

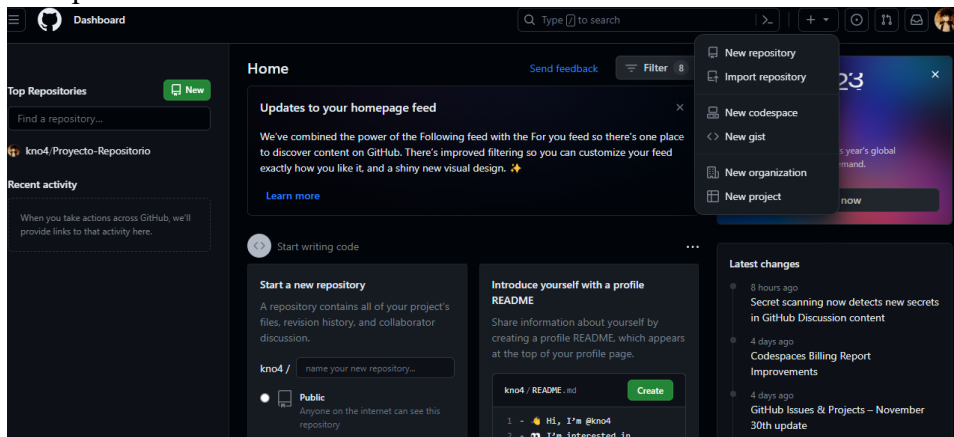
Pasos para trabajar en un repositorio de GitHub

Cano García Enrique Emiliano

1158

1.- Lo primero que se debe hacer para la creación de un repositorio en el sistema de control de versiones GitHub es tener una cuenta en GitHub

2.- Lo segundo es crear un repositorio en el que se almacenaran las versiones y cambios de el proyecto que se realizará, en la parte superior derecha hay un (+) y este nos permite crear un repositorio



3.- Se crean las especificaciones del repositorio, como son el nombre, la descripción, la privacidad y otras configuraciones.

A screenshot of the 'Create a new repository' form in GitHub. The form has a title 'Create a new repository' and a subtitle 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.' Below this, it states 'Required fields are marked with an asterisk (*)'. The 'Owner' field is set to 'kno4'. The 'Repository name' field is empty. A hint says 'Great repository names are short and memorable. Need inspiration? How about literate-guide?'. The 'Description (optional)' field is empty. Under 'Initialize this repository with:', the 'Public' radio button is selected, with the text 'Anyone on the internet can see this repository. You choose who can commit.' Below it, the 'Private' radio button is unselected, with the text 'You choose who can see and commit to this repository.' Under 'Add .gitignore', the 'Add a README file' checkbox is unselected, with the text 'This is where you can write a long description for your project. Learn more about READMEs.' Below that, the 'Add .gitignore' section shows a dropdown menu set to '.gitignore template: None'. At the bottom, it says 'Choose which files not to track from a list of templates. Learn more about ignoring files.'

4.- Creas el repositorio y vas a tus archivos locales

5.- Para trabajar de manera local se debe clonar en nuestro dispositivo. Para esto es necesario Git for Windows, se abrirá la terminal (GitBash here) en la carpeta o dirección en la que queramos clonar el repositorio

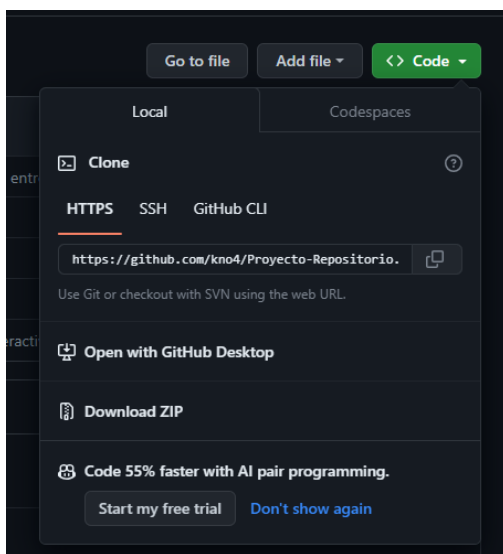
6.- Al abrir la terminal inicializaremos git con los siguientes comandos básicos.

git config --global user.name (nombre de usuario en GitHub)

git config --global user.email (correo de la cuenta de GitHub)

7.- Después de esta inicialización ya es posible clonar el repositorio creado previamente en nuestro dispositivo y almacenamiento local

8.- Copias en GitHub la URL de tu repositorio en el apartado de `<>code`



9.- Una vez copiado vas a tu terminal GitBash y escribes el comando

git clone (URL que copiaste)

y tu repositorio habrá sido clonado de manera local

10.- Si haces un cambio en el repositorio desde GitHub, este cambio no aparecerá en tu proyecto local, para que este cambio se vea reflejado directamente en tu proyecto local deberás escribir el siguiente comando

git pull

Es necesario recordar que la terminal GitBash deberá estar abierta dentro del proyecto local

11.- Si realizaste un cambio en tu proyecto local y quieres que se vea reflejado en GitHub primero se deberá revisar el estado de el proyecto y si los cambios han sido guardados, esto se hace con el comando

git status

```
MINGW64:/c/Users/milok/OneDrive/Documentos/FES Aragón/1er Semestre/Proyecto-Repositorio

milok@DESKTOP-C85V4VA MINGW64 ~/OneDrive/Documentos/FES Aragón/1er Semestre/Proyecto-Repositorio (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

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:   output/proyectoHistoriaInmersiva.exe
        modified:   proyectoHistoriaInmersiva.cpp

no changes added to commit (use "git add" and/or "git commit -a")
```

Este comando nos dirá si se han realizado cambios en el proyecto de manera local, nos lo dirá con un mensaje en rojo, esto quiere decir que los cambios están realizados, pero aún no están preparados para subirlos a GitHub

12.- El comando que se usa para preparar los cambios para subirlos a GitHub es

git add .

```
milok@DESKTOP-C85V4VA MINGW64 ~/OneDrive/Documentos/FES Aragón/1er Semestre/Proyecto-Repositorio (main)
$ git add .
```

Esto permite preparar todos los cambios realizados para subirlos a GitHub

13.- Para crear el commit con los cambios agregados previamente se usa el comando

git commit -m "(Descripción del commit)"

```
milok@DESKTOP-C85V4VA MINGW64 ~/OneDrive/Documentos/FES Aragón/1er Semestre/Proyecto-Repositorio (main)
$ git commit -m "Camino 3 decision 1 por resolver"
[main fe2fa3e] Camino 3 decision 1 por resolver
 2 files changed, 219 insertions(+), 196 deletions(-)
```

Esto permite subir todos los cambios realizados de manera local a GitHub a modo de commit

14.- Una vez creado el commit se debe subir a GitHub con otro comando

git push

```
milok@DESKTOP-C85V4VA MINGW64 ~/OneDrive/Documentos/FES Aragón/1er Semestre/Proyecto-Repositorio (main)
$ git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 151.02 KiB | 2.19 MiB/s, done.
Total 5 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/kno4/Proyecto-Repositorio.git
   c5cf21b..fe2fa3e  main -> main
```

Este comando hace que el commit previamente creado se suba a la plataforma GitHub de manera directa

15.- Para saber todos los cambios realizados, commits y demás especificaciones se usa el comando

git log

```
milok@DESKTOP-C85V4VA MINGW64 ~/OneDrive/Documentos/FES Aragón/1er Semestre/Proyecto-Repositorio (main)
$ git log
commit 083153acd44f27b8161676bc55df21a17cc5bf74 (HEAD -> main, origin/main, origin/HEAD)
Author: kno4 <milokno4@gmail.com>
Date: Tue Nov 28 11:29:31 2023 -0600

    Última Version Entregable

commit 2bf1Feaab8d0fd8ae7bcadc611a6792bd62fc19e
Author: kno4 <milokno4@gmail.com>
Date: Mon Nov 27 22:58:03 2023 -0600

    Último arreglo antes de la entrega

commit a87cc5be937443aa89493aa29d3677a94aa0d57f
Author: kno4 <milokno4@gmail.com>
Date: Sat Nov 25 21:01:24 2023 -0600

    Acabar los finales

commit fe2fa3e66beb4dd0a57d9eb13dc152d5212867b5
Author: kno4 <milokno4@gmail.com>
Date: Fri Nov 24 14:15:47 2023 -0600

    Camino 3 decision 1 por resolver

commit c5cf21b2f1c60d9a19e4321aad848432b0899d9
Author: kno4 <milokno4@gmail.com>
Date: Thu Nov 23 13:16:54 2023 -0600

    CIUDAD RENRIA adicion

commit f982290e2d2d00fad7c66658d3bbd37ea41cf267
Author: kno4 <milokno4@gmail.com>
Date: Wed Nov 22 20:15:23 2023 -0600

    Avance Proyecto Historia Inmersiva

commit 13d2c424e700fdbb6679d89f2b455dd96293d8aa
Author: kno4 <milokno4@gmail.com>
Date: Wed Nov 22 20:11:24 2023 -0600

    Avance Proyecto Historia Inmersiva
```

Esto te permite ver quien realizo los cambios, la fecha del commit y que descripción para el commit agregó