

Instalación de Git

Ubuntu 20.04



Índice

1) Instalación.....	3
2) Configuración.....	3
3) Lista de comandos Git.....	3
4) Creación de Repositorio.....	4
5) Crear archivo.....	4
6) Subir al repositorio (Commit -m).....	5
7) Creación de repositorio.....	5
8) Empujar archivos al repositorio.....	6
9) Clonado.....	6
10) Ver cambios.....	7
11) Repositorios remotos.....	7

1) Instalación

Utilizamos apt-get update para actualizar los paquetes

```
entornos@entornos:~/Escritorio$ sudo apt-get update
Obj:1 http://es.archive.ubuntu.com/ubuntu focal InRelease
Obj:2 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease
```

Accedemos al terminar y enviamos el siguiente comando:

```
entornos@entornos:~/Escritorio$ sudo apt install git
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
git-man liberror-perl
```

Versión actual

```
entornos@entornos:~/Escritorio$ git --version
git version 2.25.1
```

2) Configuración

Configuración GIT

```
entornos@entornos:~/Escritorio$ git config --global user.email javi_lopez-calderon@hotmail.com
entornos@entornos:~/Escritorio$ git config --global user.name Javier
entornos@entornos:~/Escritorio$ git config --list
user.email=javi_lopez-calderon@hotmail.com
user.name=Javier
```

3) Lista de comandos Git

```
entornos@entornos:~/Escritorio$ git --help
uso: git [--version] [--help] [-C <ruta>] [-c <nombre>=<valor>]
      [--exec-path[=<ruta>]] [--html-path] [--man-path]
      [-p | --paginate | -P | --no-pager] [--no-replace-objects]
      [--git-dir=<ruta>] [--work-tree=<ruta>] [--namespace=<nombre>]
      <comando> [<args>]

Estos son comandos comunes de Git usados en varias situaciones:

comienza un área de trabajo (ver también: git help tutorial)
  clone      Clona un repositorio dentro de un nuevo directorio
  init       Crea un repositorio de Git vacío o reinicializa uno existente
```

4) Creación de Repositorio

```
entornos@entornos:~/Escritorio/Gitentornos$ git init Gitentornos
Inicializado repositorio Git vacío en /home/entornos/Escritorio/Gitentornos/Gitentornos/.git/
```

Archivo README

```
entornos@entornos:~/Escritorio/Gitentornos$ gedit README
```

Abrir ▼

***README**
~/Escritorio/Gitentornos

1 Probando GIT

5) Crear archivo

En este caso se creará un archivo .java que es el que conozco :)

```
os$ nano ejemplo.java
```

```
GNU nano 4.8 ejemplo.java
public class ejemplo {
    public static void main (String [] args) {
        System.out.println("Hola mundo");
    }
}
```

Git ADD

```
entornos@entornos:~/Escritorio/Gitentornos/Gitentornos$ git add README
entornos@entornos:~/Escritorio/Gitentornos/Gitentornos$ git add
Nada especificado, nada agregado.
¿Tal vez quiso decir 'git add .'?
```

```
entornos@entornos:~/Escritorio/Gitentornos/Gitentornos$ git add ejemplo.java
entornos@entornos:~/Escritorio/Gitentornos/Gitentornos$
```

Para deshacer y devolver de la Stage el archivo utilizar:

```
entornos@entornos:~/Escritorio/Gitentornos/Gitentornos$ git reset README
entornos@entornos:~/Escritorio/Gitentornos/Gitentornos$ git status
En la rama master

No hay commits todavía

Cambios a ser confirmados:
  (usa "git rm --cached <archivo>..." para sacar del área de stage)
    nuevo archivo: ejemplo.java

Archivos sin seguimiento:
  (usa "git add <archivo>..." para incluirlo a lo que se será confirmado)
    README

entornos@entornos:~/Escritorio/Gitentornos/Gitentornos$ git add README
```

6) Subir al repositorio (Commit -m)

```
entornos@entornos:~/Escritorio/Gitentornos/Gitentornos$ git commit -m "Primer Commit"
[master (commit-raiz) 00a9170] Primer Commit
2 files changed, 6 insertions(+)
create mode 100644 README
create mode 100644 ejemplo.java
entornos@entornos:~/Escritorio/Gitentornos/Gitentornos$ git status
En la rama master
nada para hacer commit, el árbol de trabajo está limpio
```

Se modifica el archivo ejemplo

```
entornos@entornos:~/Escritorio/Gitentornos/Gitentornos$ git status
En la rama master
Cambios no rastreados para el commit:
(usa "git add <archivo>..." para actualizar lo que será confirmado)
(usa "git restore <archivo>..." para descartar los cambios en el directorio de trabajo)
modificado:   ejemplo.java
```


7) Creación de repositorio

Creamos el repositorio

Create a new repository

A repository contains all project files, including the revision history. Already have a project? [Import a repository.](#)

Owner *

 Zharell ▾


Repository name *


Gitentornos

✓

Great repository name Gitentornos is available. [Need inspiration? How about **spec**](#)

Description (optional)

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

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

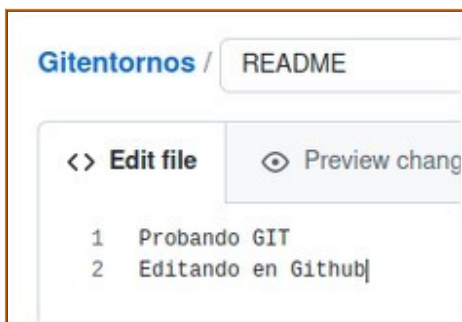
```
/Gitentornos$ git remote add origin https://github.com/Zharell/Gitentornos.git
/Gitentornos$ git branch -M main
/Gitentornos$ git push -u origin main
```


8) Empujar archivos al repositorio

Se ha decidido usar un Access Tokens ya que el log por bash ya no tiene soporte...

```
entornos@entornos:~/Escritorio/Gitentornos/Gitentornos$ git push -u origin main
Username for 'https://github.com': zharell
Password for 'https://zharell@github.com':
Rama 'main' configurada para hacer seguimiento a la rama remota 'main' de 'origin'.
Everything up-to-date
```

Comprobamos cómo se han subido los archivos a Github



Editando en Github

9) Clonado

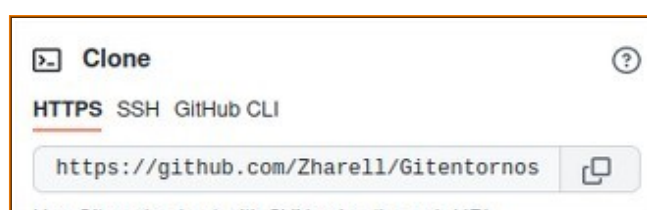
Borramos el directorio LOCAL

```
entornos@entornos:~/Escritorio/Gitentornos$ rm -R Gitentornos/
rm: ¿borrar el fichero regular 'Gitentornos/.git/objects/18/417c7'
¿borrar? (s/n) s
```

Y creamos un directorio nuevo para el clonado

```
entornos@entornos:~/Escritorio/Gitentornos$ mkdir clonadoGit
entornos@entornos:~/Escritorio/Gitentornos$ cd clonadoGit/
entornos@entornos:~/Escritorio/Gitentornos/clonadoGit$
```

Copiamos el enlace del clonado



Realizamos la clonación

```
entornos@entornos:~/Escritorio/Gitentornos/clonadoGit$ git clone https://github.com/Zharell/Gitentornos.git
Clonando en 'Gitentornos'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 7 (delta 0), reused 4 (delta 0), pack-reused 0
Desempaquetando objetos: 100% (7/7), 1021 bytes | 340.00 KiB/s, listo.
```

10) Ver cambios

Git log

```
entornos@entornos:~/Escritorio/Gitentornos/clonadoGit$ cd Gitentornos/
entornos@entornos:~/Escritorio/Gitentornos/clonadoGit/Gitentornos$ git log
commit 5ba1ab3a38ee13d044c6788498c0bb3f2380f362 (HEAD -> main, origin/main, origin/HEAD)
Author: Zharell <91702481+Zharell@users.noreply.github.com>
Date:   Wed Jan 26 21:04:22 2022 +0100

    Update README
```

11) Repositorios remotos

Nos muestra Origin que es el servidor predeterminado

```
entornos@entornos:~/Escritorio/Gitentornos/clonadoGit/Gitentornos$ git remote
origin
```

Utilizando el comando Git Fetch podemos recuperar datos de servidores remotos.

```
entornos@entornos:~/Escritorio/Gitentornos/clonadoGit/Gitentornos$ git fetch origin
entornos@entornos:~/Escritorio/Gitentornos/clonadoGit/Gitentornos$ git log
commit 5ba1ab3a38ee13d044c6788498c0bb3f2380f362 (HEAD -> main, origin/main, origin/HEAD)
Author: Zharell <91702481+Zharell@users.noreply.github.com>
Date:   Wed Jan 26 21:04:22 2022 +0100

    Update README
```