

GIT CÓMO USARLO

Lo primero que vamos a tener que hacer es instalar git esto es muy sencillo nos vamos al terminal y ejecutamos el siguiente comando:

- ***sudo apt-get install git***

Una vez que tengamos git instalado que podemos verificarlo con el siguiente comando:

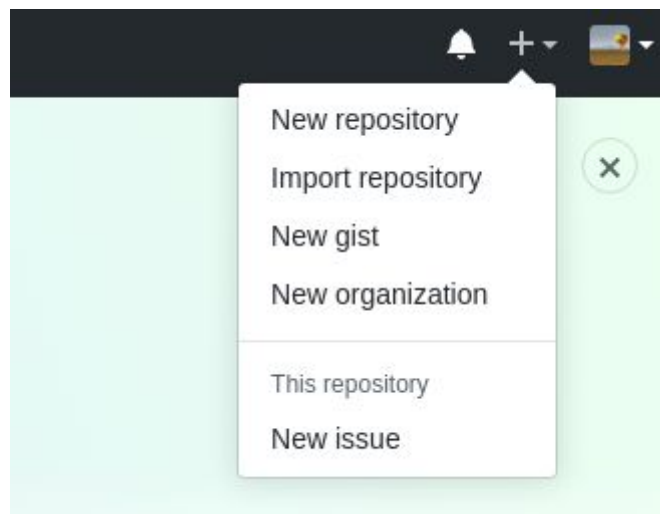
- ***apt list installed git***

Ahora podemos empezar a probar cómo usarlo lo primero que voy a intentar es subir unos cuantos scripts de bash que tengo por aquí a ver si consigo que suban a un nuevo repositorio y luego tengo acceso a ello.

Para ello en mi ordenador local voy a crear la carpeta scripts y ahí colocaré todos los códigos de bash que quiero subir a mi repositorio de github

Lo primero que voy a hacer es crear un nuevo repositorio en github con el nombre que queramos usar sin crear ni el readme ni licencia.



Por tanto pinchamos en **New repository**.



Ahora le damos el nombre que queramos al repositorio y elegimos si es público o privado, y nada más dejamos el repositorio sin licencia ni readme.


Y le damos a crear repositorio si hacemos esto nos saldrán unos pasos a seguir para desde local y línea de comandos podemos subir nuestro proyecto.


Owner **Repository name ***

 **kessusa** / 

Great repository names Your new repository will be created as **Nombre-que-queramos**. **cal-fortnight?**


Description (optional)

☒  **Public**
Anyone can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.


☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer.

Add .gitignore: **None** ▾ | Add a license: **None** ▾ 

Create repository

El código sería el siguiente


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

or **HTTPS** **SSH** 

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

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

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



Por tanto primero creo la carpeta .

- ***mkdir scripts***

Después programo mis scripts en bash ya he comentado que son muy sencillos es simplemente para poder probar como funciona git para subir archivos a un repositorio de github.

Cuando ya tengamos todo lo que queremos subir nos vamos a la carpeta en la que tengamos el proyecto en mi caso en scripts.

- ***cd scripts***

Lo primero es inicializar el repositorio.

- ***git init***

Después elegimos que añadimos al repositorio en este caso si queremos añadir todo el contenido de la carpeta scripts usaremos el comando

- ***git add .***

La opción del punto nos añadirá todo si queremos solo añadir algunas cosas podemos poner directamente detrás el nombre de cada archivo que queramos añadir al repositorio.

Después le añadimos un comentario a los archivos que vamos a subir como primera versión, o en la que estemos del proyecto.

- ***git commit -m "1 versión"***

Una vez que los hemos nombrado pondremos la url de la dirección del repositorio que hemos creado en github.

- ***git remote add origin <https://www.github.com/kessusa/figuras.git>***

Por último tenemos que hacer el push de los archivos al repositorio.

- ***git push -u origin master***

Tras esto nos pedirá tanto el usuario como la contraseña de github.

En la siguiente foto se puede ver toda la secuencia al completo, para el repositorio figuras.git es scripts de bash de la carpeta scripts.

```
(base) ambite@ambite-desktop:~/scripts$ git init
Reinitialized existing Git repository in /home/ambite/scripts/.git/
(base) ambite@ambite-desktop:~/scripts$ git add .
(base) ambite@ambite-desktop:~/scripts$ git commit -m "version_1"
On branch master
nothing to commit, working tree clean
(base) ambite@ambite-desktop:~/scripts$ git remote add origin https://github.com/kessusa/figuras.git
fatal: remote origin already exists.
(base) ambite@ambite-desktop:~/scripts$ git push -u origin master
Username for 'https://github.com': kessusa
Password for 'https://kessusa@github.com':
Counting objects: 8, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 1.37 KiB | 1.37 MiB/s, done.
Total 8 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/kessusa/figuras.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
(base) ambite@ambite-desktop:~/scripts$
```

Y ahora si nos vamos al repositorio desde github podremos ver que esta todo subido correctamente.

The screenshot shows the GitHub repository page for 'kessusa / figuras'. At the top, there are buttons for 'Unwatch', 'Star' (0), and 'Fork' (0). Below this is a navigation bar with links for 'Code', 'Issues' (0), 'Pull requests' (0), 'Actions', 'Projects' (0), 'Wiki', 'Security' (0), 'Insights', and 'Settings'. A message states 'No description, website, or topics provided.' with an 'Edit' button. Below this, a summary bar shows '1 commit', '1 branch', '0 packages', '0 releases', and '1 contributor'. The main section displays the 'master' branch with a 'New pull request' button. A table lists the files in the repository:

File	Version	Time
forpiramide.sh	1version	33 minutes ago
forrombo.sh	1version	33 minutes ago
forárbol.sh	1version	33 minutes ago
piramide.sh	1version	33 minutes ago
rombo.sh	1version	33 minutes ago
árbol.sh	1version	33 minutes ago

At the bottom, there is a prompt to 'Add a README' to help people understand the project.

Como podemos ver ya tendríamos el branch principal o master ahora voy a probar a hacer unos cambios en el código y añadir algún fichero y subirlos como un nuevo branch, no como el principal.

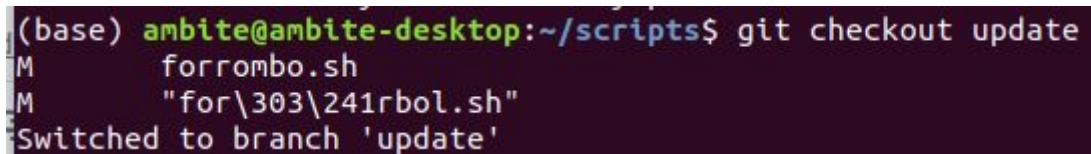
Para ello he modificado el código de dos varios de los antiguos ficheros y he añadido alguno más a la carpeta scripts.

Una vez que ya tengo los cambios y los nuevos ficheros que quiero guardar, creo el nuevo branch con los siguientes comandos.

- ***git branch "nombre que queramos para el branch"***

Lo siguiente es hacer una comprobación de los cambios con el siguiente comando.

- ***git checkout "nombre del branch"***



```
(base) ambite@ambite-desktop:~/scripts$ git checkout update
M       forrombo.sh
M       "for\303\241rbol.sh"
Switched to branch 'update'
```

Como vemos si encuentra cambios con respecto al branch anterior aquí te los notificará.

Lo siguiente que debemos hacer es añadir los ficheros nuevos que hemos creado el comando es el mismo de la otra vez para añadir.

- ***git add "nombres de los ficheros"***

Les podemos poner un comentario como la otra vez.

- ***git commit -m "comentario"***

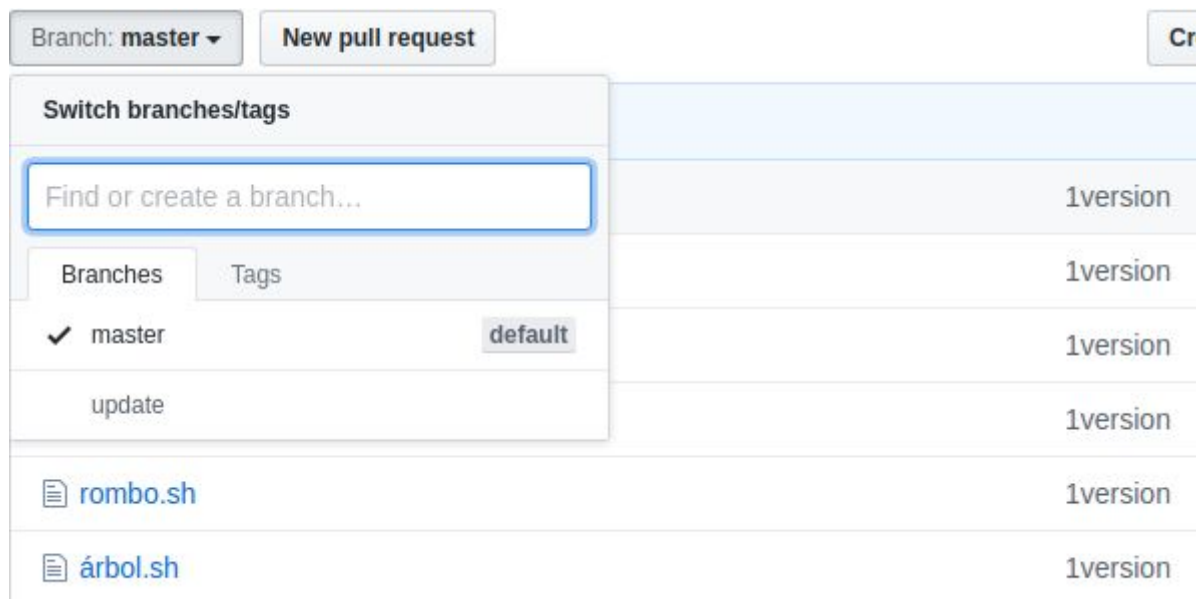
Por último los subimos al repositorio con el siguiente comando.

- ***git push --set-upstream origin "nombre del nuevo branch"***

La secuencia de comandos completa con sus resultados en mi caso la podéis ver en la siguiente fotografía.

```
(base) ambite@ambite-desktop:~/scripts$ git branch update
(base) ambite@ambite-desktop:~/scripts$ git che
checkout      cherry      cherry-pick
(base) ambite@ambite-desktop:~/scripts$ git checkout update
M       forrombo.sh
M       "for\303\241rbol.sh"
Switched to branch 'update'
(base) ambite@ambite-desktop:~/scripts$ ls
árbol.sh      dialogousuarios.sh  forpiramide.sh  piramide.sh  tablero.sh
descifrador.sh  forárbol.sh        forrombo.sh     rombo.sh
(base) ambite@ambite-desktop:~/scripts$ git add dialogousuarios.sh descifrador.sh tablero.sh
(base) ambite@ambite-desktop:~/scripts$ git commit -m "2version"
[update e83ddab] 2version
3 files changed, 100 insertions(+)
create mode 100755 descifrador.sh
create mode 100755 dialogousuarios.sh
create mode 100755 tablero.sh
(base) ambite@ambite-desktop:~/scripts$ git push --set-upstream origin update
Username for 'https://github.com': kessusa
Password for 'https://kessusa@github.com':
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.59 KiB | 1.59 MiB/s, done.
Total 5 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'update' on GitHub by visiting:
remote:   https://github.com/kessusa/figuras/pull/new/update
remote:
To https://github.com/kessusa/figuras.git
 * [new branch]      update -> update
Branch 'update' set up to track remote branch 'update' from 'origin'.
```

Si nos vamos a github veremos como tenemos el nuevo branch ya disponible.



Vemos como aparece para seleccionarlo y si lo hacemos tendremos disponibles los nuevos fichero y los cambios de los antiguos.

Branch: **update** New pull request Create new file Upload files Find file Clone

This branch is 1 commit ahead of master. Pull request

File	Version	Latest commit
descifrador.sh	2version	13
dialogousuarios.sh	2version	13
forpiramide.sh	1version	1
forrombo.sh	1version	1
forárbol.sh	1version	1
piramide.sh	1version	1
rombo.sh	1version	1
tablero.sh	2version	13
árbol.sh	1version	1

Help people interested in this repository understand your project by adding a README. Add

Y una vez que ya lo tenemos subido podemos hacer un pull request para ver cambios en el código entre los branch por ejemplo.

Update #3

Open kessusa wants to merge 3 commits into **master** from **update** Edit

Conversation 0 Commits 3 Checks 0 Files changed 5 +102 -2

Changes from 1 commit File filter... Clear filters Jump to... Review changes

2version < Prev Next >

update (#3)

kessusa committed 3 minutes ago commit 5ff7cb49145bf9674786ed269f3d3df1568cb55a

2 forrombo.sh ...

```

...    ...    @@ -1,5 +1,5 @@
1      1      #!/bin/bash
2      2      - echo "Vamos a hacer una pirámide,dime de cuantos pisos la hacemos"
3      3      + echo "Dime la semialtura del rombo que vamos a dibujar"
4      4      read Pisos
5      5      for((piso=1; piso<=Pisos; piso++))
        do

```


Vemos como nos marca la línea de código borrada y la que acabamos de poner en este caso es una tontería pero lo he hecho para ver cómo funciona.


Por ejemplo con la primera versión de mi branch aparecerán todos los nuevos códigos añadidos.

Podemos ver que todo está en verde porque son nuevos scripts que nos existían en el branch master original.

2version

< PrevNext >

 update (#3)

 kessusa committed 26 minutes agocommit e83ddabbd207ad3afbfc844d906428514472ae2

14 descifrador.sh

... ... @@ -0,0 +1,14 @@

```
1 +#!/bin/bash
2 + echo "hola chicos vamos a jugar"
3 + read -p "dime el hash MD5 que quiere romper" clave
4 + read -p "dime el diccionario que vamos a usar" diccionario
5 + while read linea
6 + do
7 + hash=$(echo -n $linea | md5sum)
8 +
9 + if [ "$clave" == "$hash" ];
10 + then
11 + echo "esta es la clave que buscas $linea"
12 + break
13 + fi
14 + done < $diccionario
```

67 dialogousuarios.sh

... ... @@ -0,0 +1,67 @@

```
1 +
2 +#!/bin/bash
3 + #CREADOR ÁLVARO SÁNCHEZ AMBITE
4 + #FECHA 13/01/2020
5 + TMPFCMD="/tmp/cmd.tmp"
6 + condicion=0
7 + while [ $condicion = 0 ]; do
8 + dialog --title 'Menú' --menu 'Elija una opción:' 0 0 0 \
9 + 1 'Número de usuarios' \
10 + 2 'Número de grupos' \
11 + 3 'Usuarios que tienen bash' \
```

Como hemos podido comprobar es muy cómoda la gestión de repositorios por comandos,haciendo muy sencilla la sincronización del proyecto local con nuestro repositorio en github.