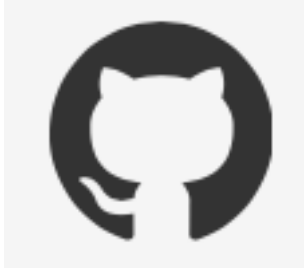




Git y GitHub

Índice: Git y GitHub

◆ <u>Introducción a Git y a sus comandos</u>	<u>3</u>
◆ <u>El repositorio local y el directorio de trabajo: init, log, status, diff, add, reset y commit</u>	<u>6</u>
◆ <u>Repositorios compartidos en GitHub: new_repository y push</u>	<u>22</u>
◆ <u>Crear commit inicial en GitHub y clonar: new_repository, clone, remote y push</u>	<u>36</u>



Git y GitHub

Introducción a Git y a sus comandos

Git



◆ Git es un gestor de **repositorios** de versiones software

- Desarrollado por Linus Torwalds en 2005 en código libre
 - Para soportar el desarrollo de Linux

◆ git es un comando de UNIX/Linux

- Documentación: <https://git-scm.com/documentation>
- Instalación de git o actualización a la última versión
 - Instrucciones de GitHub: <https://help.github.com/articles/set-up-git/>
 - Instrucciones git-scm: <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>
 - Tutorial Atlassian: <https://www.atlassian.com/git/tutorials/install-git>

\$ **git --version** # Indica la versión instalada. Si Git no esta instalado, lo indicará.

\$ **git --help** # Equivale a git help y muestra lista de los comandos de git mas habituales.

```
venus:proy jq$  
venus:proy jq$ git --help  
usage: git [--version] [--help] [-C <path>] [-c name=value]  
        [--exec-path=<path>] [--html-path] [--man-path] [--info-path]  
        [-p|--paginate|--no-pager] [--no-replace-objects] [--bare]  
        [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]  
        <command> [<args>]
```

The most commonly used git commands are:

add	Add file contents to the index
bisect	Find by binary search the change that introduced a bug
branch	List, create, or delete branches

<https://git-scm.com/docs>

Git: manuales en línea y configuración

```
$ git init ... # git es un meta-comando, donde el primer parámetro (init) la operación solicitada  
# Los parámetros add, bisect, branch, checkout,... invocan distintas operaciones.
```

```
$ git init --help # Equivale a git help init, ayuda del comando git init, igual para: add, bisect, ..
```

```
# El comando git config configura git con las credenciales del desarrollador:
```

```
$ git config --global user.name "Pedro Ramirez"  
$ git config --global user.email pramirez@dit.upm.es
```

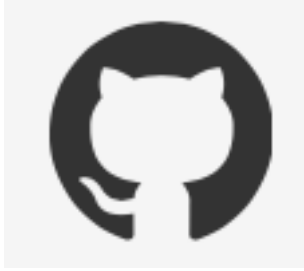
```
# Consultar el valor de todas las opciones configuradas:
```

```
$ git config --list  
user.name=Pedro Ramirez  
user.email=pramirez@dit.upm.es  
color.ui=true  
.....
```

```
# Consultar el valor de una opción:
```

```
$ git config user.name  
Pedro Ramirez  
$
```

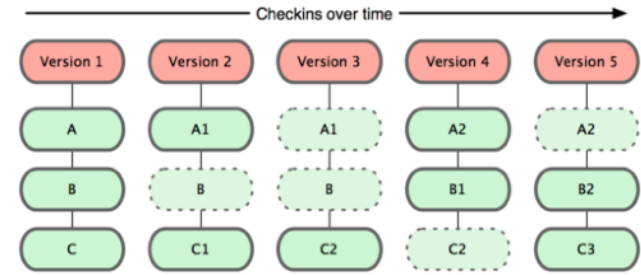
El desarrollador debe **firmar** todos los **commits** que crea en la historia de un proyecto, porque Git es un software de colaboración. Por ello debe configurar sus credenciales antes de utilizar Git.



Git y GitHub

El repositorio local y el directorio de trabajo: init, log, status, diff, add, reset y commit

Proyecto Software



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

◆ Los proyectos software están activos durante largos periodos

- Durante su vida generan muchas versiones y variantes diferentes
 - Unas corrigen errores, otras añaden funcionalidad o adaptan nuevo hardware/software, etc.

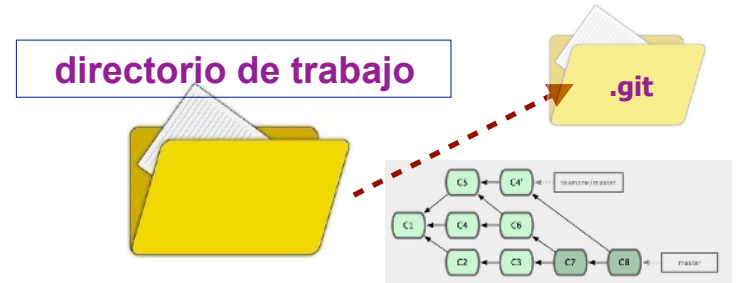
◆ Commit o versión

- Ficheros de un proyecto guardados en un repositorio, que pueden restaurarse
 - Algunos commits se etiquetan con tags especiales de versión, p.e. v1, v1.3, ..

◆ Rama

- Secuencia de commits ordenada por fechas que soporta un desarrollo
 - Los nuevos commits se añaden al final de la rama de desarrollo
- La **rama de desarrollo principal** se denomina **rama master**
 - La rama master se crea automáticamente al crear el primer commit

Directorio de trabajo y repositorio de commits



◆ Directorio de trabajo

- **Directorio** del S.O. donde se crean las **versiones** del proyecto: código fuente, datos, ...
 - Se denomina también **área o espacio de trabajo** (workspace)
 - **árbol de trabajo** (work-tree) por la estructura en árbol de los subdirectorios que agrupan ficheros
 - **base de código** (codebase)

◆ El comando: **git init**

- Transforma el directorio del S.O. en un **directorio de trabajo Git**
 - Añadiendo el **repositorio de commits** al directorio de trabajo
 - Añade el **subdirectorio oculto .git** con una base de datos donde guardar commits

◆ **git init** debe invocarse en el dir. de trabajo, como otros comandos Git

- Es decir, si el **directorio de trabajo del terminal** es el **directorio de trabajo Git**

Índice o area de cambios

◆ Índice o área de cambios (staging area, index)

- **Registro de cambios** del directorio de trabajo a incluir en el próximo commit
 - Los cambios no registrados en el índice no se incluyen al generar un nuevo commit
- Los ficheros **no modificados** del commit anterior siguen también en el siguiente

◆ git add ...

- registra en el índice los ficheros indicados
 - `git add .` registra en el índice todos los ficheros **nuevos** o **modificados**
 - `git add LICENSE README.md` registra los ficheros **LICENSE README.md** en el índice

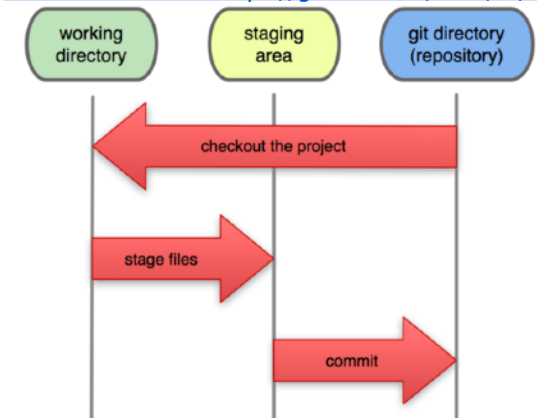
◆ git reset ...

- extrae **<ficheros>** del índice (deshace `git add ..`)
 - `git reset .` extrae del índice todos los ficheros
 - `git reset LICENSE` extrae LICENSE del índice

◆ git commit ...

- Genera un nuevo commit con lo registrado en el índice
 - `git commit -m "Descripción"` guarda nuevo commit con mensaje o título **"Descripción"**
 - `git commit` guarda nuevo commit y abre editor vi/vim para crear mensaje del commit

*de Scott Chanson: <https://git-scm.com/book/es/v1>



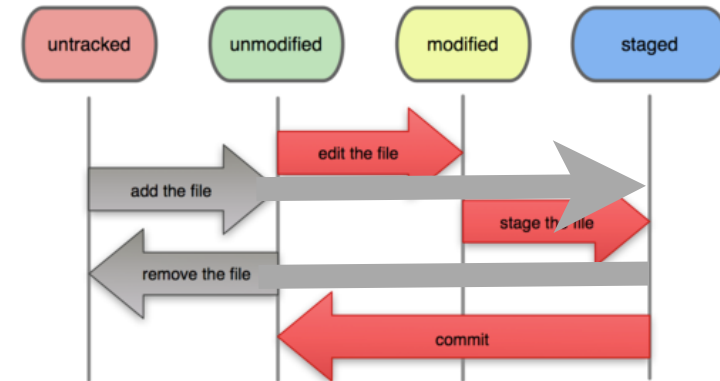
Análisis del directorio de trabajo

◆ Git controla **cambios** en los ficheros del dir. de trabajo y de sus subdir.

- Los cambios son siempre respecto al commit anterior

◆ git status

- muestra estado de los ficheros del directorio:
 - **modified**: modificados respecto al commit anterior
 - **untracked**: no existentes en el commit anterior
 - **staged**: registrados para el próximo commit
- **git status -s**
 - Muestra estado en formato compacto muy cómodo y conciso



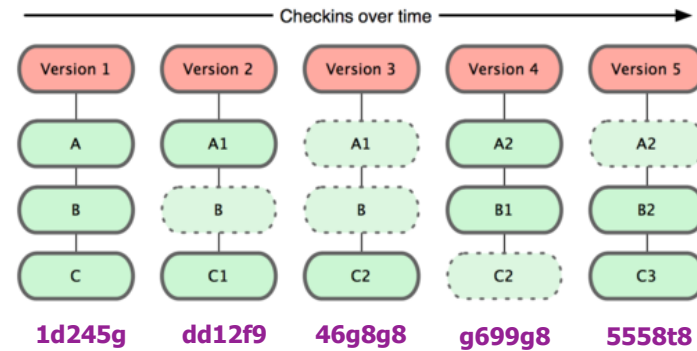
*de Scott Chanson: <https://git-scm.com/book/es/v1>

◆ git diff

- Mostrar diferencias en los ficheros **modified** respecto al commit anterior
 - **git diff** muestra los cambios en todos los ficheros **modified** del directorio de trabajo
 - **git diff README.md** muestra solo los cambios en **README.md**, pero solo si es **modified**
- Mostrar diferencias en los ficheros **staged** respecto al commit anterior
 - **git diff --cached** muestra los cambios en todos los ficheros **staged** del directorio de trabajo
 - **git diff --cached README.md** muestra solo los cambios en **README.md**, pero solo si es **staged**
- **git diff** muestra las diferencias en las líneas del código así:
 - **Líneas añadidas**: en verde y comienzan por +
 - **Líneas eliminadas**: en rojo y comienzan por -

Identificador de commit

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



◆ **git commit ...** asigna un **identificador único** a cada nuevo commit

- El identificador actua como nombre o referencia única del commit
 - Ningún otro commit en ningún otro repositorio poseerá el mismo identificador
 - Garantiza la integridad del commit: igualdad de identificadores implica igualdad de commits

◆ **Identificador de commit**

- Número hexadecimal de 40 dígitos generado como clave de hash SHA1
 - Ejemplo de identificador: **973751d2**1c4a71f13a2e729ccf77f3a960885682

◆ Se suele utilizar el **formato corto** (formato largo es incomodo)

- 7-8 dígitos iniciales (únicos en un proyecto): **973751d2**
 - Los comandos git permiten **identificadores cortos o largos**
- **git log --oneline** muestra la historia de commits (e ids cortos) de una rama

Ejemplo: Crear directorio del S.O.

Crear el **directorio proy** del S.O. y entrar en él.

```
venus:proy jq$  
venus:proy jq$ mkdir cal  
venus:proy jq$ cd cal  
venus:cal jq$  
venus:cal jq$ git init  
Initialized empty Git repository in /Users/jq/proy/cal/.git/  
venus:cal jq$  
venus:cal jq$ git status -s  
venus:cal jq$  
venus:cal jq$ # Edit README.md & LICENSE  
venus:cal jq$  
venus:cal jq$ git status -s  
?? LICENSE  
?? README.md  
venus:cal jq$  
venus:cal jq$ git add .  
venus:cal jq$  
venus:cal jq$ git status -s  
A LICENSE  
A README.md  
venus:cal jq$  
venus:cal jq$ git commit -m "Readme & License"  
[master (root-commit) 1096247] Readme & License  
2 files changed, 23 insertions(+)  
create mode 100644 LICENSE  
create mode 100644 README.md  
venus:cal jq$  
venus:cal jq$ git log --oneline  
1096247 Readme & License  
venus:cal jq$
```



Crear directorio de trabajo GIT y repositorio vacío

```
venus:proy jq$  
venus:proy jq$ mkdir cal  
venus:proy jq$ cd cal  
venus:cal jq$  
venus:cal jq$ git init  
Initialized empty Git repository in /Users/jq/proy/cal/.git/  
venus:cal jq$  
venus:cal jq$ git status -s  
venus:cal jq$  
venus:cal jq$ # Edit README.md & LICENSE  
venus:cal jq$  
venus:cal jq$ git status -s  
?? LICENSE  
?? README.md  
venus:cal jq$  
venus:cal jq$ git add .  
venus:cal jq$  
venus:cal jq$ git status -s  
A LICENSE  
A README.md  
venus:cal jq$  
venus:cal jq$ git commit -m "Readme & License"  
[master (root-commit) 1096247] Readme & License  
2 files changed, 23 insertions(+)  
create mode 100644 LICENSE  
create mode 100644 README.md  
venus:cal jq$  
venus:cal jq$ git log --oneline  
1096247 Readme & License  
venus:cal jq$
```

Transformar cal en un directorio de trabajo Git con su repositorio en .git

directorio de trabajo Git

Muestra directorio sin cambios.

Crear ficheros del commit y registrarlos

```
venus:proy jq$  
venus:proy jq$ mkdir cal  
venus:proy jq$ cd cal  
venus:cal jq$  
venus:cal jq$ git init  
Initialized empty Git repository in /Users/jq/proy/cal/.git/  
venus:cal jq$  
venus:cal jq$ git status -s  
venus:cal jq$  
venus:cal jq$ # Edit README.md & LICENSE  
venus:cal jq$  
venus:cal jq$ git status -s  
?? LICENSE  
?? README.md  
venus:cal jq$  
venus:cal jq$ git add .  
venus:cal jq$  
venus:cal jq$ git status -s  
A LICENSE  
A README.md  
venus:cal jq$  
venus:cal jq$ git commit -m "Readme & License"  
[master (root-commit) 1096247] Readme & License  
2 files changed, 23 insertions(+)  
create mode 100644 LICENSE  
create mode 100644 README.md  
venus:cal jq$  
venus:cal jq$ git log --oneline  
1096247 Readme & License  
venus:cal jq$
```

Llevar estos 2 ficheros al directorio de trabajo. Copiarlos o editarlos: vi, vim, sublime-text, Webstorm, Atom,

directorio de trabajo Git



```
# cal  
Educational Git project. Creates a simple calculator in  
HTML and JavaScript in short steps.
```

MIT License

Copyright (c) 2016 Juan Quemada

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Crear ficheros del commit y registrarlos

```
venus:proy jq$  
venus:proy jq$ mkdir cal  
venus:proy jq$ cd cal  
venus:cal jq$  
venus:cal jq$ git init  
Initialized empty Git repository in /Users/jq/proy/cal/.git/  
venus:cal jq$  
venus:cal jq$ git status -s  
venus:cal jq$  
venus:cal jq$ # Edit README.md & LICENSE  
venus:cal jq$  
venus:cal jq$ git status -s  
?? LICENSE  
?? README.md  
venus:cal jq$  
venus:cal jq$ git add .  
venus:cal jq$  
venus:cal jq$ git status -s  
A LICENSE  
A README.md  
venus:cal jq$  
venus:cal jq$ git commit -m "Readme & License"  
[master (root-commit) 1096247] Readme & License  
2 files changed, 23 insertions(+)  
create mode 100644 LICENSE  
create mode 100644 README.md  
venus:cal jq$  
venus:cal jq$ git log --oneline  
1096247 Readme & License  
venus:cal jq$
```

Muestra los 2 ficheros sin registrar en el índice

directorio de trabajo Git



Crear ficheros del commit y registrarlos

```
venus:proy jq$  
venus:proy jq$ mkdir cal  
venus:proy jq$ cd cal  
venus:cal jq$  
venus:cal jq$ git init  
Initialized empty Git repository in /Users/jq/proy/cal/.git/  
venus:cal jq$  
venus:cal jq$ git status -s  
venus:cal jq$  
venus:cal jq$ # Edit README.md & LICENSE  
venus:cal jq$  
venus:cal jq$ git status -s  
?? LICENSE  
?? README.md  
venus:cal jq$  
venus:cal jq$ git add .  
venus:cal jq$  
venus:cal jq$ git status -s  
A LICENSE  
A README.md  
venus:cal jq$  
venus:cal jq$ git commit -m "Readme & License"  
[master (root-commit) 1096247] Readme & License  
2 files changed, 23 insertions(+)  
create mode 100644 LICENSE  
create mode 100644 README.md  
venus:cal jq$  
venus:cal jq$ git log --oneline  
1096247 Readme & License  
venus:cal jq$
```

Registrar ficheros en el índice

directorio de
trabajo Git



Crear ficheros del commit y registrarlos

```
venus:proy jq$  
venus:proy jq$ mkdir cal  
venus:proy jq$ cd cal  
venus:cal jq$  
venus:cal jq$ git init  
Initialized empty Git repository in /Users/jq/proy/cal/.git/  
venus:cal jq$  
venus:cal jq$ git status -s  
venus:cal jq$  
venus:cal jq$ # Edit README.md & LICENSE  
venus:cal jq$  
venus:cal jq$ git status -s  
?? LICENSE  
?? README.md  
venus:cal jq$  
venus:cal jq$ git add .  
venus:cal jq$  
venus:cal jq$ git status -s  
A LICENSE  
A README.md  
venus:cal jq$  
venus:cal jq$ git commit -m "Readme & License"  
[master (root-commit) 1096247] Readme & License  
2 files changed, 23 insertions(+)  
create mode 100644 LICENSE  
create mode 100644 README.md  
venus:cal jq$  
venus:cal jq$ git log --oneline  
1096247 Readme & License  
venus:cal jq$
```

Muestra los ficheros registrados

directorio de
trabajo Git



Crear nuevo commit y ver historia

```
venus:proy jq$  
venus:proy jq$ mkdir cal  
venus:proy jq$ cd cal  
venus:cal jq$  
venus:cal jq$ git init  
Initialized empty Git repository in /Users/jq/proy/cal/.git/  
venus:cal jq$  
venus:cal jq$ git status -s  
venus:cal jq$  
venus:cal jq$ # Edit README.md & LICENSE  
venus:cal jq$  
venus:cal jq$ git status -s  
?? LICENSE  
?? README.md  
venus:cal jq$  
venus:cal jq$ git add .  
venus:cal jq$  
venus:cal jq$ git status -s  
A LICENSE  
A README.md  
venus:cal jq$  
venus:cal jq$ git commit -m "Readme & License"  
[master (root-commit) 1096247] Readme & License  
2 files changed, 23 insertions(+)  
create mode 100644 LICENSE  
create mode 100644 README.md  
venus:cal jq$  
venus:cal jq$ git log --oneline  
1096247 Readme & License  
venus:cal jq$
```

Generar nuevo commit.

Muestra nuevo commit
en formato 1 línea

directorio de
trabajo Git



master

Readme & License

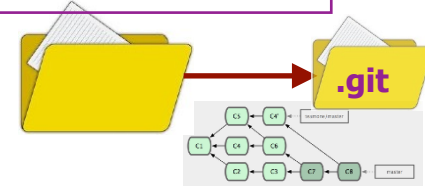
Inspeccionar historia y área de trabajo

```
venus:cal jq$  
venus:cal jq$  
venus:cal jq$ git log --oneline  
1096247 Readme & License  
venus:cal jq$  
venus:cal jq$ git status -s  
venus:cal jq$ # Edit calculator.html  
venus:cal jq$  
venus:cal jq$ git status -s  
?? calculator.html  
venus:cal jq$  
venus:cal jq$ git add calculator.html  
venus:cal jq$  
venus:cal jq$ git commit -m "x^2 button"  
[master b0e63ad] x^2 button  
1 file changed, 17 insertions(+)  
create mode 100644 calculator.html  
venus:cal jq$  
venus:cal jq$ git log --oneline  
b0e63ad x^2 button  
1096247 Readme & License  
venus:cal jq$
```

Muestra el primer commit ya generado.

Muestra el directorio de trabajo Git limpio.

directorio de trabajo



master

Readme & License

Añadir nuevo fichero

```
venus:cal jq$
venus:cal jq$
venus:cal jq$ git log --oneline
1096247 Readme & License
venus:cal jq$
venus:cal jq$ git status -s
venus:cal jq$ # Edit calculator.html
venus:cal jq$
venus:cal jq$ git status -s
?? calculator.html
venus:cal jq$
venus:cal jq$ git add calculator.html
venus:cal jq$
venus:cal jq$ git commit -m "x^2 button"
[master b0e63ad] x^2 button
1 file changed, 17 insertions(+)
create mode 100644 calculator.html
venus:cal jq$
venus:cal jq$ git log --oneline
b0e63ad x^2 button
1096247 Readme & License
venus:cal jq$
```

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

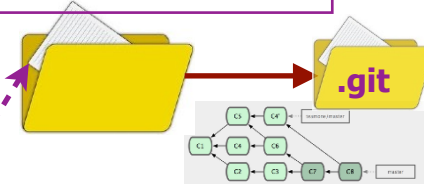
```
<!DOCTYPE html><html><head>
<title>Calculator</title><meta charset="utf-8">
<script type="text/javascript">

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>
```

Number: 2

x^2

directorio de trabajo

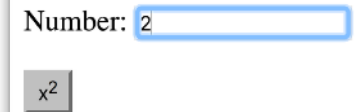


Muestra el fichero **calculator.html** en el directorio de trabajo todavía sin registrar.

master

Readme & License

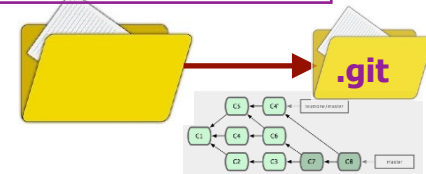
Añadir un nuevo commit a master



```
venus:cal jq$
venus:cal jq$
venus:cal jq$ git log --oneline
1096247 Readme & License
venus:cal jq$
venus:cal jq$ git status -s
venus:cal jq$ # Edit calculator.html
venus:cal jq$
venus:cal jq$ git status -s
?? calculator.html
venus:cal jq$
venus:cal jq$ git add calculator.html
venus:cal jq$
venus:cal jq$ git commit -m "x^2 button"
[master b0e63ad] x^2 button
1 file changed, 17 insertions(+)
create mode 100644 calculator.html
venus:cal jq$
venus:cal jq$ git log --oneline
b0e63ad x^2 button
1096247 Readme & License
venus:cal jq$
```

Registra fichero cal_square.html en el índice.

directorio de trabajo



Generar nuevo commit con mensaje "x^2 button"

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

master

x^2 button

Readme & License



Git y GitHub

Repositorios compartidos en GitHub:
`new_repository` y `push`

Repositorios de trabajo o compartidos

◆ Repositorio de **trabajo** o **local**

- Tiene un **directorio de trabajo** donde desarrollar el proyecto software
 - Además del directorio de trabajo tienen el **repositorio de commits** en **.git**
 - Se crean con: **git init**
- **No se puede actualizar remotamente** con: **git push ..**

◆ Repositorio **compartido** o **remoto**

- **Solo** tienen el **repositorio de commits**, no tienen directorio de trabajo
 - Suelen estar en un **servidor en Internet** y se identifican por un **URL**
 - Por ejemplo: **<https://github.com/jquemada/cal>**
 - Se crean con: **git init --bare**
- Se puede **actualizar remotamente** con: **git push ..**

◆ Portales Web de repositorios compartidos: **GitHub, Bitbucket, ..**

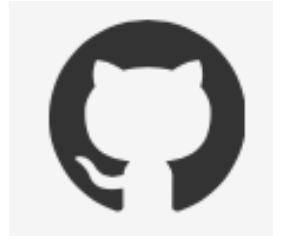
- Los utilizan equipos u organizaciones para alojar **repositorios compartidos**
 - Suelen permitir tanto acceso web, como acceso con comandos Git



GitHub

- ◆ **Portal** de repositorios Git -> lema "Social coding"
 - Red social donde programadores comparten repositorios remotos Git
 - Nos da acceso a ellos a través del navegador Web (además de Git)
- ◆ Repositorios **públicos son gratis**, los privados de pago
 - Algunos proyectos libres en Github: Linux, Eclipse, jQuery, RoR, ...
- ◆ GitHub suele albergar **repositorios compartidos** por un equipo
 - Organización con proyectos del curso: <https://github.com/CORE-UPM>
- ◆ Este curso requiere tener cuenta en GitHub: <https://github.com>
 - Al crearla nos da instrucciones claras y precisas sobre uso de GitHub y **Git**

Funciones principales de GitHub

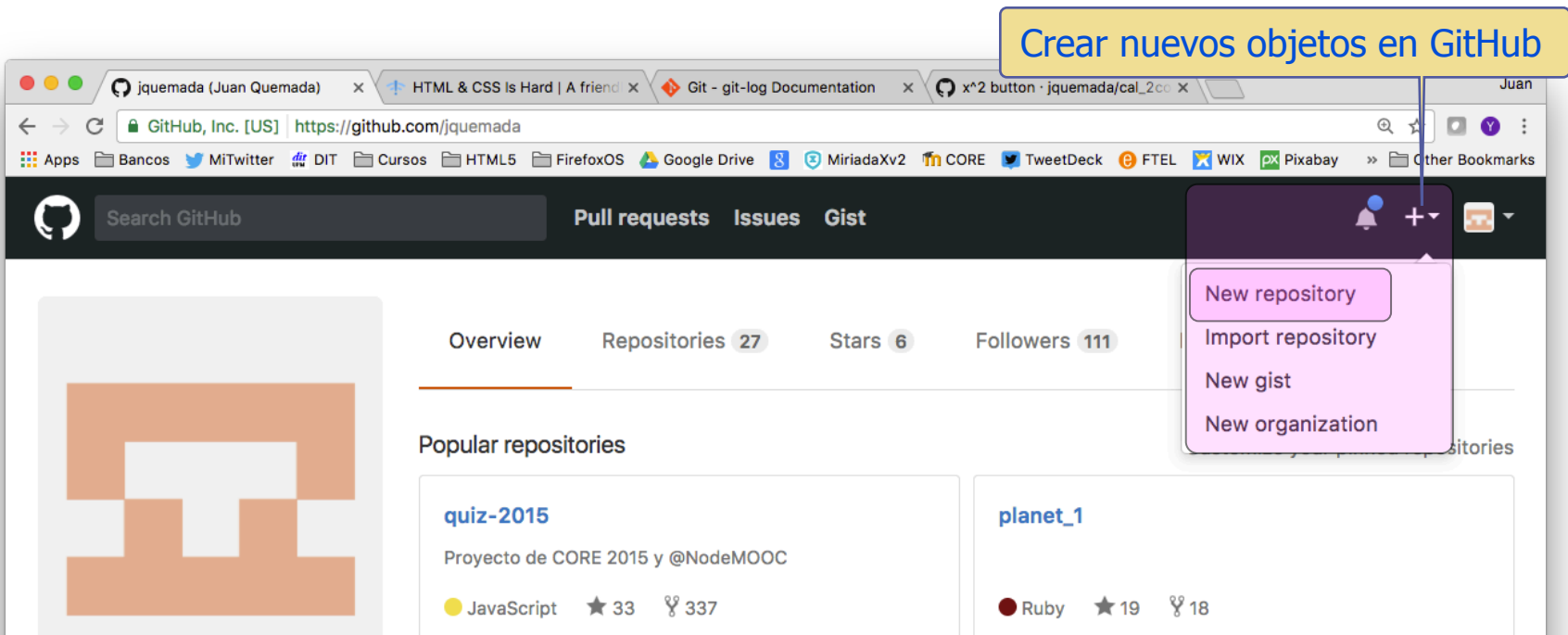


- ◆ La función principal de **GitHub** es **compartir** repositorios con terceros
- ◆ Las operaciones principales de un **usuario registrado** son
 - **Crear repositorio remoto** inicial nuevo para albergar un proyecto
 - Utilizando el botón: **New repository**
 - **Realiza copia** de un repositorio albergado en GitHub (para contribuir)
 - Utilizando el botón: **Fork**
 - **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
- ◆ Permite operaciones Git de **sincronización de repositorios**
 - **push** (subir rama), **clone** (clonar repositorio), **fetch** (traer rama), **pull** ..

Crear un repositorio vacío en GtHub

◆ GitHub permite gestión de repositorios remotos vía Web

- Un repositorio compartido vacío se crea desde el navegador
 - En este tema se crea un repositorio en GitHub accesible en el URL
 - <https://github.com/jquemada/cal>
- Los repositorios iniciales vacíos se crean con el botón: **New repository**
 - GitHub los crea en sus servidores invocando: **git init --bare ...**



Crear nuevos objetos en GitHub

GitHub, Inc. [US] | <https://github.com/jquemada>

Search GitHub Pull requests Issues Gist

Overview Repositories 27 Stars 6 Followers 111

Popular repositories

quiz-2015
Proyecto de CORE 2015 y @NodeMOOC
JavaScript ★ 33 337

planet_1
Ruby ★ 19 18

Crear jquemada/cal vacío en GitHub

Crear nuevo repositorio vacío en GitHub

New repository
Import repository
New gist
New organization

Nombre del repositorio

Repositorio público sin .gitignore, LICENSE y README

Create a new repository

A repository contains all the files for your project, including

Owner Repository name

jquemada / cal

Great repository names are short and memorable. Need ins

Public

Anyone can see this repository. You choose who can commi

Private

You choose who can see and commit to this repository.

Initialize this repository with a README

This will let you immediately clone the repository to your comput
repository.

Add .gitignore: None

Add a license: None

Create repository

jquemada / cal

<> Code

Nuevo repositorio creado con URL:
<https://github.com/jquemada/cal>

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH <https://github.com/jquemada/cal.git>

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# cal" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/jquemada/cal.git
git push -u origin master
```

Instrucciones de uso del repositorio.

...or push an existing repository from the command line

```
git remote add origin https://github.com/jquemada/cal.git
git push -u origin master
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS

Import code

Actualizar repositorios en GitHub con git push

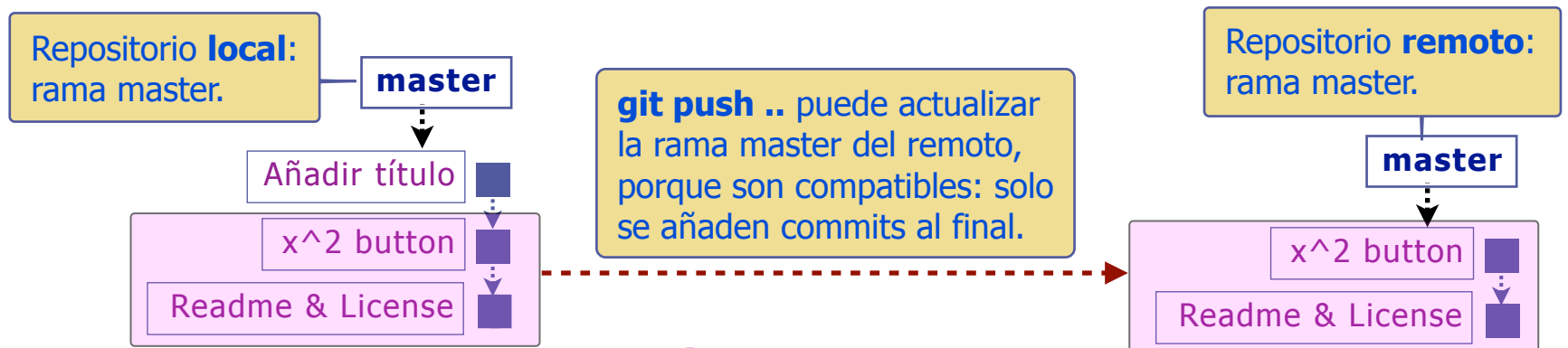
Actualizar un repositorio en GitHub con push

◆ 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

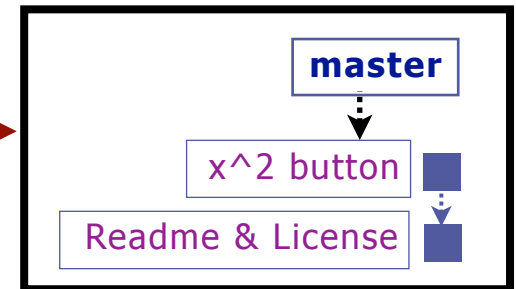
- Se debe tener credenciales de acceso al repositorio remoto
 - Por ejemplo, un repositorio en una cuenta u organización del usuario que lo actualiza
- La actualización de commits debe ser compatible con la rama actualizada en el remoto
 - Solo debe **añadir nuevos commits al final de la rama remota**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



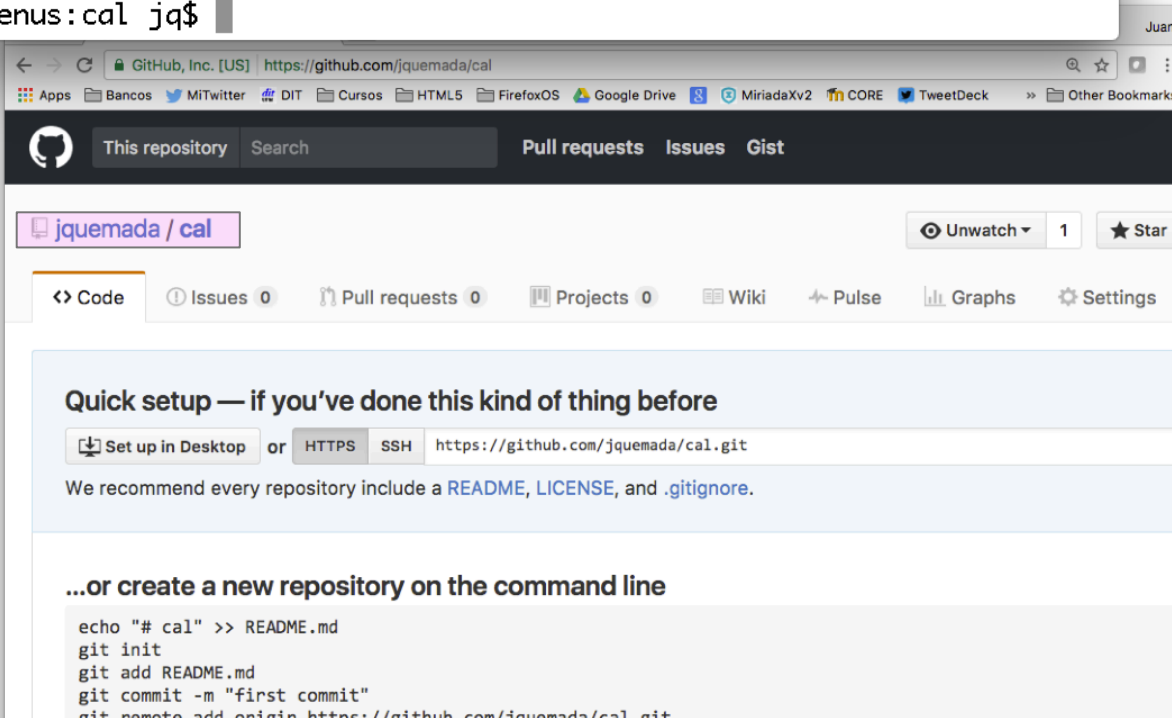
Historia del repositorio local



```
cal — -bash — 67x15
venus:cal jq$
venus:cal jq$ git log --oneline
b0e63ad x^2 button
1096247 Readme & License
venus:cal jq$
venus:cal jq$ git push https://github.com/jquemada/cal master
Counting objects: 7, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 1.40 KiB | 0 bytes/s, done.
Total 7 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/jquemada/cal
 * [new branch]      master -> master
venus:cal jq$
```



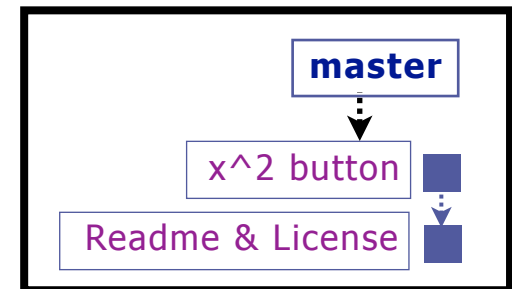
git log --oneline muestra los commits de la rama master del repositorio local **cal**.



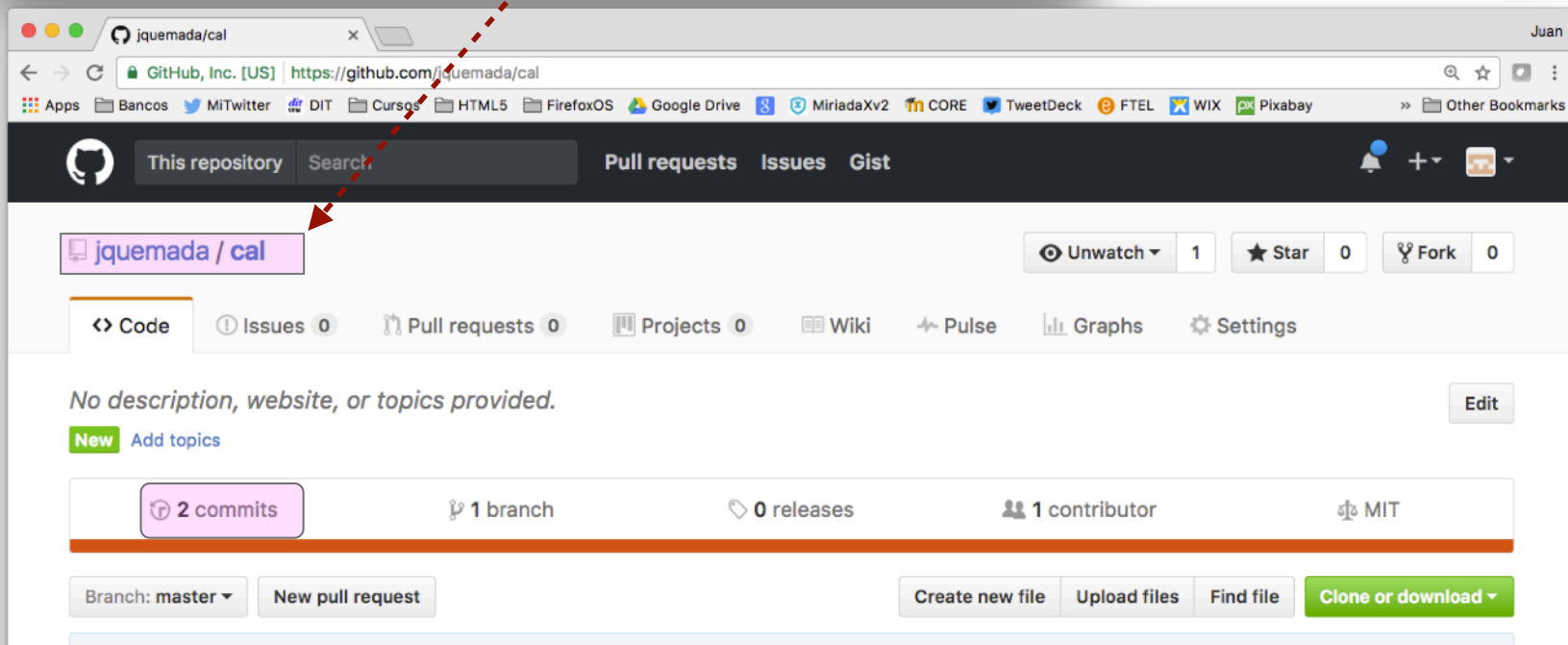
Sincronizar rama master remota con la local



```
venus:cal jq$
venus:cal jq$ git log --oneline
b0e63ad x^2 button
1096247 Readme & License
venus:cal jq$
venus:cal jq$ git push https://github.com/jquemada/cal master
Counting objects: 7, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 1.40 KiB | 0 bytes/s, done.
Total 7 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/jquemada/cal
 * [new branch]      master -> master
venus:cal jq$
```



Sube la rama **master** del repositorio **local** al repositorio **remoto** en GitHub con URL: <https://github.com/jquemada/cal> sincronizando el contenido de ambos repositorios.



El repositorio en GitHub

Repositorio en GitHub I

Es un repositorio público accesible con el URL <https://github.com/jquemada/cal> a cualquier persona través de Internet.

Clonar o descargar el repositorio en nuestro ordenador:
-> crear un **repositorio local**.

Hay 2 commits (versiones)

El proyecto: último commit de la rama master con los 3 ficheros indicados.

Fichero README.md se ve aquí. Es muy conveniente incluirlo en un fichero en GitHub describiendo el proyecto o repositorio.

cal

Educational Git project. Creates a simple calculator in HTML and JavaScript in short steps.

Repositorio en GitHub II

jquemada / cal

Code

Issues 0

Pull requests 0

Projects

No description, website, or topics provided.

New Add topics

2 commits

Último commit de la rama master con los 3 ficheros indicados.

Branch: master

New pull request

jquemada x^2 button

LICENSE

README.md

calculator.html

README.md

cal

Educational Git project. Creates a simple calculator in HTML

© Juan



This repository

Search

Pull requests

jquemada / cal

Code

Issues 0

Pull requests 0

Projects 0

Branch: master

cal / calculator.html

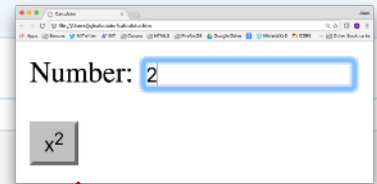
jquemada x^2 button

1 contributor

18 lines (15 sloc)

350 Bytes

```
1 <!DOCTYPE html><html><head>
2 <title>Calculator</title><meta charset="utf-8">
3 <script type="text/javascript">
4
5 function square() {
6     var num = document.getElementById("n1");
7     num.value = num.value * num.value;
8 }
9 </script>
10 </head>
11 <body>
12     Number:
13     <input type="text" id="n1"><p>
14
15     <button onclick="square()"> x<sup>2</sup> </button>
16 </body>
17 </html>
```



Repositorio en GitHub III

jqemada / cal

<> Code ! Issues 0 Pull requests 0 Projects 0

No description, website, or topics provided.

New Add topics

2 commits

1 branch

Branch: master New pull request

- jqemada x^2 button
- LICENSE
- README.md
- calculator.html

README.md

cal

Educational Git project. Creates a simple calculator in

Verde: código añadido

No hay rojo, ni negro porque al ser un fichero nuevo solo se añade.

Branch: master

Commits on Feb 12, 2017

x^2 button
jqemada committed 3 hours ago

Commits on Feb 9, 2017

Readme & License
jqemada committed 3 days ago

Comparing

GitHub, Inc.

x^2 button
master

jqemada committed an hour ago

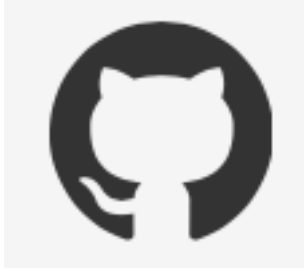
Showing 1 changed file with 17 additions and 0 deletions.

```
17 calculator.html
... @@ -0,0 +1,17 @@
1  +<!DOCTYPE html><html><head>
2  +<title>Calculator</title><meta charset="utf-8">
3  +<script type="text/javascript">
4  +
5  +function square() {
6  +  var num = document.getElementById("n1");
7  +  num.value = num.value * num.value;
8  +}
9  +</script>
10 +</head>
11 +<body>
12 +  Number:
13 +  <input type="text" id="n1"><p>
14 +
15 +  <button onclick="square()"> x<sup>2</sup> </button>
16 +</body>
17 +</html>
```

0 comments on commit 0e9f90a

Number: 2

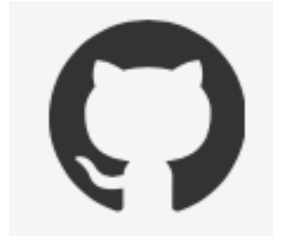
x²



Git y GitHub

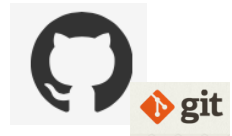
Crear commit inicial en GitHub y clonar:
new_repository, clone, remote y push

Funciones principales de GitHub



- ◆ La función principal de **GitHub** es **compartir** repositorios con terceros
- ◆ Las operaciones principales de un **usuario registrado** son
 - **Crear repositorio remoto** inicial nuevo para albergar un proyecto
 - Utilizando el botón: **New repository**
 - Se pueden añadir los ficheros: **README.md**, **LICENSE** o **.gitignore**
 - **Realiza copia** de un repositorio albergado en GitHub (para contribuir)
 - Utilizando el botón: **Fork**
 - **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
- ◆ Permite operaciones Git de **sincronización de repositorios**
 - **push** (subir rama), **clone** (clonar repositorio), **fetch** (traer rama), **pull** ..

Commit inicial en GitHub

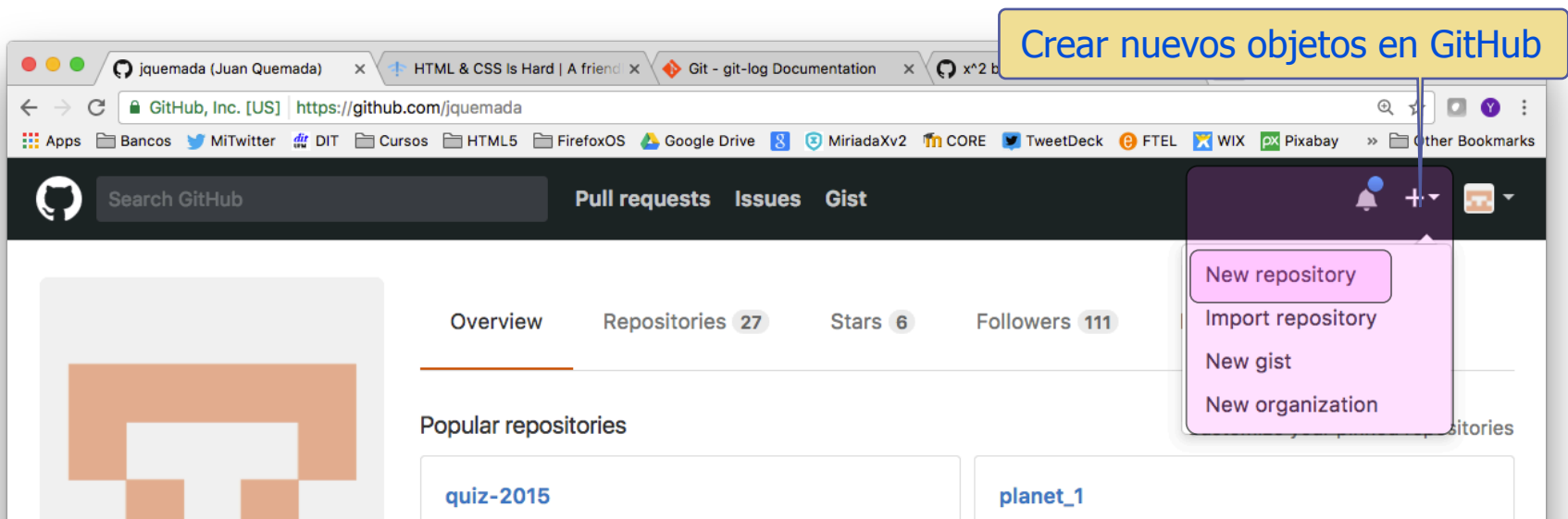


◆ GitHub permite crear el repositorio remoto con un commit inicial

- Es 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
- Esta forma de arrancar un proyecto se ilustra con el repositorio en GitHub
 - https://github.com/jquemada/cal_1

◆ Este repositorio se crea también con el botón: **New repository**

- GitHub lo crea en sus servidores invocando: **git init --bare ...**
 - Además crea automáticamente el primer commit con los ficheros que seleccionemos



Crear commit inicial en GitHub

The image shows a screenshot of the GitHub 'Create a new repository' page. The browser tabs include 'jquemada/cal_inic: Educational', 'Create a New Repository', and 'Juan Quemada, Inc. [US]'. The URL is 'https://github.com/new'. The page has a dark header with 'Pull requests', 'Issues', and 'Gist' links. A dropdown menu is open, showing options: 'New repository', 'Import repository', 'New gist', and 'New organization'. The 'New repository' option is selected. The main form has the following fields and options:

- Owner:** 'jquemada' (selected)
- Repository name:** 'cal_inic' (with a green checkmark)
- Description (optional):** 'Educational Git project. Creates a simple calculator in HTML and JavaScript in sh'
- Visibility:** 'Public' (selected), with a note 'Anyone can see this repository. You choose who can commit to this repository.'
- Initialize this repository with a README:** Checked, with a note 'This will let you immediately clone the repository to your computer.'
- Add .gitignore:** 'None' (selected)
- Add a license:** 'MIT License' (selected)
- Create repository:** A green button at the bottom.

Annotations in yellow boxes point to specific elements:

- 'Crear nuevo repositorio GitHub' points to the 'New repository' dropdown option.
- 'Nombre del repositorio' points to the 'Repository name' field.
- 'Description del repositorio' points to the 'Description (optional)' field.
- 'Incluir README.md' points to the 'Initialize this repository with a README' checkbox.
- 'Seleccionar licencia' points to the 'Add a license' dropdown.
- 'Incluir .gitignore (no incluido)' points to the 'Add .gitignore' dropdown.
- 'Crear el repositorio' points to the 'Create repository' button.

On the left side of the screenshot, there is a section for 'Popular repositories' with a list of repositories: 'quiz-2015', 'planet_1', 'planet', 'random', 'planet2010', and 'rails'. At the bottom left, there is a '137 contributions in the last year' chart showing a calendar grid from February to September.

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.

Educational Git project. Create

New Add topics

1 commit

Commits on Nov 18, 2016



Initial commit

jquemada committed 17 seconds ago

Branch: master

New

Hay 1 commit creado

jquemada Initial commit

LICENSE

README.md

README.md

cal_I

Educational Git project. Creates a simple calculator in HTML and JavaS

Este commit incluye 2 ficheros:
- LICENSE
- README.md

jquemada / cal_I

Code

Issues 0

Pull requests 0

Projects 0

Branch: master

cal_I / LICENSE

jquemada Initial commit

1 contributor

22 lines (17 sloc) 1.04 KB

```
1 MIT License
2
3 Copyright (c) 2017 Juan Quemada
4
5 Permission is hereby granted, free of charge, to any person ob
6 of this software and associated documentation files (the "Soft
7 in the Software without restriction, including without limitat
8 to use, copy, modify, merge, publish, distribute, sublicense,
9 copies of the Software, and to permit persons to whom the Soft
10 furnished to do so, subject to the following conditions:
11
12 The above copyright notice and this permission notice shall be
```


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_1
- **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 repositorio remoto compartido

- Puede copiarse al ordenador local con **git clone ...** para desarrollar en sus ramas
 - Los nuevos commits desarrollados en local pueden subirse al repositorio remoto con **git push ...**

◆ **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**
 - 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

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

```
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$
```

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

master

Initial commit

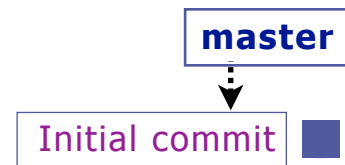
ls muestra que el directorio de trabajo del repositorio contiene los ficheros **README.md** y **LICENSE**.

Generar nuevo commit en el repositorio clonado

Directorio de trabajo limpio

```
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 calculadora.html

```
venus:cal_I jq$  
venus:cal_I jq$ git status -s  
venus:cal_I jq$ # Edit/create calculator.html  
venus:cal_I jq$  
venus:cal_I jq$ ls  
LICENSE      README.md  
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$ 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$
```

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

calculator.html

calculator.html no está registrada.

```
<!DOCTYPE html><html>  
<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.

master

Initial commit

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
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.

calculator.html

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

master

x^2 button

Initial commit

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

Actualizar el repositorio origin

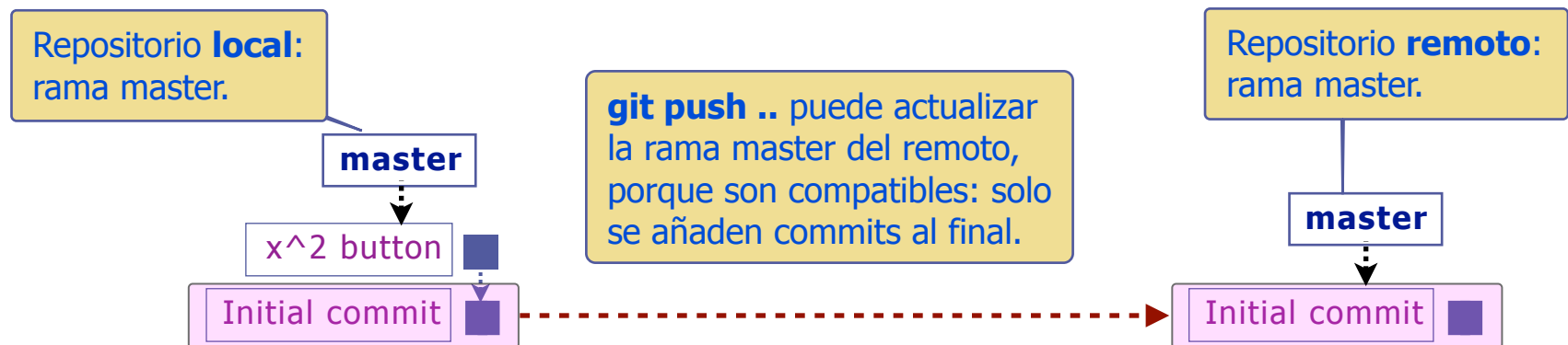
Actualizar un repositorio en GitHub con push

◆ 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**.

master

x^2 button

Initial commit

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

Mostrar URL asociado a origen

```
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**.

master

Initial commit

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**.

master

x^2 button

Initial commit

Rama master del repositorio **remoto**.

master

Initial commit

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**!

jqemada / cal_I

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

Educational Git project. Creates a simple calculator in HTML and JavaScript in short steps.

New Add topics

2 commits 1 branch 0 releases 1 contributor MIT

Branch: master New pull request

Create new file Upload files Find file Clone or download



Final del tema