

A thick dark brown vertical bar runs down the left side of the page. An orange arrow points to the right from this bar, containing the date. Below the bar, several thin, curved lines in dark brown and grey sweep upwards and to the right.

29-9-2023

# UD 0 Ejercicio 1

Entornos de Desarrollo

Francisco David Martin Reguero  
NUEVO DESGLOSE IES CAMPANILLAS

En este ejercicio vamos a tratar de familiarizarnos con sus funciones básicas:

- Instalemos Git en nuestro PC: <https://git-scm.com/downloads>

Instalaremos la versión de Git correspondiente dependiendo de nuestro sistema operativo

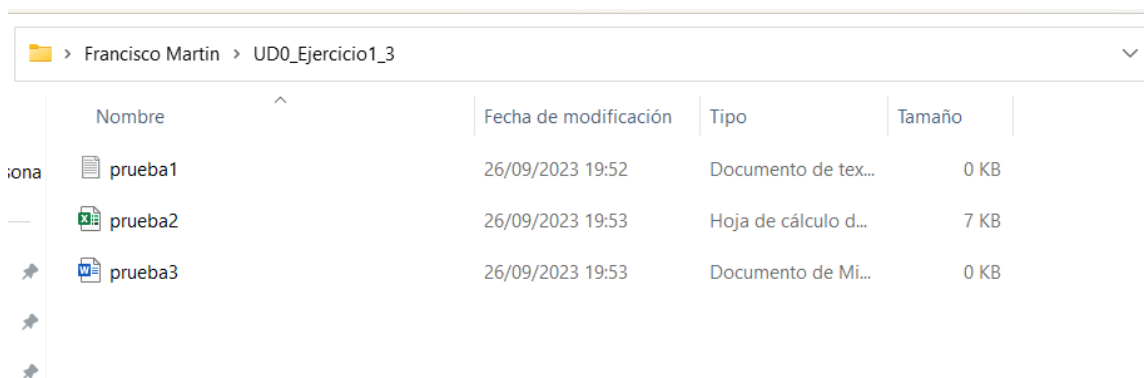
- Comprobemos la instalación con 'git --version'

Para ellos abrimos la terminal y escribimos el comando “**git --version**”, si todo ha salido bien nos dará la versión instalada de git

```
PS C:\Users\franc\UD0_Ejercicio1_3> git --version
git version 2.42.0.windows.2
PS C:\Users\franc\UD0_Ejercicio1_3>
```

- Creemos un directorio para este ejercicio y añadamos algunos ficheros de texto en él

En este directorio guardaremos los ficheros que subiremos próximamente a GitHub



Francisco Martin > UD0_Ejercicio1_3				
	Nombre	Fecha de modificación	Tipo	Tamaño
prueba1	prueba1	26/09/2023 19:52	Documento de tex...	0 KB
prueba2	prueba2	26/09/2023 19:53	Hoja de cálculo d...	7 KB
prueba3	prueba3	26/09/2023 19:53	Documento de Mi...	0 KB

- Creemos nuestra propia cuenta en Github

Nos registramos para poder vincular el directorio a GitHub en un futuro

- Crea un repositorio desde Github (preferentemente con el mismo nombre que tu directorio)

Para crear el repositorio nos dirigimos al siguiente enlace “ <https://github.com/new> ” donde para que no haya ningún problema lo dejaremos todo por defecto y le ponemos el nombre que queramos

- Inicialicemos desde consola nuestro repositorio: 'git init'

Escribiremos el comando una vez estemos dentro de la carpeta que será nuestro repositorio con el comando “**cd**” y ya usamos el comando “**git init**” para instalar el repositorio local

- Consulta el estado del "árbol de trabajo" con 'git status' y haz captura de pantalla

Consultamos el status con el comando “**git status**” ya que nos indicará si hay archivos modificados, sin subir a GitHub, etc

```
PS C:\Users\franc\UD0_Ejercicio1> git init
Reinitialized existing Git repository in C:/Users/franc/UD0_Ejercicio1/.git/
PS C:\Users\franc\UD0_Ejercicio1> git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
PS C:\Users\franc\UD0_Ejercicio1> |
```

- Notifiquemos a Git qué ficheros queremos añadir al repositorio: 'git add'
- Consulta el estado y haz captura

En las siguientes capturas se verá como primero los archivos están en rojo debido a que no están añadidos con el comando “**git add**” pero una vez añadamos los ficheros nos saldrán en verde.

```
PS C:\Users\franc\UD0_Ejercicio1> git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
prueba1.txt
prueba2.xlsx
prueba3.docx
```

```
PS C:\Users\franc\UD0_Ejercicio1> git add --all
PS C:\Users\franc\UD0_Ejercicio1> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
new file:   prueba1.txt
new file:   prueba2.xlsx
new file:   prueba3.docx

PS C:\Users\franc\UD0_Ejercicio1> |
```

- "Empaquetemos" o "preparemos para remitir" nuestra aportación al repositorio con 'git commit' (¡Ojo, este es nuestro primer commit!)
- Consulta el estado y haz captura

Escribimos el comando “**git commit -m “nombre”**” con el nombre que queramos entre comillas y prepara los archivos para poder ser subidos a GitHub

```
PS C:\Users\franc\UD0_Ejercicio1> git commit -m "primero"
[master (root-commit) f1c432e] primero
 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 prueba1.txt
 create mode 100644 prueba2.xlsx
 create mode 100644 prueba3.docx
PS C:\Users\franc\UD0_Ejercicio1> git status
On branch master
nothing to commit, working tree clean
PS C:\Users\franc\UD0_Ejercicio1> |
```

- Cambia la rama sobre la que trabajará Git de 'master' (por defecto) a 'main' (que es la que usa github)

La cambiamos ya que por defecto está en master que es la que usa git pero para poder subir los archivos debemos de estar en la rama main que es la que usa GitHub y cambiamos la rama con el comando **“git Branch -M main”**

- Comunica a la herramienta Git la localización de tu repositorio en la nube (Github)
- Sube tus paquete a Github (git push) y haz captura

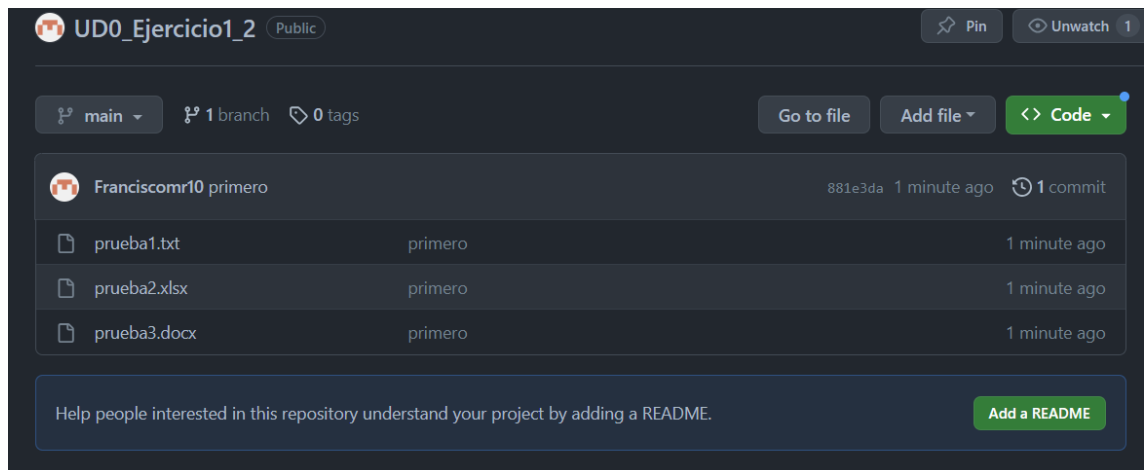
Con estos pasos primero indicamos a donde queremos que se envíen los archivos, en este caso al repositorio creado anteriormente con el comando **“git temote add origin <enlace>”**.

Luego con el comando **“git push -u origin main”** subiríamos los ficheros pero antes tendríamos que indicar que somos nosotros mediante un login

```
PS C:\Users\franc\UD0_Ejercicio1_3> git branch -M main
PS C:\Users\franc\UD0_Ejercicio1_3> git remote add origin https://github.com/Franciscomr10/UD0_Ejercicio1_2.git
PS C:\Users\franc\UD0_Ejercicio1_3> git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 5.37 KiB | 5.37 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Franciscomr10/UD0_Ejercicio1_2.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\franc\UD0_Ejercicio1_3> |
```

- Comprueba en Github la operación y haz una nueva captura

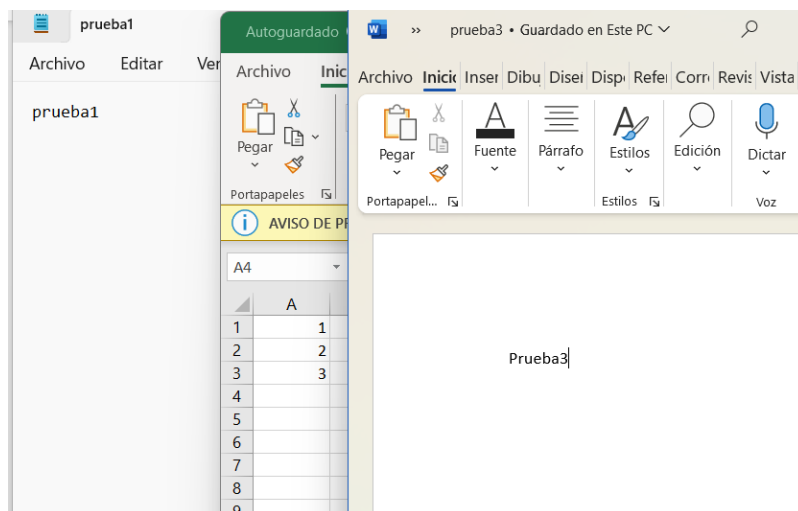
Ahora si nos vamos a la página de GitHub en nuestro repositorio se podrá apreciar que ahora están los 3 archivos que hemos subido en el paso anterior



Cuando termines:

- Haz modificaciones manualmente a los ficheros de texto locales

Hacemos algunas modificaciones a los archivos que hemos creado para actualizar los ficheros en GitHub



- Consulta el estado del "árbol de trabajo" con 'git status' y haz captura de pantalla

Realizamos un “**git status**” y veremos que está otra vez los ficheros en rojo debido a que han sido actualizados

```

PS C:\Users\franc\UD0_Ejercicio1_3> 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:   prueba1.txt
        modified:   prueba2.xlsx
        modified:   prueba3.docx

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\franc\UD0_Ejercicio1_3> |

```

- Notifica a Git los ficheros a añadir al repositorio: 'git add'

Añadimos los ficheros de nuevo con el comando “**git add**”

- Consulta el estado y haz captura

Una vez subidos los archivos con el comando “**git status**” comprobamos que ya están los archivos de nuevo en verde

```

PS C:\Users\franc\UD0_Ejercicio1_3> git add --all
PS C:\Users\franc\UD0_Ejercicio1_3> git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   prueba1.txt
        modified:   prueba2.xlsx
        modified:   prueba3.docx

```

- "Empaquetemos" o "preparemos para remitir" nuestra aportación al repositorio con 'git commit'

Volvemos a preparar para mandar los archivos con el comando “**git commit -m “nombre”**”

- Consulta el estado y haz captura

Cuando hacemos “**git status**” veremos que se ha preparado correctamente ya que no hay nada para preparar

```

PS C:\Users\franc\UD0_Ejercicio1_3> git commit -m "segundo"
[main 409ef9b] segundo
3 files changed, 1 insertion(+)
PS C:\Users\franc\UD0_Ejercicio1_3> git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean
PS C:\Users\franc\UD0_Ejercicio1_3> |

```

- Sube tus paquetes a Github (git push) y haz captura

Ahora escribimos el comando “**git push**” para enviar los ficheros actualizados y ya al tener preparado y conectado a GitHub no tenemos que usar los otros comandos indicando la rama, el repositorio en GitHub y el login.

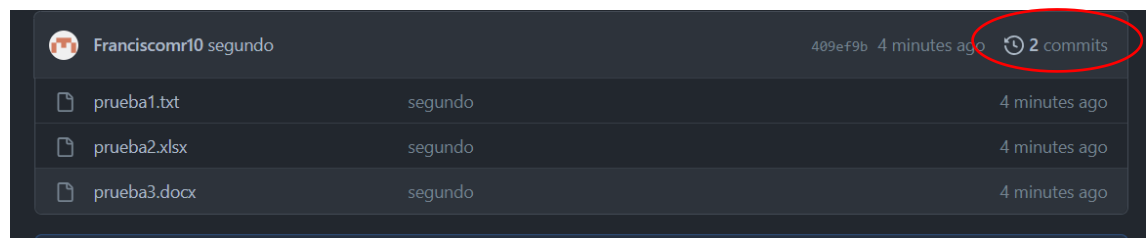
```

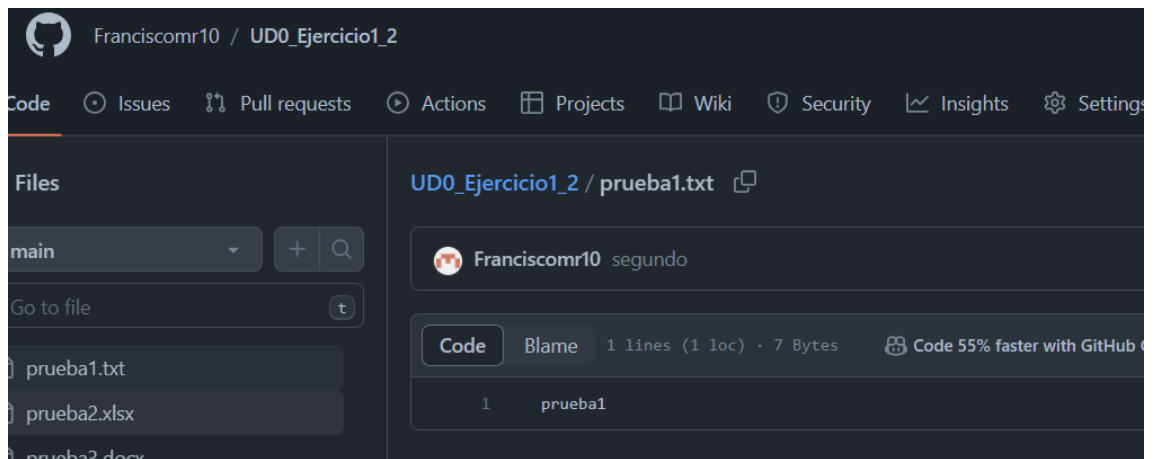
PS C:\Users\franc\UD0_Ejercicio1_3> git push
Enumerating objects: 9, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 12.61 KiB | 12.61 MiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Franciscomr10/UD0_Ejercicio1_2.git
881e3da..409ef9b main -> main
PS C:\Users\franc\UD0_Ejercicio1_3> |

```

- Comprueba en Github la operación y haz una nueva captura

Al terminar el proceso podremos apreciar que se han guardado los cambios, en este caso solo mostraré de un fichero para no tener demasiadas capturas. Y también podemos apreciar que tiene el nombre de “2 commit” que es el nombre que le puse

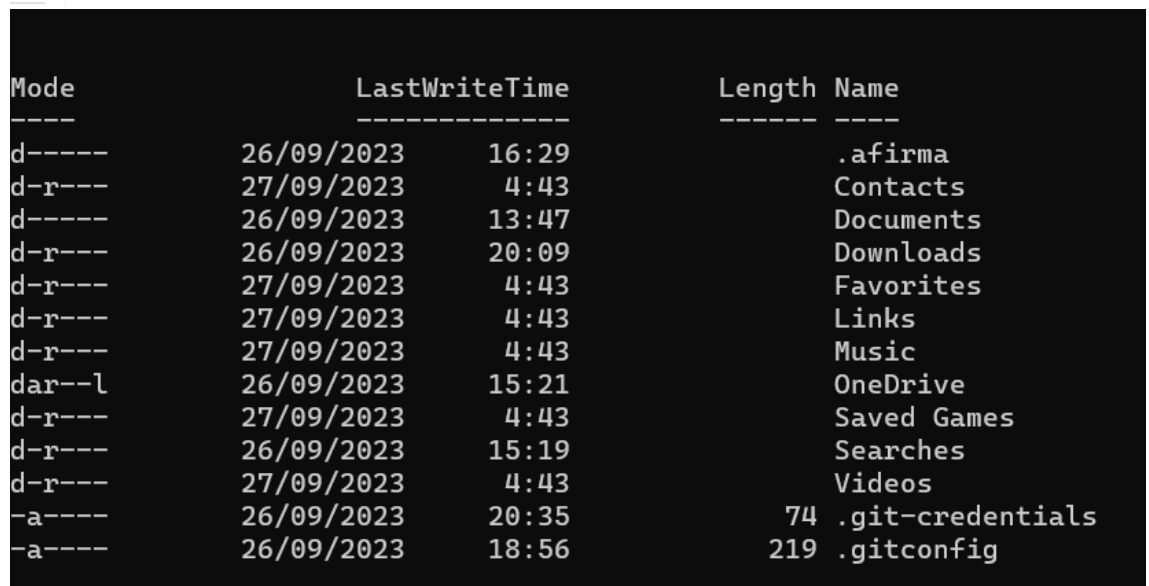
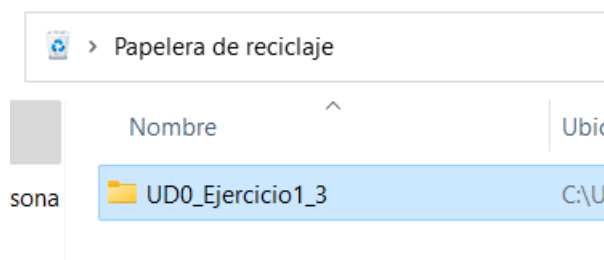




Por último:

- Borra completamente el directorio local de trabajo (con sus ficheros)

Para ello simplemente lo mandamos a la papelera y la vaciamos



- Recupéralo desde tu repositorio en Github. Haz una captura de pantalla

Para poder recuperar el directorio local simplemente tenemos que escribir el comando “**git clone <repositorio>**” cambiando repositorio por el link de nuestro repositorio lo que empezaría a clonar lo que está subido a GitHub a nuestra máquina local restaurando todo lo anterior



```
Administrador: Windows Pow x + v
PS C:\Users\franc\UD0_Ejercicio1_3> cd ..
PS C:\Users\franc> git clone https://github.com/Franciscomr10/UD0_Ejercicio1_2.git
Cloning into 'UD0_Ejercicio1_2'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 9 (delta 1), reused 9 (delta 1), pack-reused 0Receiving objects: 66% (6/9)
Receiving objects: 100% (9/9), 17.94 KiB | 573.00 KiB/s, done.
Resolving deltas: 100% (1/1), done.
PS C:\Users\franc> |
```

UD0\_Ejercicio1\_2 26/09/2023 20:48 Carpeta de archivos

```
remote: Total 9 (delta 1), reused 9 (delta 1), pack-reused 0
Receiving objects: 100% (9/9), 17.94 KiB | 1.28 MiB/s, done.
Resolving deltas: 100% (1/1), done.
PS C:\Users\franc> ls

Directorio: C:\Users\franc

Mode                LastWriteTime         Length Name
----                -
d-----          26/09/2023         16:29      .afirma
d-r---          27/09/2023           4:43     Contacts
d-----          26/09/2023         13:47     Documents
d-r---          26/09/2023         20:09     Downloads
d-r---          27/09/2023           4:43     Favorites
d-r---          27/09/2023           4:43      Links
d-r---          27/09/2023           4:43      Music
dar--l          26/09/2023         15:21     OneDrive
d-r---          27/09/2023           4:43     Saved Games
d-r---          26/09/2023         15:19     Searches
d-----          26/09/2023         21:02     UD0_Ejercicio1_2
d-r---          27/09/2023           4:43     Videos
-a----          26/09/2023         21:02        64 .gitconfig
```