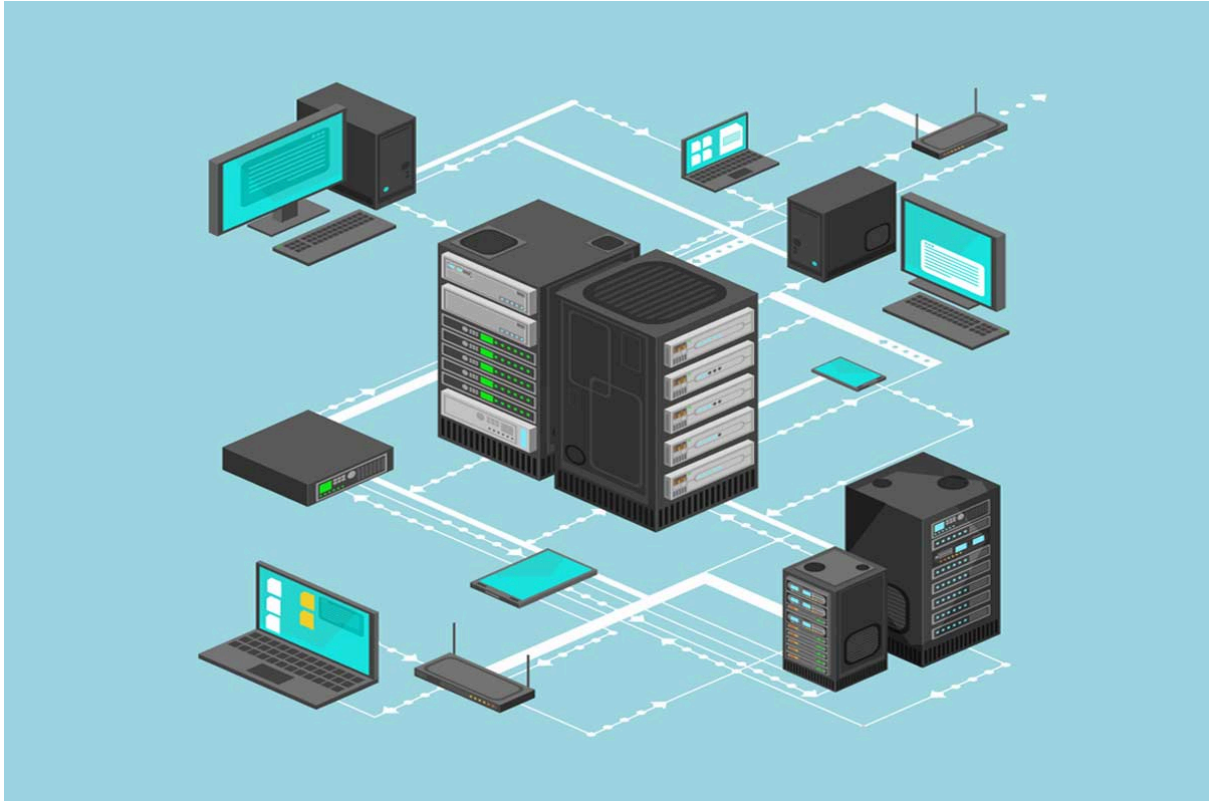


Comandos de git



José gutiérrez López

Índice

[Comandos de GIT](#)

[Paso 1: Configurar git una vez instalado en un pc.](#)

[Paso 2: Crear un directorio](#)

[Paso 3: Creación de fichero](#)

[Diferentes comandos](#)

[Clave ssh en git hub](#)

Comandos de GIT

Paso 1: Configurar git una vez instalado en un pc.

Lo primero que tenemos que hacer es poner este comando que lo que hace es que tu usuario se llame como el nombre que pongas entre comillas.

```
git config --global user.name "jose"
```

Luego tenemos que poner el siguiente comando para asociar un correo con nuestro nombre.

```
git config --global user.email "jose.gutierrez@salesianoslosboscos.com"
```

Paso 2: Crear un directorio

Lo primero que tenemos que hacer es crear un directorio donde nosotros queramos y luego tenemos que entrar desde git poniendo la ruta. Una vez estemos dentro del directorio tenemos que ejecutar el comando git int.

```
git init
```

Luego podemos utilizar este comando para aplicarle un nombre a nuestro directorio.

```
git branch -m main
```

```
PC15@PC15Jose MINGW64 /c/smr_git/rep_1 (main)
```

Paso 3: Creación de fichero

Luego lo que tenemos que hacer es crear un fichero de un html por ejemplo en la carpeta y luego tenemos que aplicar el siguiente comando.

```
git add Ejercicio1.html
```

Hemos puesto este comando porque solo tenemos un archivo pero si hemos hecho varios y queremos coger varios podemos utilizar uno “.”

Luego utilizamos este comando para ver el estado de los repositorios.

```
git status
```

José gutiérrez López

Y luego ponemos el siguiente comando que sirve para meter un nombre a nuestro commit.

```
git commit -m "Este es mi primer comit"
```

Luego ponemos el siguiente comando que sirve para ver todos nuestro commit y sus modificaciones.

```
git log
```

José gutiérrez López

Diferentes comandos

Este comando sirve para ver todos los cambios que has hecho en ese commit.

```
$ git show
commit 6770cda6ce7b16bad742ef558ab6fe1080118165 (HEAD -> prueba)
Author: jose <jose.gutierrez@salesianoslosboscos.com>
Date: Mon Dec 9 08:42:08 2024 +0100

    eliminacion del archivo prueba

diff --git a/prueba_de_git.txt b/prueba_de_git.txt
deleted file mode 100644
index 8b46a46..0000000
--- a/prueba_de_git.txt
+++ /dev/null
@@ -1,2 +0,0 @@
-hola que tal estas
-
```

Este comando sirve para ver todos los cambios que has hecho hasta este momento.

```
PC15@PC15Jose MINGW64 /c/smr_git/rep_2 (prueba)
$ git diff
diff --git a/hola.txt b/hola.txt
index aaa5c2c..d720fe3 100644
--- a/hola.txt
+++ b/hola.txt
@@ -1 +1 @@
-sino
+sin pan
\ No newline at end of file
```

Este comando sirve para ver todos los ficheros que hay dentro de un commit.

```
$ git ls-tree --name-only 6770cda6ce7b16bad742ef558ab6fe1080118165
hola.txt
```

Este comando sirve para darle un alias a un comando (se suele utilizar en comandos largos)

```
PC15@PC15Jose MINGW64 /c/smr_git/rep_2 (prueba)
$ git config --global alias.tree "log --graph --decorate --all --oneline"

PC15@PC15Jose MINGW64 /c/smr_git/rep_2 (prueba)
$ git tree
* 6770cda (HEAD -> prueba) eliminacion del archivo prueba
* a3a829b creacion de nuevo archivo hola y modificacion de archivo prueba
* 4077726 commit de prueba
```

José gutiérrez López

Este comando sirve para revertir comandos que hayas hecho siempre y cuando no los hayas subido. (puedes utilizar el nombre de un fichero o un . para coger todos los puntos)

```
PC15@PC15Jose MINGW64 /c/smr_git/rep_2 (prueba)
$ git checkout .
```

Si metemos este comando más el hash de un commit lo que nos hace es movernos a se commit que le hemos dicho. Es lo podemos ver con el (HEAD) esas letras que están en azul nos indica en el commit que estamos en este momento. (recordar que el comando “git tree” no existe es un alias).

```
PC15@PC15Jose MINGW64 /c/smr_git/rep_2 (prueba)
$ git checkout abfce00
Note: switching to 'abfce00'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at abfce00 Subir archivo para ignorar
PC15@PC15Jose MINGW64 /c/smr_git/rep_2 ((abfce00...))
$ git tree
* 8d8e950 (prueba) Subir fichero para ignorar arhcivos y modificacion de diferentes ficheros
* abfce00 (HEAD) Subir archivo para ignorar
* 087b34b Añadir texto al archivo
* e10dfa1 Dejando como el commit anterior
* e53269c Subida de dos ficheros dejando uno sin commitear
* 6770cda eliminacion del archivo prueba
* a3a829b creacion de nuevo archivo hola y modificacion de archivo prueba
* 4077726 commit de prueba
```

José gutiérrez López

Este comando sirve para volver a un commit anterior, es decir cuando metes el comando tienes que poner el commit al que quieres volver no eliminar.

Commit el cual
quiera eliminar

```
PC15@PC15Jose MINGW64 /c/smr_git/rep_2 (prueba)
$ git tree
f5551b4 (HEAD -> prueba) Commit para probar a eliminar un commit
8d8e950 Subir fichero para ignorar archivos y modificacion de diferentes ficheros
* abfce00 Subir archivo para ignorar
* 087b34b Añadir texto al archivo
* e10dfa1 Dejando como el commit anterior
* e53269c Subida de dos ficheros dejando uno sin commitear
* 6770cda eliminacion del archivo prueba
* a3a829b creacion de nuevo archivo hola y modificacion de archivo prueba
* 4077726 commit de prueba

PC15@PC15Jose MINGW64 /c/smr_git/rep_2 (prueba)
$ git reset --hard 8d8e950 ← Comando
HEAD is now at 8d8e950 Subir fichero para ignorar archivos y modificacion de diferentes ficheros

PC15@PC15Jose MINGW64 /c/smr_git/rep_2 (prueba)
$ git tree
8d8e950 (HEAD -> prueba) Subir fichero para ignorar archivos y modificacion de diferentes ficheros
* abfce00 Subir archivo para ignorar
* 087b34b Añadir texto al archivo
* e10dfa1 Dejando como el commit anterior
* e53269c Subida de dos ficheros dejando uno sin commitear
* 6770cda eliminacion del archivo prueba
* a3a829b creacion de nuevo archivo hola y modificacion de archivo prueba
* 4077726 commit de prueba
```

Commit al cual
he vuelto

Este comando sirve para añadir tags a los commit, lo añade al commit en el que estés en ese momento.

```
PC15@PC15Jose MINGW64 /c/smr_git/rep_2 (prueba)
$ git tag V3.1
```

```
PC15@PC15Jose MINGW64 /c/smr_git/rep_2 (prueba)
$ git tree
8d8e950 (HEAD -> prueba, tag: V3.1) Subir fichero para ignorar archivos y modificacion de diferentes ficheros
* abfce00 Subir archivo para ignorar
* 087b34b Añadir texto al archivo
```

Este comando sirve para eliminar un tag, no hace falta estar en el commit en el cual está puesto este tag.

```
PC15@PC15Jose MINGW64 /c/smr_git/rep_2 (prueba)
$ git tag --delete V3.0
Deleted tag 'V3.0' (was abfce00)
```

José gutiérrez López

Este comando sirve para agregar una rama en el commit que estemos "login" es el nombre de la rama.

```
PC15@PC15Jose MINGW64 /c/smr_git/rep_2 (prueba)
$ git branch login
```

Este comando sirve para entrar a la rama.

```
PC15@PC15Jose MINGW64 /c/smr_git/rep_2 (prueba)
$ git switch login
Switched to branch 'login'
```

Este comando sirve para fusionar el contenido de la rama y el contenido del main.

```
PC15@PC15Jose MINGW64 /c/smr_git/rep_2 (login)
$ git merge prueba
```

Es una instantánea en local. Parecido a un status.

```
PC15@PC15Jose MINGW64 /c/smr_git/hello_git (smr)
$ git stash
```

Sirve para ver los stash

```
PC15@PC15Jose MINGW64 /c/smr_git/hello_git (smr)
$ git stash list
```

Sirve para volver a editar lo que has dejado en el stash.

```
PC15@PC15Jose MINGW64 /c/smr_git/hello_git (smr)
$ git stash pop
```

Este comando sirve para bajar lo que tienes en github sin mergear.

```
PC15@PC15Jose MINGW64 /c/smr_git/t (main)
$ git fetch
```

Este comando sirve para bajar lo que tienes en github pero ya mergeado.

```
PC15@PC15Jose MINGW64 /c/smr_git/t (main)
$ git pull
```


José gutiérrez López

Este comando sirve subir a la nueva de github.

```
PC15@PC15Jose MINGW64 /c/smr_git/t (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 287 bytes | 287.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Guti17903/smr_pract.git
a03ebd7..3e4ff7f main -> main
```

clona la carpeta que nos descargamos desde github

```
PC18@hp-elitebook-850-g6 MINGW64 /c/smr_git/clonado
$ git clone git@github.com:hugooo23/smr_pract.git
Cloning into 'smr_pract'...
remote: Enumerating objects: 30, done.
remote: Counting objects: 100% (30/30), done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 30 (delta 12), reused 20 (delta 8), pack-reused 0 (from 0)
Receiving objects: 100% (30/30), 4.46 KiB | 652.00 KiB/s, done.
Resolving deltas: 100% (12/12), done.

PC18@hp-elitebook-850-g6 MINGW64 /c/smr_git/clonado
$
```

Clave ssh en git hub

Este comando sirve para ver si tengo alguna clave ssh. (alt-126 para poner ~).

```
PC15@PC15Jose MINGW64 ~  
$ ls -a ~/.ssh
```

Comando para saber si esta el ssh levantado

```
PC15@PC15Jose MINGW64 /c/users/usuario/.ssh  
$ eval "$(ssh-agent -s)"  
Agent pid 1952
```

Comando para generar nuestras claves.

```
PC15@PC15Jose MINGW64 /c/users/usuario/.ssh  
$ ssh-keygen -t ed25519 -C "jose.gutierrez@salesianoslosboscos.com"  
Generating public/private ed25519 key pair.  
Enter file in which to save the key (/c/Users/usuario/.ssh/id_ed25519):  
Enter passphrase for "/c/Users/usuario/.ssh/id_ed25519" (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /c/Users/usuario/.ssh/id_ed25519  
Your public key has been saved in /c/Users/usuario/.ssh/id_ed25519.pub  
The key fingerprint is:  
SHA256:9Jjp7gNwHtwVChoj7dhcyh800Ashl3iANAGx9CFaToY jose.gutierrez@salesianoslosboscos.com  
The key's randomart image is:  
+--[ED25519 256]--+  
|O@O*o . . . . .|  
|E00.++.. . . .|  
|o..X.= .o. . . .|  
| . X =..= . . . .|  
| B oS . . . . .|  
| =. . . . .|  
| .. . . .|  
| .. . . .|  
| .o. . . . .|  
+-----[SHA256]-----+
```

Este comando sirve para incluir la clave. El id tiene que ser el nombre de tu clave no siempre es (id_ed25519).

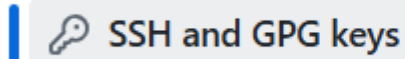
```
PC15@PC15Jose MINGW64 /c/users/usuario/.ssh  
$ ssh-add ~/.ssh/id_ed25519  
Identity added: /c/Users/usuario/.ssh/id_ed25519 (jose.gutierrez@salesianoslosboscos.com)
```

José gutiérrez López

Luego tenemos que crear un archivo sin extensión llamado config y poner el siguiente texto dentro. Y cambiar el (id_rsa) por el nombre del archivo como hicimos arriba.

```
Host github.com
  User git
  Hostname ssh.github.com
  PreferredAuthentications publickey
  IdentityFile ~/.ssh/id_rsa
  Port 443
```

Luego tenemos que ir a settings o configuración en github y luego a ssh keys



Una vez hay le damos a key. Y rellenamos esto con el título que queramos y en key tenemos que poner lo que hay dentro del archivo .pub que nos han creado. Suele ser algo parecido a esto (

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIi87Btw4YR1XdF/PTCidHoMICGzVj0Hai0JEf1xcxtid jose.gutierrez@salesianoslosboscos.com
```

)

Title

Key type

Authentication Key ↕

Key

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

Add SSH key

Bifurcación repositorio

Lo primero que tenemos que hacer es darle a fork que está arriba a la derecha.

Y una vez estamos aqui le damos a crear fork.

Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Required fields are marked with an asterisk ().*

Owner *

 Guti17903 ▾

Repository name *

/ smr_pract_07


✔ smr_pract_07 is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

☒ Copy the `main` branch only

Contribute back to ilopez83/smr_pract_07 by adding your own branch. [Learn more.](#)

 You are creating a fork in your personal account.

Create fork

José gutiérrez López

Luego tenemos que ir a code y copiar el link de SSH y luego clonarlo con un git clone y el link.

