


CREACIÓN DE UN REPOSITORIO CON GIT

- Empezamos creando un nuevo repositorio en Github

Create a new repository


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

Owner *

 garciavelaismael ▾


/

Repository name *


prueba01 

Great repository names are short and memorable. Need inspiration? How about **fluffy-funicular**?

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.

Initialize this repository with:

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

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

- Ahora vamos a crear el repositorio de forma local para luego hacerle un push y subirlo a la nube. Creamos la carpeta del proyecto, creamos el archivo README.md y iniciamos el repositorio.

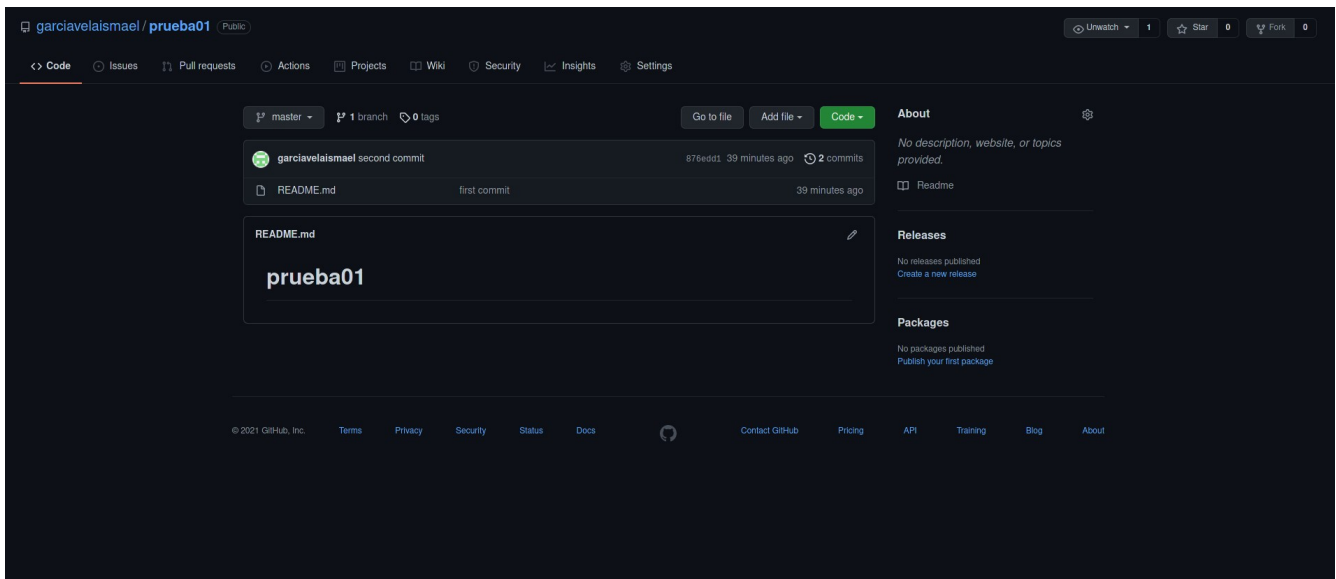
```
~/Documents/2ºASIR/SGBD > mkdir prueba01
~/Documents/2ºASIR/SGBD > cd prueba01
~/Documents/2ºASIR/SGBD/prueba01 > echo "# prueba01" >> README.md
~/Documents/2ºASIR/SGBD/prueba01 > git init
```

- Añadimos al push el README.md y le hacemos el commit, en este caso al ser la primera vez pedirá añadir tu cuenta de github por email o por username. Luego ya dejará hacer commit.

```
~/Documents/2ºASIR/SGBD/prueba01 > master ?1 > git add README.md
~/Documents/2ºASIR/SGBD/prueba01 > master +1 > git commit -m "first commit"
Author identity unknown
*** Please tell me who you are.
Run
  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"
to set your account's default identity.
Omit --global to set the identity only in this repository.
fatal: unable to auto-detect email address (got 'ismael@manjaro.(none)')
~/Documents/2ºASIR/SGBD/prueba01 > master +1 > git config --global user.email "igarvel129@g.educaand.es"
~/Documents/2ºASIR/SGBD/prueba01 > master +1 > git commit -m "first commit"
[master (root-commit) ab4a4e8] first commit
```

- Por último habrá que añadir el origen y la rama donde queremos hacer el push y al hacer push nos pedirá username y password, una vez introducido se subirá todo a nuestro repositorio de Github.

```
~/Documents/2ºASIR/SGBD/prueba01 > master > git branch -M master
~/Documents/2ºASIR/SGBD/prueba01 > master > git remote add origin https://github.com/garciavelaismael/prueba01.git
~/Documents/2ºASIR/SGBD/prueba01 > master > git push -u origin master
Username for 'https://github.com': garciavelaismael
Password for 'https://garciavelaismael@github.com':
```



- Para ver como funciona el git status añadiremos un “HOLA” a el archivo README.md. Al añadir el texto y SIN hacer add, saldrá en rojo y cuando se le haga el add saldrá en verde. Una vez le hagase el commit desaparece y pondrá que todo esta listo para push.

```
~/Documents/2ºASIR/SGBD/prueba01 > master echo "HOLA" >> README.md

~/Documents/2ºASIR/SGBD/prueba01 > master !1 cat README.md
# prueba01
HOLA

~/Documents/2ºASIR/SGBD/prueba01 > master !1 git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

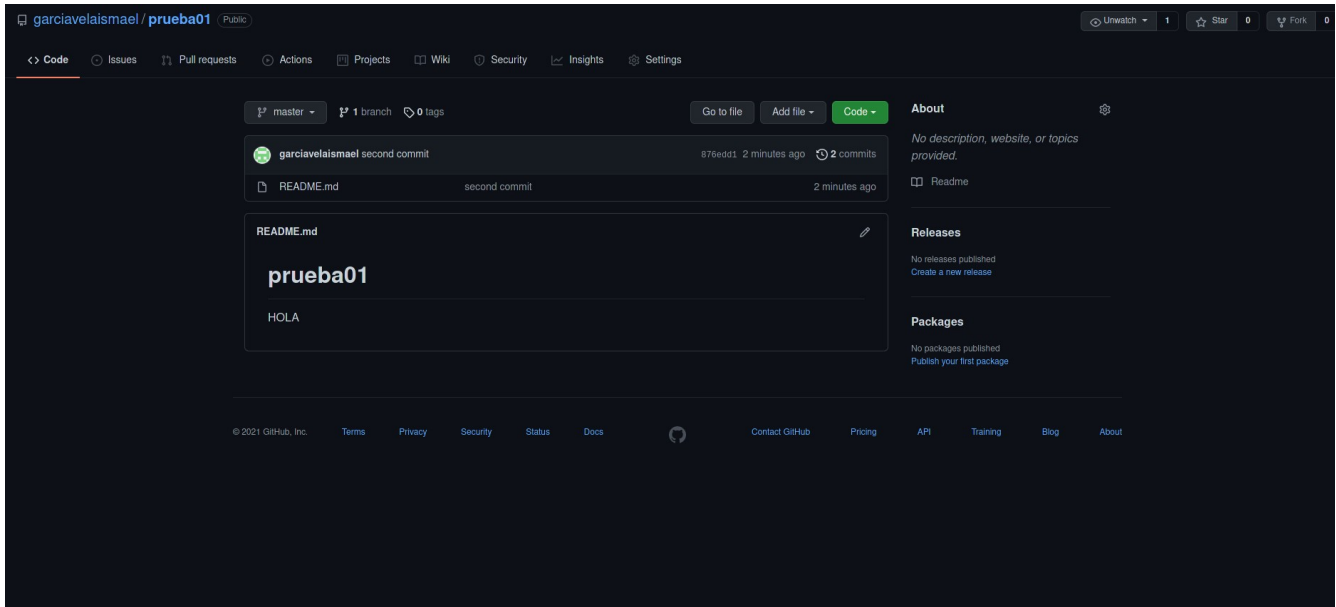
no changes added to commit (use "git add" and/or "git commit -a")
~/Documents/2ºASIR/SGBD/prueba01 > master !1 git add README.md

~/Documents/2ºASIR/SGBD/prueba01 > master +1 git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md

~/Documents/2ºASIR/SGBD/prueba01 > master +1
```

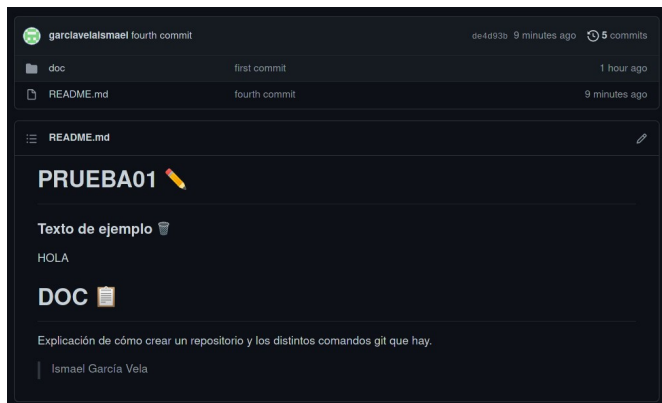
- Volvemos al repositorio después de volver a hacer el **commit** y el **push** y veremos que se ha añadido el HOLA al README.md.



- Al igual que en Windows las credenciales de inicio de sesión de Github se almacenan en el Administrador de Credenciales, en Linux se crea un archivo oculto llamado **git-credentials** en el home del usuario donde se está usando git y ahí se almacena el inicio de sesión.

```
~ : zsh — Konsole  
~ ➤ cat ~/.git-credentials  
https://garciavelismael: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@github.com
```

- Ahora vamos a dar uso al **pull**, aquí he modificado el README.md en la nube y al hacer un **push** desde el repositorio local dará **error**. Luego he hecho un pull y vemos como lo ha recogido correctamente.



```
> git push -u
To https://github.com/garciavelaismael/prueba01.git
! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'https://github.com/garciavelaismael/prueba01.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
> git pull
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), 1.45 KiB | 1.45 MiB/s, done.
From https://github.com/garciavelaismael/prueba01
  1abc5d6..de4d93b  master    -> origin/master
Updating 1abc5d6..de4d93b
Fast-forward
 README.md | 10 ++++++++--
 1 file changed, 9 insertions(+), 1 deletion(-)
```

~/Doc/2/SG/prueba01 master 09:36:46

- Para probar el **git clone** me he ido a una carpeta distinta y he hecho el comando añadiendo la dirección url. Este clona el repositorio por completo, en caso de querer hacer otro repositorio con esos mismos datos en vez de un clone habría que descargar desde Github el repositorio en formato ZIP.

```
prueba01: zsh — Konsole <2>
> git clone https://github.com/garciavelaismael/prueba01.git
Cloning into 'prueba01'...
remote: Enumerating objects: 16, done.
remote: Counting objects: 100% (16/16), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 16 (delta 1), reused 10 (delta 0), pack-reused 0
Receiving objects: 100% (16/16), 427.07 KiB | 1.75 MiB/s, done.
Resolving deltas: 100% (1/1), done.
> ls
prueba01
> cd prueba01
> ls
doc README.md
> cat README.md
# PRUEBA01 =
### Texto de ejemplo 🗑

HOLA

# DOC 📄

Explicación de cómo crear un repositorio y los distintos comandos git que hay.

> Ismael García Vela
```

- Creamos un archivo y añadimos el nombre del archivo al .gitignore, así ese archivo se ignorará a la hora de hacer el add y no se subirá.

```
prueba01:
> touch nosubir
> echo "nosubir" > .gitignore
> cat .gitignore
nosubir
> git add .
> git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .gitignore

> git commit -m "add .gitignore"
[master babb651] add .gitignore
1 file changed, 1 insertion(+)
create mode 100644 .gitignore
> git push -u
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 315 bytes | 315.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
```

garciavelaismael add .gitignore babb651 36 seconds ago 6 commits

doc	first commit	1 hour ago
.gitignore	add .gitignore	36 seconds ago
README.md	fourth commit	39 minutes ago

README.md

PRUEBA01 ✎

Texto de ejemplo 🗑

HOLA

DOC 📄

Explicación de cómo crear un repositorio y los distintos comandos git que hay.

Ismael García Vela

COMANDOS GIT

- **git init** → Inicia el repositorio
- **git add** → Añade archivos para el commit. Se puede crear un archivo **.gitignore** para meter los nombres de archivos o carpetas que **NO** quieres que se añadan.
- **git commit** → Da paso a poder hacer el push y añade un comentario a todos los archivos/carpetas.
- **git push** → Sube al repositorio
- **git pull** → Trae todos los cambios de la nube al repositorio local
- **git status** → Para ver el estado del git
- **git clone** → Clona el repositorio completo