

Primer Proyecto

Base de Datos

Git

Los primeros que necesitamos es instalar es gift bash para poner tener una consola en la que introducir los comandos de a continuación y esta evidentemente es la más cómoda para gift.

Debes crearte una cuenta en gifthub la cual es muy sencilla e intuitiva y es conveniente un editor de texto, yo he elegido visual estudio code.

En este trabajo solo se nombrarán etiquetas y sus funciones con textos simples.

Para empezar abrimos gift bash, dentro de este escribimos “cd Desktop” en este caso es desktop porque es dentro del escritorio donde se encuentra la carpeta que vamos a trabajar. Lo siguiente es introducir el comando “cd” y a continuación el nombre de la carpeta que es PROYECTO en el ejercicio.



```
MINGW64/c/Users/ismae/Desktop/proyecto

Casa@LAPTOP-I6MTRBSS MINGW64 ~
$ cd Desktop

Casa@LAPTOP-I6MTRBSS MINGW64 ~/Desktop
$ cd proyecto

Casa@LAPTOP-I6MTRBSS MINGW64 ~/Desktop/proyecto (master)
$ |
```

A continuación usaremos los comandos “git init” para meternos a interactuar con git dentro de él, ya dentro “git status” nos dará un informe general de este.

```

MINGW64/c/Users/ismae/Desktop/proyecto
Casa@LAPTOP-I6MTRBSS MINGW64 ~/Desktop/proyecto (master)
$ git init
Reinitialized existing Git repository in C:/Users/ismae/Desktop/proyecto/.git/

Casa@LAPTOP-I6MTRBSS MINGW64 ~/Desktop/proyecto (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
      src/

nothing added to commit but untracked files present (use "git add" to track)

Casa@LAPTOP-I6MTRBSS MINGW64 ~/Desktop/proyecto (master)
$

```

Como puedes ver la carpeta creada con varios textos ,dentro de la carpeta proyectos, no está digamos en esa especie de zona de espera entre el usuario y el repositorio. Se sabe que es una carpeta porque tiene una barra al final.

Para solucionar esto el comando “git add (nombre carpeta o archivo)” permite agregarlo a esta zona. Luego usaremos “git commit” para hacer como si fuera un punto de guardado o versión del código.

```

MINGW64/c/Users/ismae/Desktop/proyecto
track)

Casa@LAPTOP-I6MTRBSS MINGW64 ~/Desktop/proyecto (master)
$ git add src

Casa@LAPTOP-I6MTRBSS MINGW64 ~/Desktop/proyecto (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
      new file:   src/Fin.txt
      new file:   "src/continuaci\303\263n.txt"
      new file:   src/principio.txt

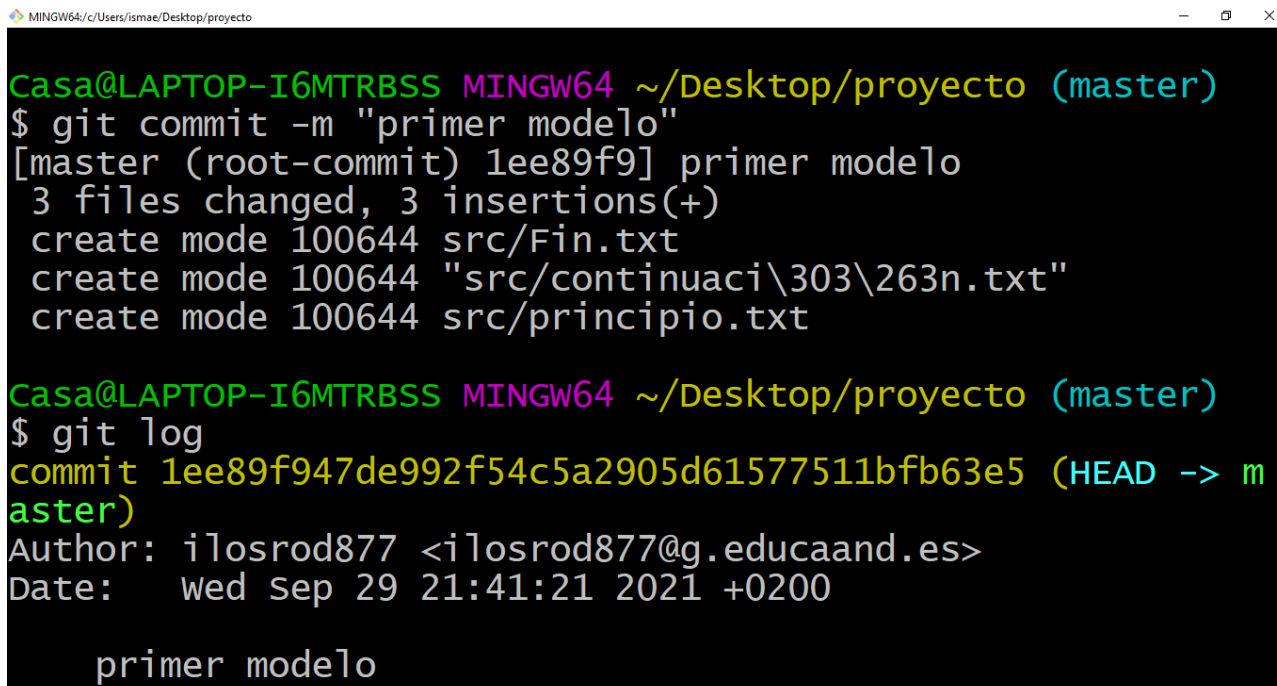
Casa@LAPTOP-I6MTRBSS MINGW64 ~/Desktop/proyecto (master)
$

```

Aquí vemos que ya está agregado después de comprobarlo de nuevo con “git status”.

Cuando intentes introducir el comando “git commit” te pedirá que metas dos comandos que se pueden copiar y pegar cambiando el ejemplo por el primero con tu correo en github y el segundo po tu nombre de usuario.

Una vez hecho introduce el comando “git commit -m (frase a insertar)”. La frase debe ir entre comillas.

A screenshot of a terminal window with a black background and green, yellow, and white text. The window title is 'MINGW64/c/Users/ismae/Desktop/proyecto'. The user is at the prompt 'Casa@LAPTOP-I6MTRBSS MINGW64 ~/Desktop/proyecto (master)'. They enter '\$ git commit -m "primer modelo"', which results in '[master (root-commit) 1ee89f9] primer modelo', '3 files changed, 3 insertions(+)', and 'create mode 100644 src/Fin.txt', 'create mode 100644 "src/continuaci\303\263n.txt"', and 'create mode 100644 src/principio.txt'. Then they enter '\$ git log', which shows 'commit 1ee89f947de992f54c5a2905d61577511bfb63e5 (HEAD -> master)', 'Author: ilosrod877 <ilosrod877@g.educaand.es>', 'Date: wed sep 29 21:41:21 2021 +0200', and 'primer modelo'.

Esto se hace para evitar tener que abrir una máquina en a que introducir la frase que he introducido anteriormente.

Como se puede apreciar luego he introducido el comando “git log” para comprobar que está correcto Nos muestra la fecha de la modificación del proyecto y por quien ha sido hecha y le da una ID que nos permita acceder a esa versión.

Lo siguiente es dentro de github creamos un repositorio. Le damos un nombre, una descripción si queremos, conseguimos la versión gratuita o pública y no elegimos usar read.me.

A continuación te da unos pasos para usar el comando “push” y menos el de read.me y el de init seguimos los pasos copiando pegando, cuando lo hacemos se abre una ventana en la que introduces tu usuario y contraseña y ya se realizan los cambios.

Ahora bien lo que conseguimos con esto es llevar al repositorio ese o esos puntos de guardado del usuario al repositorio. El comando “pull” lo lleva del repositorio al usuario.

Solo queda el comando “git.clone” que clona el repositorio a un usuario; y “.gitignore”que hace los documentos dentro de esto no los tenga en cuenta el programa.