

Resumen

En esta Nota vamos a ver cómo actualizar los archivos del repositorio de Github mediante la consola Git. Previamente se describirá qué es Git y la diferencia con GitHub.

Para la elaboración de esta nota se han seguido las instrucciones del vídeo —Cómo actualizar o subir cambios a un repositorio de Github por la consola de comandos “modo fácil”— del canal Nube Colectiva Devs (https://www.youtube.com/watch?v=_GFgMjjWSQQ)

Índice

1. Actualización de Github mediante consola de comandos	1
1.1. Introducción.	1
1.1.1. ¿Qué es Git?	1
1.1.2. Funciones principales.	1
1.1.3. Diferencia clave: Git vs. GitHub	1
1.2. Instalación de consola de comandos —Git—	2
1.3. Clonación de repositorio en directorio local	4
1.4. Actualización del repositorio modificando/creando archivos en local	5

1. Actualización de Github mediante consola de comandos

1.1. Introducción.

1.1.1. ¿Qué es Git?

Git es un sistema de control de versiones distribuido, gratuito y de código abierto, diseñado para registrar los cambios en los archivos de un proyecto a lo largo del tiempo. En 2026, sigue siendo el estándar fundamental en el desarrollo de software para gestionar el código fuente de forma profesional.

1.1.2. Funciones principales.

- **Historial de cambios:** Permite guardar “instantáneas” (llamadas *commits*) del código, lo que facilita volver a versiones anteriores si algo falla.
- **Trabajo en equipo:** Facilita que varios desarrolladores trabajen simultáneamente en el mismo proyecto sin que sus cambios entren en conflicto.
- **Ramas (Branches):** Permite crear líneas de trabajo independientes para desarrollar nuevas funcionalidades o corregir errores sin alterar el código principal hasta que estén listas.
- **Arquitectura distribuida:** Cada colaborador tiene una copia completa del repositorio y su historial en su propia máquina, permitiendo trabajar sin conexión a internet

1.1.3. Diferencia clave: Git vs. GitHub

Es común confundirlos, pero tienen propósitos distintos:

- **Git:** Es la herramienta técnica (el *software*) que se instala localmente para gestionar las versiones de tus archivos.
- **GitHub:** Es una plataforma en la nube que aloja proyectos que usan Git, facilitando la colaboración remota y la gestión de proyectos a través de una interfaz web.

1.2. Instalación de consola de comandos —Git—

Dentro de nuestro buscador preferido tecleamos git y obtenemos la página oficial del programa: <https://git-scm.com/>. Una vez dentro buscamos la versión instalable, en nuestro caso para Windows, como se muestra en la figura 1.

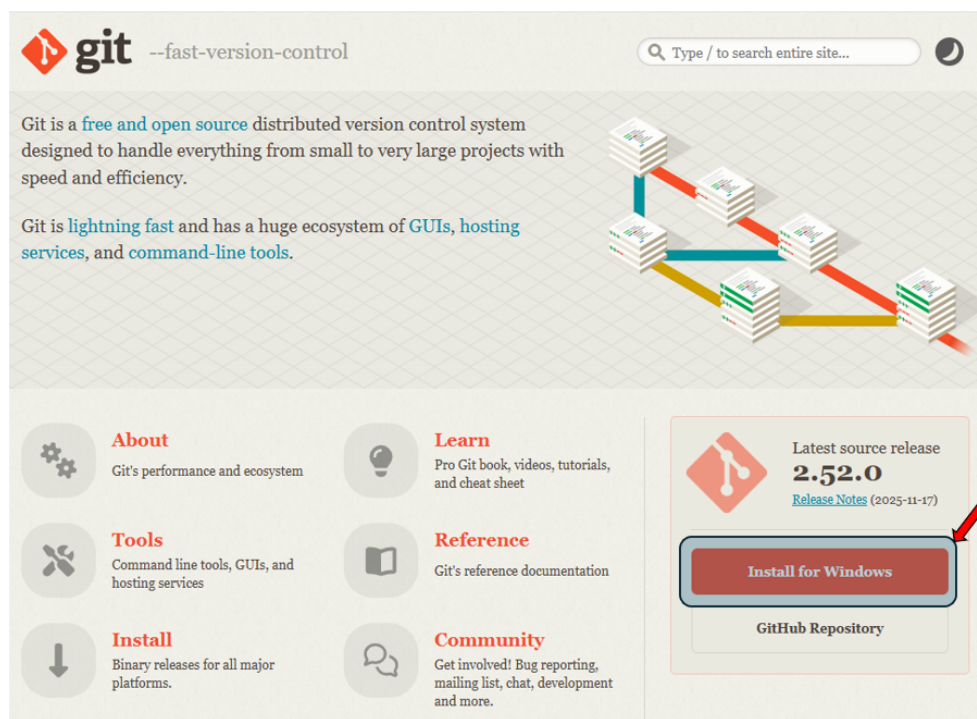


Figura 1: Selección de versión de instalación en Windows

En la siguiente pantalla que se muestra se puede observar que también hay disponibles versiones de descarga para macOS y Linux, así como para hacerlo desde la fuente. Como ya se ha indicado en nuestro caso será la versión de Windows, y la opción “Standalone Installer” para un sistema de 64 bits, como se puede ver en la figura 2

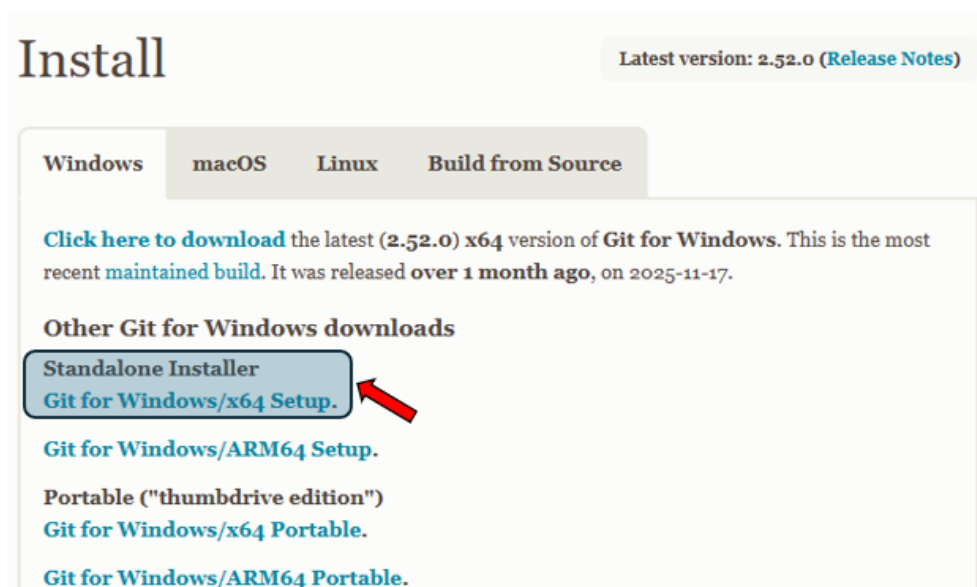


Figura 2: Git for Windows/x64 Setup.

Una vez descargado hacemos clic con el botón derecho del ratón sobre el archivo de instalación y buscamos la opción de Ejecutar como administrador.

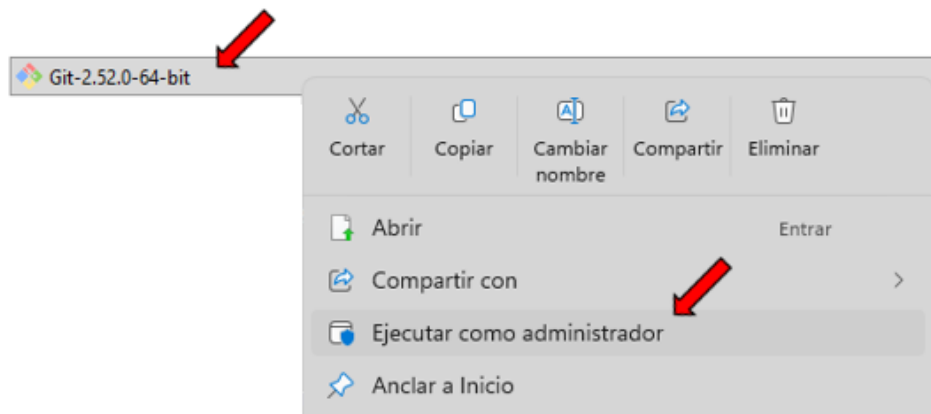


Figura 3: Ejecutar como administrador

Una vez empieza a instalarse el programa podemos ir aceptando todo por defecto hasta llegar a la ventana en la que se puede elegir el editor por defecto (Choosing the default editor used by Git), en este caso elegimos la opción de Sublime Text como muestra la figura 4.

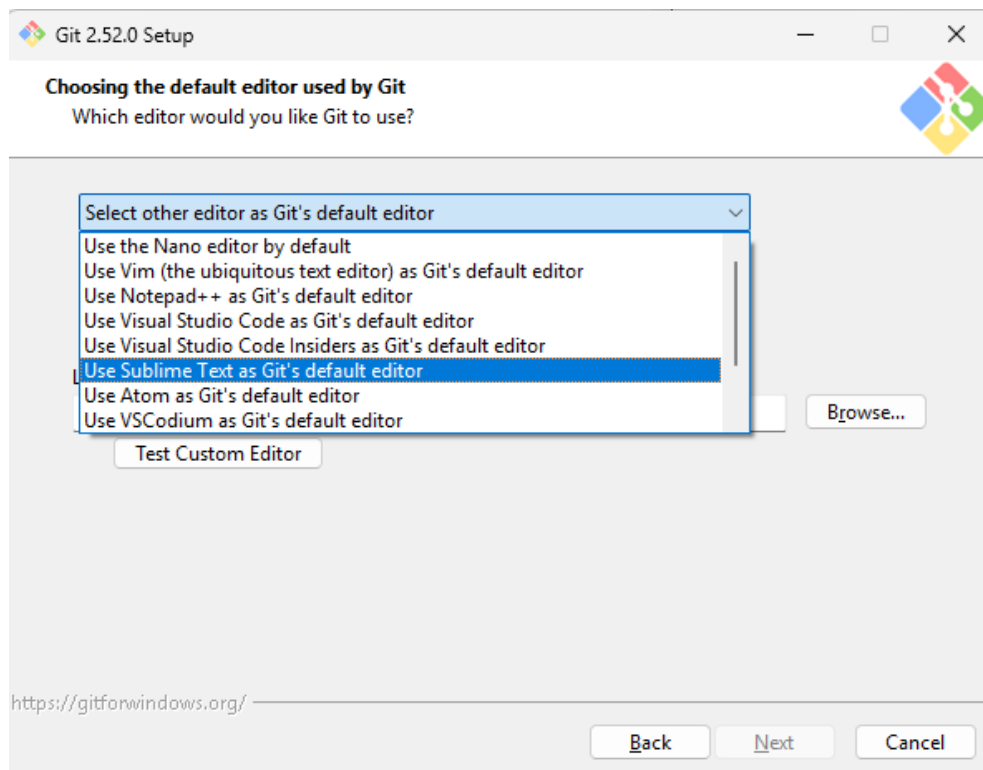


Figura 4: Elección del editor de Git

Una vez elegido el editor seguimos aceptando por defecto las opciones que nos muestran las distintas ventanas hasta llegar a la última, en la que da comienzo la instalación del programa.

1.3. Clonación de repositorio en directorio local

Una vez instalado podemos acceder a la consola haciendo clic en el icono Git Bash que se genera en la lista de programas o haciendo clic con el botón derecho del ratón¹ para abrir el menú contextual y ahí seleccionar igualmente Git Bash, como se puede ver en la figura 5. En este caso vamos a abrir la consola desde una carpeta llamada Git que se ubica en raíz, y que será donde haremos una copia de nuestro repositorio.

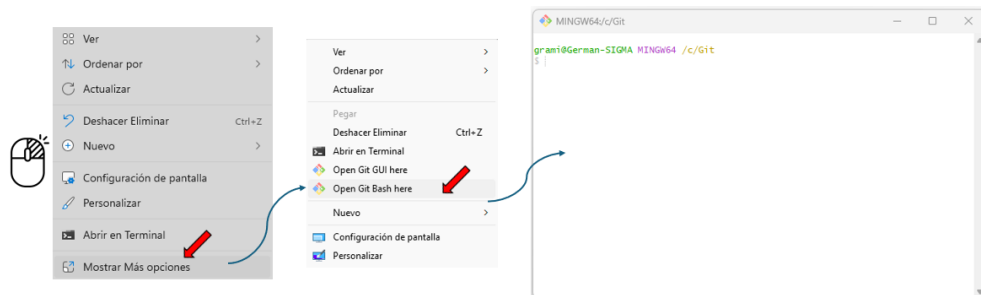


Figura 5: Elección de ubicación de copia del repositorio

Para copiar o clonar nuestro repositorio en la carpeta seleccionada debemos seguir los siguientes pasos, como se muestra en la figura 6:

1. Entrar en Github y hacer clic en nuestro usuario, se abre un menú inferior.
2. Hacer clic en Repositories, los repositorios que tenemos en Github.
3. Seleccionar el repositorio a clonar y hacer clic en él.
4. Hacer clic en el botón verde `<>Code`.
5. Dentro de la opción HTTPS hacer clic en el botón de copiar la dirección del repositorio.

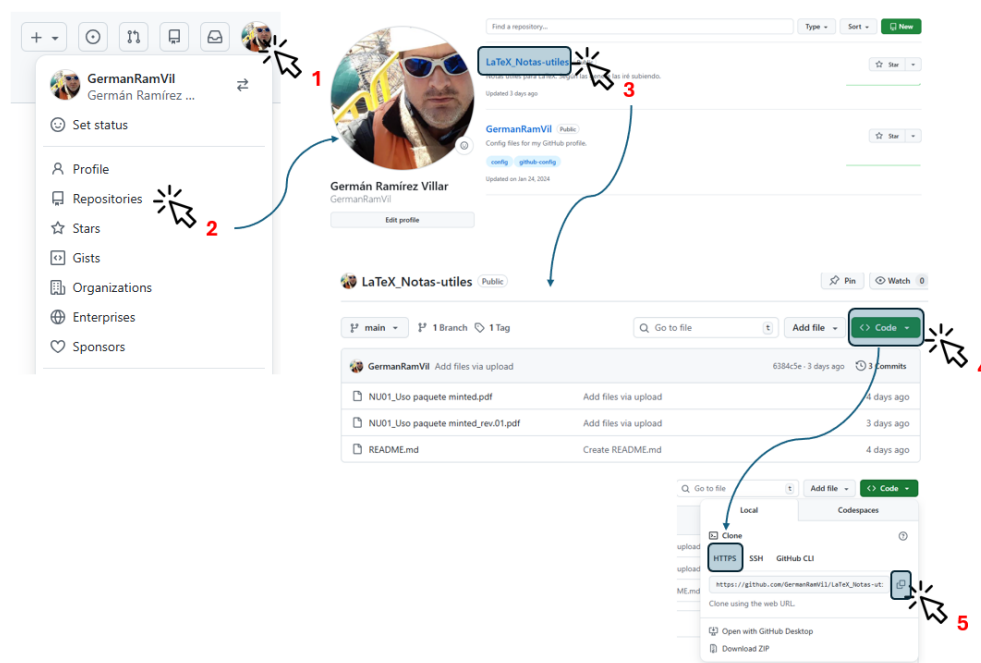
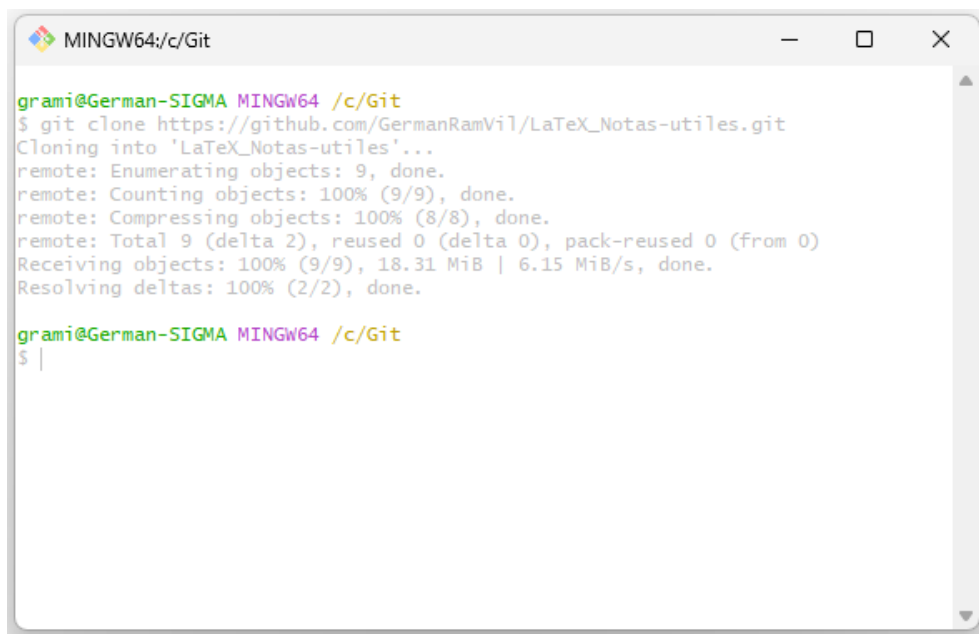


Figura 6: Elección de repositorio a clonar

¹Iconos de flaticon.com

Abrimos la consola como se ha indicado en el paso anterior y se teclea el siguiente código, donde se pega la dirección previamente copiada: `git clone <Dirección Repositorio>`, como se muestra en la figura 7, pulsamos **Enter** para ejecutar.

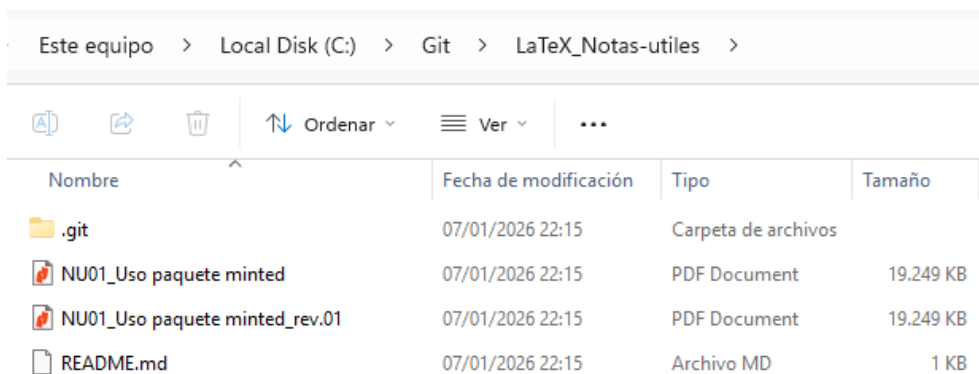


```
mingw64 /c/Git
$ git clone https://github.com/GermanRamVil/LaTeX_Notas-utiles.git
Cloning into 'LaTeX_Notas-utiles'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 9 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (9/9), 18.31 MiB | 6.15 MiB/s, done.
Resolving deltas: 100% (2/2), done.

mingw64 /c/Git
$
```

Figura 7: Comando para clonar repositorio

Como se puede apreciar se ha creado en el directorio elegido una copia de lo contenido en nuestro repositorio de Github.



Este equipo > Local Disk (C:) > Git > LaTeX_Notas-utiles >			
Ordenar Ver ...			
Nombre	Fecha de modificación	Tipo	Tamaño
.git	07/01/2026 22:15	Carpeta de archivos	
NU01_Uso paquete minted	07/01/2026 22:15	PDF Document	19.249 KB
NU01_Uso paquete minted_rev.01	07/01/2026 22:15	PDF Document	19.249 KB
README.md	07/01/2026 22:15	Archivo MD	1 KB

Figura 8: Comprobación de la clonación del repositorio

1.4. Actualización del repositorio modificando/creando archivos en local

Vamos a ver cómo se actualiza el repositorio modificando o creando un nuevo archivo en nuestro directorio local. En este caso vamos a modificar el archivo *NU01_Uso paquete minted_rev.01.pdf*, creando una revisión con imágenes de menor peso, para así generar un pdf de menor tamaño.

Para subir este cambio a nuestro repositorio entraremos en la consola en el directorio afectado² y teclearemos el siguiente código: `git add .`³

²Si se hace desde un directorio que no tenga relación con ningún repositorio la consola nos mostrará un error (fatal: not a git repository (or any of the parent directories): .git)

³`git add .` es un comando de Git que añade todos los cambios (archivos nuevos, modificados y eliminados) del

Este equipo > Local Disk (C:) > Git > LaTeX_Notas-utiles >			
<div> <div>Ordenar</div> <div>Ver</div> <div>...</div> </div>			
Nombre	Fecha de modificación	Tipo	Tamaño
.git	07/01/2026 22:15	Carpeta de archivos	
NU01_Usa paquete minted_R01	11/01/2026 12:26	PDF Document	2.940 KB
NU01_Usa paquete minted	07/01/2026 22:15	PDF Document	19.249 KB
README.md	07/01/2026 22:15	Archivo MD	1 KB

Figura 9: Modificación de archivo por uno de menor tamaño

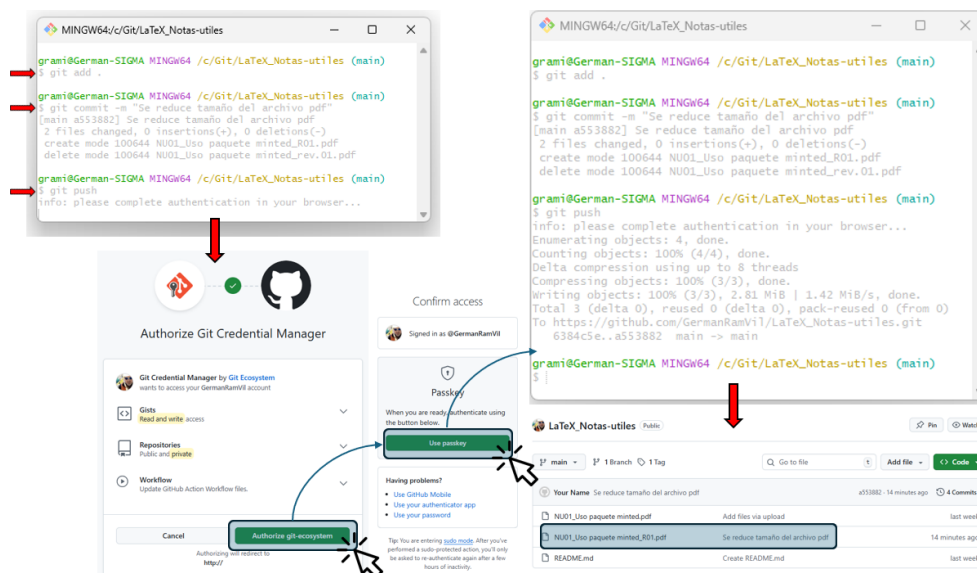
Alternativas a `git add .`

- `git add <archivo>`, para añadir un archivo específico.
- `git add -p`, para preparar partes específicas de un archivo de forma interactiva.
- `git add -A` o `git add --all`, similar a `.` pero a nivel de todo el repositorio, no sólo del directorio actual.

Acto seguido tecleamos en la consola `git commit -m "motivo de la revisión"` y pulsamos Enter, con lo que se actualicen los cambios, en nuestro caso se ha creado un archivo nuevo y se ha eliminado.

Finalmente debemos “empujar” los cambios desde la consola a Github, para lo que teclearemos `git push`. En este caso, al ser la primera vez que lo ejecuto se solicita autenticación, por lo que se accede a Github y se procede haciendo clic en el botón `Authorize git-ecosystem`, que da acceso a otra ventana de autorización, en la que debemos de hacer clic en el botón `Use passkey`.

Al hacer clic en este último paso de autenticación se ejecuta el código en la consola, comprobándose que la actualización se ha subido a Github.

Figura 10: Modificación de archivos en Github desde la consola mediante “empujado” (*push*)

Conforme a lo indicado en la página web de Git ⁴ “es necesario establecer tu nombre de usuario y dirección de correo electrónico. Esto es importante porque los commits de Git usan esta información, y es introducida de manera inmutable en los commits que envías”.

directorio de trabajo actual y sus subdirectorios al área de preparación (*staging area*) para la próxima confirmación (*commit*), permitiéndote preparar una “instantánea” de tu proyecto antes de guardarla permanentemente en el historial de tu repositorio. El punto `.` indica a Git que incluya todos los archivos y directorios dentro de la ubicación actual (y sus subdirectorios). Los cambios no se guardan en el repositorio hasta que se ejecuta `git commit` después de usar `git add`.

⁴<https://git-scm.com/book/es/v2/Inicio—Sobre-el-Control-de-Versiones-Configurando-Git-por-primera-vez>

Si así lo solicita la consola se debe de escribir lo siguiente:

`git config --global user.name "Nombre de usuario"`, y pulsamos Enter.

Acto seguido tecleamos:

`git config user.mail micorreo@ejemplo.com` pulsando Enter una última vez.

“..., sólo necesitas hacer esto una vez si especificas la opción `--global`, ya que Git siempre usará esta información para todo lo que hagas en ese sistema. Si quieres sobrescribir esta información con otro nombre o dirección de correo para proyectos específicos, puedes ejecutar el comando sin la opción `--global` cuando estés en ese proyecto”.