

## GIT

- Instalamos la última versión de git.
- Miramos si está correctamente instalada en el powershell: `git --version`

```
PS C:\Users\Usuario> git -version
>> ^C
PS C:\Users\Usuario> git -version
unknown option: -version
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
          [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
          [-p | --paginate | -P | --no-pager] [--no-replace-objects]
          [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
          [--super-prefix=<path>] [--config-env=<name>=<envvar>]
          <command> [<args>]
PS C:\Users\Usuario> git --version
git version 2.37.3.windows.1
PS C:\Users\Usuario>
```

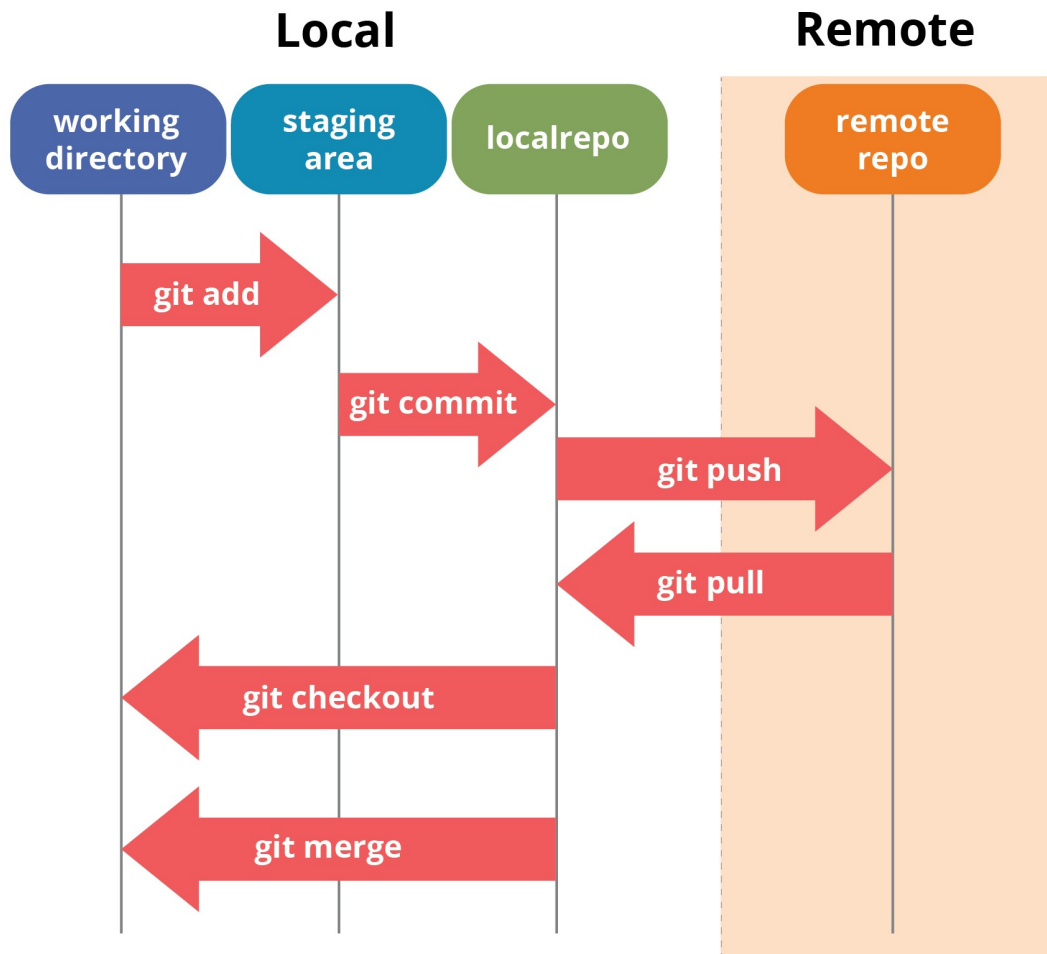
- Luego nos tocará crear nuestro repositorio git en local: `git init`.
- `git config` Iniciar sesión con nuestra cuenta para poder commitear
- `git config --global user.email carlos@gmail.com`
- `git config --global user.name "Carlos"`

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

- `git status` Vemos el estado del archivo.
- `git add nombrearchivo.txt` Añadimos a la "staging area". Ahora ya no saldrá untrucked, saldrá en verde.
- `git commit -m "mensaje del commit"`.
- Si volvemos a hacer `GIT STATUS` ya no saldrá nada.

```
PS C:\Users\Usuario\Desktop\proyecto-git> git commit -m "mi primer commit"
[master (root-commit) 9fb1147] mi primer commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 hola.txt
PS C:\Users\Usuario\Desktop\proyecto-git>
```

- `Git log` -> vemos los commits hechos.



- Git clone -> podemos clonar un repositorio subido de github

```

mi primer commit
PS C:\Users\Usuario\Desktop\proyecto-git> git clone https://github.com/SilSuvi/LM-MD11.git
Cloning into 'LM-MD11'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 6 (delta 2), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.
Resolving deltas: 100% (2/2), done.

```

#### SUBIR COMMITS A UN REPOSITORIO REMOTO

- Necesitamos tener github y crear un repositorio

## Create a new repository

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

### Repository template

Start your repository with a template repository's contents.

No template ▾

Owner \*



SilSuvi ▾

Repository name \*


/ git-proyecto



Great repository names are short and memorable. Need inspiration? How about [special-tribble?](#)

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](#).

.gitignore template: None ▾

### Choose a license

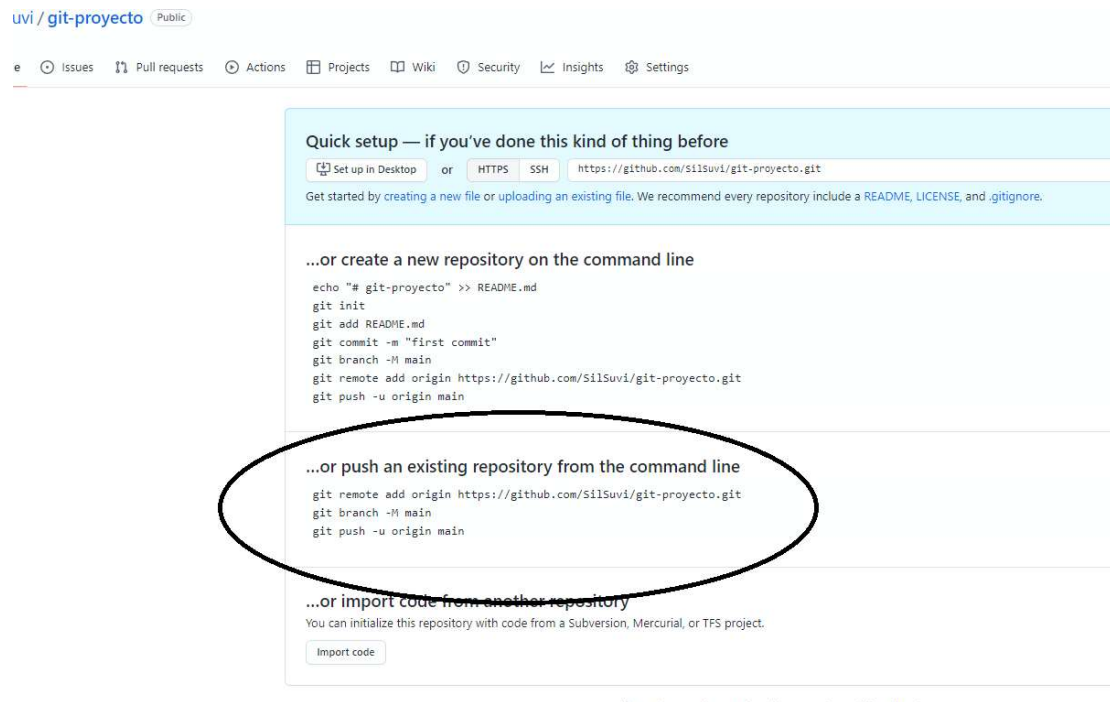
A license tells others what they can and can't do with your code. [Learn more](#).

License: None ▾

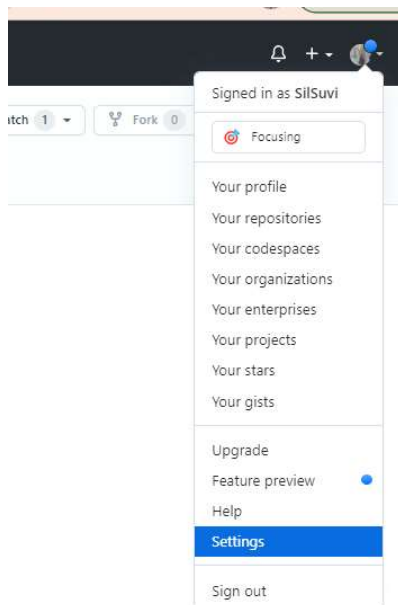
 You are creating a public repository in your personal account.

Create repository

- Una vez creado, ponemos en la terminal el siguiente código.



- Al hacerlo, nos pedirá logearnos a github o utilizar un token.
- ¿Cómo crear un token? Vamos a settings, developer settings



Access

Billing and plans

Emails

Password and authentication

SSH and GPG keys

Organizations

Enterprises

Moderation

Code, planning, and automation

Repositories

Packages

GitHub Copilot

Pages

Saved replies

Security

Code security and analysis

Integrations

Applications

Scheduled reminders

Archives

Security log

Sponsorship log

<> Developer settings

Public email

Select a verified email to display

You have set your email address to private. To toggle email privacy, go to [email settings](#) and uncheck "Keep my email address private."

Bio

Tell us a little bit about yourself

You can @mention other users and organizations to link to them.

URL

Twitter username

Company

You can @mention your company's GitHub organization to link it.

Location

Tarragona

All of the fields on this page are optional and can be deleted at any time, and by filling them out, you're giving us consent to share this data wherever your user profile appears. Please see our [privacy statement](#) to learn more about how we use this information.

Update profile

Contributions & Activity

☐ Make profile private and hide activity (Beta)

Enabling this will hide your contributions and activity from your GitHub profile and from social features like followers, stars, feeds.

GitHub Apps

OAuth Apps

Personal access tokens

Personal access tokens

Generate new token

Tokens you have generated that can be used to access the [GitHub API](#).

probado — admin:gpg\_key, admin:org, admin:org\_hook, admin:public\_key, admin:repo\_hook, admin:ssh\_signing\_key, project, repo

Last used within the last week

Expires on Tue, Oct 4 2022.

ts — admin:enterprise, admin:gpg\_key, admin:org, admin:org\_hook, admin:public\_key, admin:repo\_hook, delete:packages, delete\_repo, gist, notifications, repo, user, workflow, write:discussion, write:packages

Last used within the last 5 month

This token has no expiration date.

BBK — admin:enterprise, admin:gpg\_key, admin:org, admin:org\_hook, admin:public\_key, admin:repo\_hook, delete:packages, delete\_repo, gist, notifications, repo, user, workflow, write:discussion, write:packages

Last used within the last 12 month

This token has no expiration date.

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or [authenticate to the API over Basic Authentication](#).

© 2022 GitHub, Inc.

[Terms](#)

[Privacy](#)

[Security](#)

[Status](#)

[Docs](#)

[Contact GitHub](#)

[Pricing](#)

[API](#)

[Training](#)

[Blog](#)

[About](#)

- Personal acces token y lo generamos. ESPECIFICAMOS exactamente qué podrá hacer el usuario que tenga acceso a dicho token.