# Clonar CORE-UPM/cal



# Clonar un repositorio remoto

Clonar el repositorio **CORE-UPM/cal** de GitHub. Es un fork de **jquemada/cal** que se realizó cuando tenía los 2 primeros commits y al que se añadió un tercer commit.

```
cal_2 — -bash — 73x21
venus:proy jq$
venus:proy jq$ git clone -q https://github.com/CORE-UPM/cal cal_2
venus:proy jq$ cd cal_2
venus:cal_2 jq$
venus:cal_2 jq$ git branch
* master
venus:cal_2 jq$
venus:cal_2 jq$ git log --oneline
5c70f9c Añadir título
b0e63ad x^2 button
1096247 Readme & License
venus:cal_2 jq$
venus:cal_2 jq$ git remote -v
origin https://github.com/CORE-UPM/cal (fetch)
origin https://github.com/CORE-UPM/cal (push)
venus:cal_2 jq$
venus:cal_2 jq$ git branch -a -vv
* master                    5c70f9c [origin/master] Añadir título
remotes/origin/HEAD         -> origin/master
remotes/origin/master       5c70f9c Añadir título
venus:cal_2 jq$
```

# El repositorio local clonado

Clonar el repositorio **CORE-UPM/cal** de GitHub. Es un fork de **jquemada/cal** que se realizó cuando tenía los 2 primeros commits y al que se añadió un tercer commit.

```
venus:proy jq$
venus:proy jq$ git clone -q https://github.com/CORE-UPM/cal cal_2
venus:proy jq$ cd cal_2
venus:cal_2 jq$
venus:cal_2 jq$ git branch
* master
venus:cal_2 jq$
venus:cal_2 jq$ git log --oneline
5c70f9c Añadir título
b0e63ad x^2 button
1096247 Readme & License
venus:cal_2 jq$
venus:cal_2 jq$ git remote -v
origin https://github.com/CORE-UPM/cal (fetch)
origin https://github.com/CORE-UPM/cal (push)
venus:cal_2 jq$
venus:cal_2 jq$ git branch -a -vv
* master                5c70f9c [origin/master] Añadir título
remotes/origin/HEAD      -> origin/master
remotes/origin/master    5c70f9c Añadir título
venus:cal_2 jq$
```

El repositorio local clonado solo contiene la rama master

La rama master contiene estos 3 commits

master

Añadir título

x^2 button

Readme & Lic.

# El repositorio remoto

```
cal_2 — -bash — 73x21
venus:proy jq$
venus:proy jq$ git clone -q https://github.com/CORE-UPM/cal cal_2
venus:proy jq$ cd cal_2
venus:cal_2 jq$
venus:cal_2 jq$ git branch
* master
venus:cal_2 jq$
venus:cal_2 jq$ git log --oneline
5c70f9c Añadir título
b0e63ad x^2 button
1096247 Readme & License
venus:cal_2 jq$
venus:cal_2 jq$ git remote -v
origin https://github.com/CORE-UPM/cal (fetch)
origin https://github.com/CORE-UPM/cal (push)
venus:cal_2 jq$
venus:cal_2 jq$ git branch -a -vv
* master 5c70f9c [origin/master] Añadir título
remotes/origin/HEAD -> origin/master
remotes/origin/master 5c70f9c Añadir título
venus:cal_2 jq$
```

**git remote -v** muestra como también se ha definido el remote **origin** (para operaciones fetch y push) asociado al repositorio en GitHub: <https://github.com/CORE-UPM/cal>

**git branch -a -vv** muestra la rama **master** definida como **tracking** de **origin/master**.

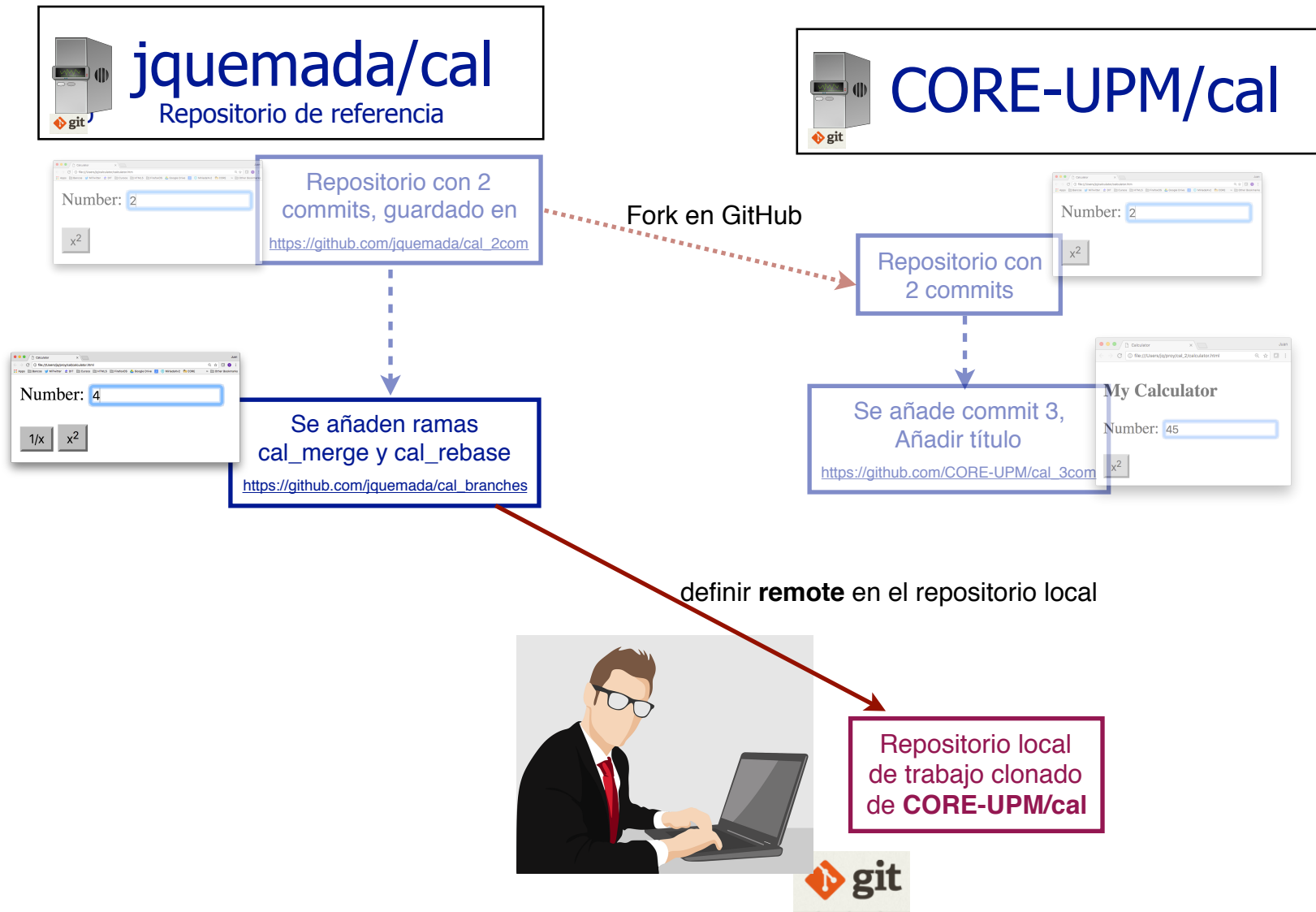
**git branch -a -vv** muestra las ramas remotas de origen definidas con fetch: HEAD y master



# Definir un remote y sus ramas



# jquemada/cal como remote



# Crear un remote

Se crea el remote **cal** asociado <https://github.com/jquemada/cal> del que se creo el repositorio remoto origin (<https://github.com/CORE-UPM/cal>) con **Fork** cuando tenía 2 commits.

```
venus:cal_2 jq$
venus:cal_2 jq$ git remote add cal https://github.com/jquemada/cal
venus:cal_2 jq$ git remote -v
cal      https://github.com/jquemada/cal (fetch)
cal      https://github.com/jquemada/cal (push)
origin   https://github.com/CORE-UPM/cal (fetch)
origin   https://github.com/CORE-UPM/cal (push)
venus:cal_2 jq$
venus:cal_2 jq$ git branch -a -vv
* master          5c70f9c [origin/master] Añadir título
remotes/origin/HEAD -> origin/master
remotes/origin/master 5c70f9c Añadir título
venus:cal_2 jq$
venus:cal_2 jq$ git log --oneline --graph --all
* 5c70f9c Añadir título
* b0e63ad x^2 button
* 1096247 Readme & License
venus:cal_2 jq$
```

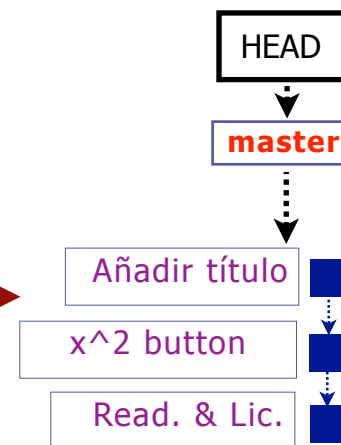
Muestra el nuevo remote cal.

# Crear un remote

**git branch -a -vv** muestra que al crear el remote cal no se han creado nuevas ramas remotas, esto ocurrirá cuando se actualicen con fetch.

**git log --oneline --graph --all** muestra que el grafo de commits tampoco se ha modificado. Solo tiene los 3 commits que se clonaron.

```
venus:cal_2 jq$  
venus:cal_2 jq$ git remote add cal https://github.com/jquemada/cal  
venus:cal_2 jq$  
venus:cal_2 jq$ git remote -v  
cal https://github.com/jquemada/cal (fetch)  
cal https://github.com/jquemada/cal (push)  
origin https://github.com/CORE-UPM/cal (fetch)  
origin https://github.com/CORE-UPM/cal (push)  
venus:cal_2 jq$  
venus:cal_2 jq$ git branch -a -vv  
* master 5c70f9c [origin/master] Añadir título  
remotes/origin/HEAD -> origin/master  
remotes/origin/master 5c70f9c Añadir título  
venus:cal_2 jq$  
venus:cal_2 jq$ git log --oneline --graph --all  
* 5c70f9c Añadir título  
* b0e63ad x^2 button  
* 1096247 Readme & License  
venus:cal_2 jq$
```



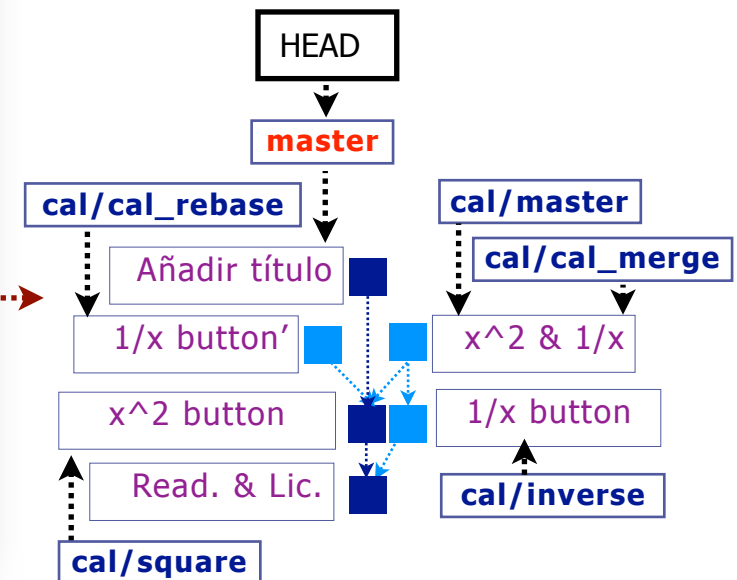
# Crear ramas remotas

**git fetch -q cal** crea **ramas remotas** en el repositorio local asociadas a las ramas que existen en el repositorio remoto asociado a cal.

```
venus:cal_2 jq$  
venus:cal_2 jq$ git fetch -q cal  
venus:cal_2 jq$  
venus:cal_2 jq$ git branch -a -vv  
* master 5c70f9c [origin/master] Añadir título  
remotes/cal/cal_merge 1898ac7 Integrate x^2 & 1/x  
remotes/cal/cal_rebase 795c2da 1/x button  
remotes/cal/inverse e868dc4 1/x button  
remotes/cal/master 1898ac7 Integrate x^2 & 1/x  
remotes/cal/square b0e63ad x^2 button  
remotes/origin/HEAD -> origin/master  
remotes/origin/master 5c70f9c Añadir título  
venus:cal_2 jq$  
venus:cal_2 jq$ git log --oneline --graph --all  
* 5c70f9c Añadir título  
| * 795c2da 1/x button  
|/  
| * 1898ac7 Integrate x^2 & 1/x  
| |\  
| |/  
| |/  
| |/  
| * b0e63ad x^2 button  
| * e868dc4 1/x button  
|/  
* 1096247 Readme & License  
venus:cal_2 jq$
```

**git branch -a -vv** muestra las ramas del remote **cal** creadas con fetch.

**git log --oneline --graph --all** muestra que el grafo de commits del repositorio local incluye ahora también los commits de las nuevas ramas remotas creadas.





# Copiar ramas remotas en locales

# Copiar una rama remota en una local

## ◆ Para añadir desarrollos a una **rama remota** debemos copiarla en una local

- Los desarrollos se realizan sobre la rama local añadiendo nuevos commits

## ◆ **git checkout ...**

- Copia una rama remota en una local tracking y restaura la rama local (solo cuando no existe)
  - **git checkout square** crea y restaura la rama tracking square asociada a **<remote>/square**
- Restaurar una rama remota en modo detached HEAD, es decir no asociada a ninguna rama
  - **git checkout origin/square** restaura **origin/square** en modo detached HEAD

## ◆ **git fetch ....**

- Copia una rama remota en una local utilizando refsspecs: **[+<local\_branch>:<remote\_branch>**
  - **git fetch origin square:sqrt**
    - Crea o actualiza la rama local sqrt con los commits de la remota **origin/square**
  - **git fetch origin pull/1/head:s1**
    - Crea o actualiza la rama local **s1** con el pull\_request 1 del repositorio remoto origin en GitHub
  - **git fetch cal\_branches +s1:s1**
    - Crea o actualiza la rama local **s1** con la remota **cal\_branches/s1** aunque sean incompatibles (+)
  - **git fetch https://github.com/jquemada/cal square:square**
    - Crea o actualiza la rama local square con la rem. square de **https://github.com/jquemada/cal**

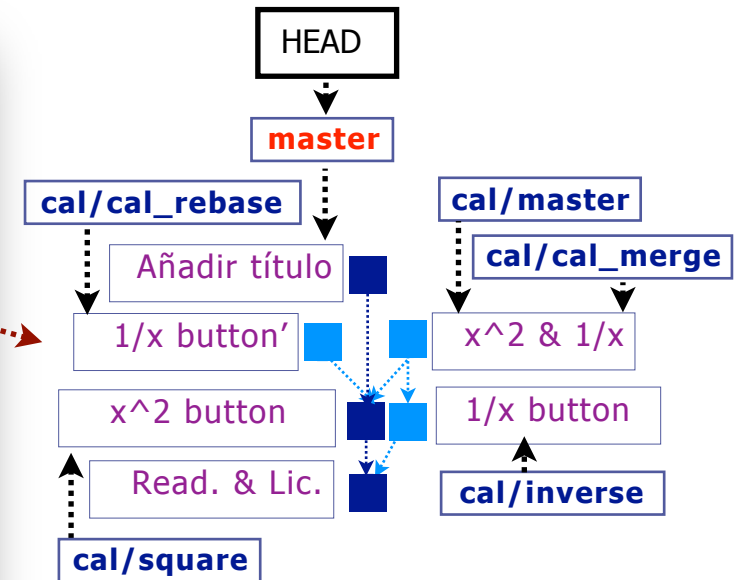
## ◆ **git pull ....**

- traer la rama remota indicada e integrarla con una rama del repositorio local
  - **git pull cal\_branches square** integra la rama **square** de **cal\_branches** en la rama activa
  - **git pull origin pull/1/head** Integra el **pull\_request #1** en la rama activa

# Ramas remotas y locales

**git branch -a -vv** muestra que master es la única rama local y todas las remotas de los remotes origin y cal.

```
venus:cal_2 jq$  
venus:cal_2 jq$ git branch -a -vv  
* master 5c70f9c [origin/master] Añadir título  
remotes/cal/cal_merge 1898ac7 Integrate x^2 & 1/x  
remotes/cal/cal_rebase 795c2da 1/x button  
remotes/cal/inverse e868dc4 1/x button  
remotes/cal/master 1898ac7 Integrate x^2 & 1/x  
remotes/cal/square b0e63ad x^2 button  
remotes/origin/HEAD -> origin/master  
remotes/origin/master 5c70f9c Añadir título  
venus:cal_2 jq$  
venus:cal_2 jq$ git checkout -q cal_merge  
venus:cal_2 jq$  
venus:cal_2 jq$ git branch -vv  
* cal_merge 1898ac7 [cal/cal_merge] Integrate x^2 & 1/x  
master 5c70f9c [origin/master] Añadir título  
venus:cal_2 jq$  
venus:cal_2 jq$  
venus:cal_2 jq$ git fetch -q cal cal_rebase:cal_rebase  
venus:cal_2 jq$  
venus:cal_2 jq$ git branch -vv  
* cal_merge 1898ac7 [cal/cal_merge] Integrate x^2 & 1/x  
cal_rebase 795c2da 1/x button  
master 5c70f9c [origin/master] Añadir título  
venus:cal_2 jq$
```



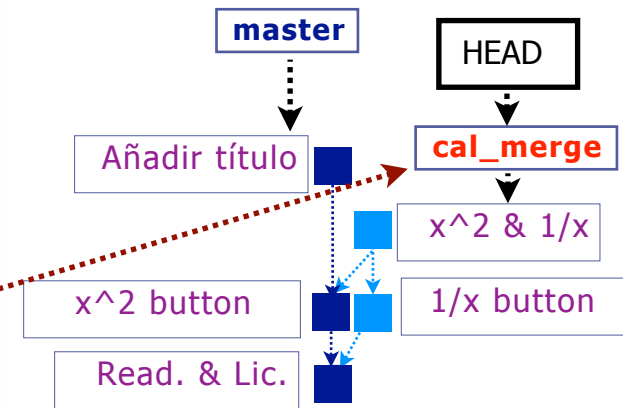
# checkout: crea rama local tracking

```
venus:cal_2 jq$
venus:cal_2 jq$ git branch -a -vv
* master          5c70f9c [origin/master] Añadir título
remotes/cal/cal_merge 1898ac7 Integrate x^2 & 1/x
remotes/cal/cal_rebase 795c2da 1/x button
remotes/cal/inverse e868dc4 1/x button
remotes/cal/master 1898ac7 Integrate x^2 & 1/x
remotes/cal/square b0e63ad x^2 button
remotes/origin/HEAD -> origin/master
remotes/origin/master 5c70f9c Añadir título

venus:cal_2 jq$
venus:cal_2 jq$ git checkout -q cal_merge
venus:cal_2 jq$
venus:cal_2 jq$ git branch -vv
* cal_merge 1898ac7 [cal/cal_merge] Integrate x^2 & 1/x
  master    5c70f9c [origin/master] Añadir título

venus:cal_2 jq$
venus:cal_2 jq$
venus:cal_2 jq$ git fetch -q cal cal_rebase:cal_rebase
venus:cal_2 jq$
venus:cal_2 jq$ git branch -vv
* cal_merge 1898ac7 [cal/cal_merge] Integrate x^2 & 1/x
  cal_rebase 795c2da 1/x button
  master     5c70f9c [origin/master] Añadir título

venus:cal_2 jq$
```



**git checkout cal\_merge** copia la rama remota **cal/cal\_merge** en una local del mismo nombre y la restaura en el directorio de trabajo.

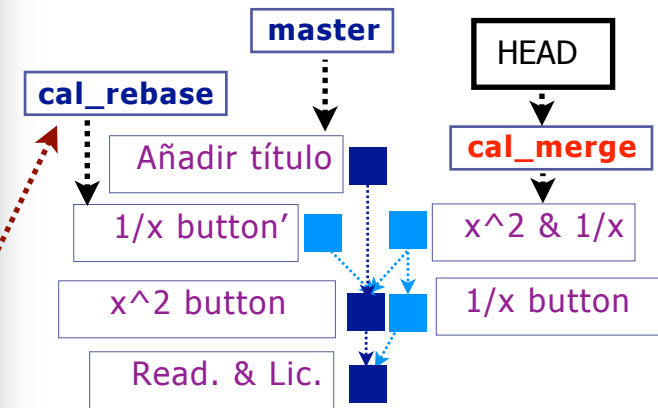
Es una forma muy cómoda de **restaurar ramas locales** para trabajar en ellas.

**git branch -vv** muestra como la nueva rama creada y restaurada es tracking.



# fetch con refspec: crea rama local

```
venus:cal_2 jq$  
venus:cal_2 jq$ git branch -a -vv  
* master          5c70f9c [origin/master] Añadir título  
remotes/cal/cal_merge 1898ac7 Integrate x^2 & 1/x  
remotes/cal/cal_rebase 795c2da 1/x button  
remotes/cal/inverse e868dc4 1/x button  
remotes/cal/master 1898ac7 Integrate x^2 & 1/x  
remotes/cal/square b0e63ad x^2 button  
remotes/origin/HEAD -> origin/master  
remotes/origin/master 5c70f9c Añadir título  
venus:cal_2 jq$  
venus:cal_2 jq$ git checkout -q cal_merge  
venus:cal_2 jq$  
venus:cal_2 jq$ git branch -vv  
* cal_merge 1898ac7 [cal/cal_merge] Integrate x^2 & 1/x  
  master    5c70f9c [origin/master] Añadir título  
venus:cal_2 jq$  
venus:cal_2 jq$ git fetch -q cal cal_rebase:cal_rebase  
venus:cal_2 jq$  
venus:cal_2 jq$ git branch -vv  
* cal_merge 1898ac7 [cal/cal_merge] Integrate x^2 & 1/x  
  cal_rebase 795c2da 1/x button  
  master     5c70f9c [origin/master] Añadir título  
venus:cal_2 jq$
```



**git fetch cal cal\_rebase:cal\_rebase** es otra forma de crear la rama local (cal\_rebase) utilizando fetch y refspecs a partir de la rama remota (cal/cal\_rebase)

**git branch -vv** muestra como la nueva rama creada ahora **no es tracking**.



Final del tema