

# **AWE Instalación GIT**

1

The screenshot shows a GitHub repository page for 'prueba1'. At the top, there are buttons for 'Pin', 'Watch 0', 'Fork 0', and 'Star 0'. Below the header, there are two sections: 'Set up GitHub Copilot' and 'Add collaborators to this repository'. Under 'Set up GitHub Copilot', there is a link to 'Get started with GitHub Copilot'. Under 'Add collaborators', there is a link to 'Invite collaborators'. A large blue box at the bottom contains instructions for quick setup ('if you've done this kind of thing before') and command-line creation. It includes a code block for setting up a new repository:

```
echo "# prueba1" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/David24652/prueba1.git
git push -u origin main
```

El enlace al repositorio es <https://github.com/David24652/prueba1>

2

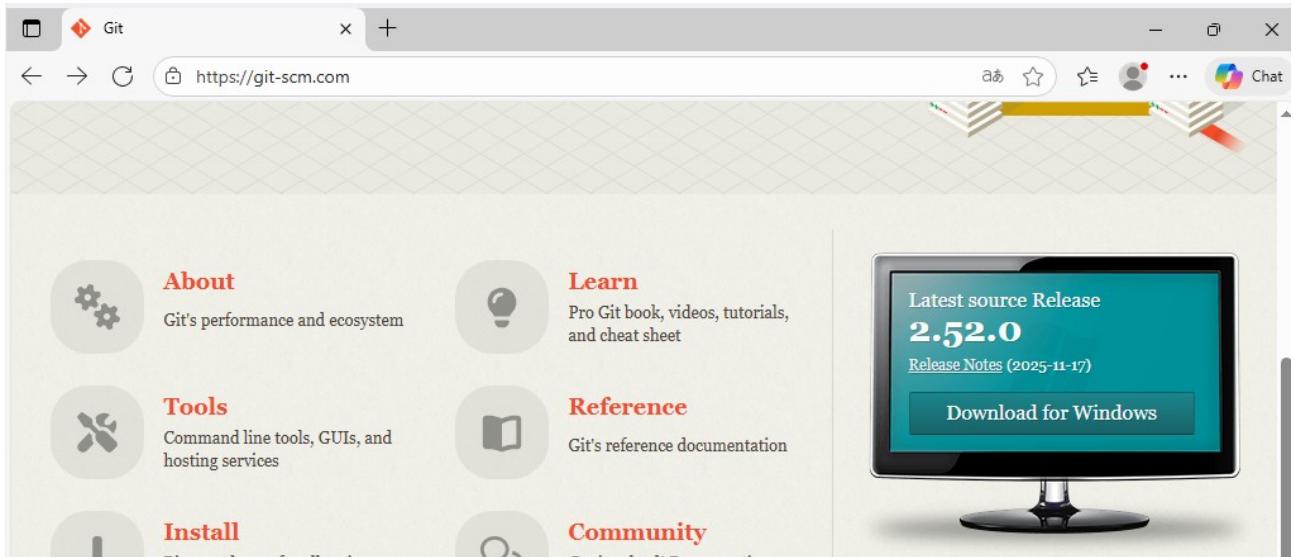
Para MAC he encontrado GitUp, <https://gitup.co/>

Para Windows TortoiseGit, <https://tortoisegit.org/>

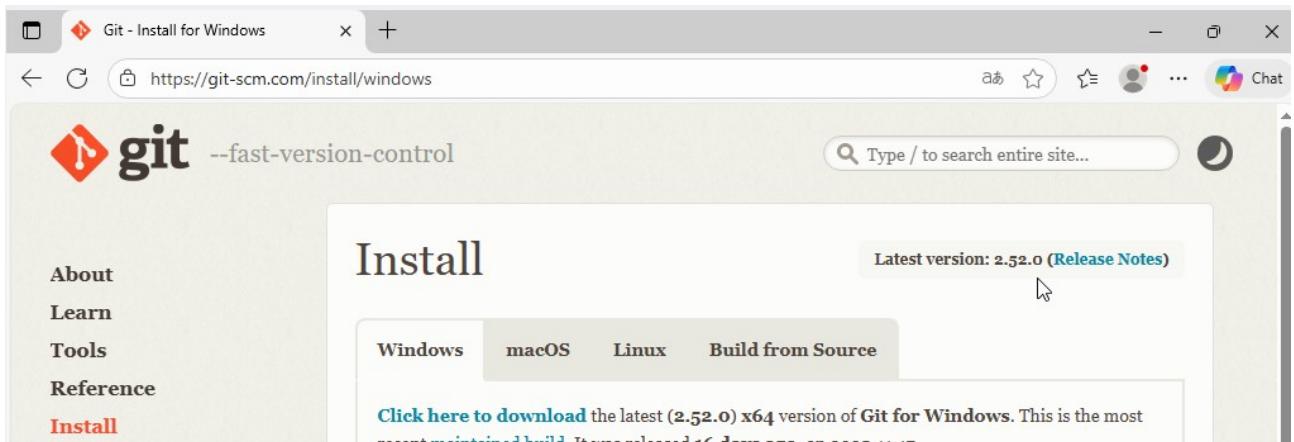
Para Linux Git Cola, <https://git-cola.github.io/>

GitUp es una versión más ligera y simple y está diseñada solamente para MAC y está hecha para tener una vista clara de las ramas. TortoiseGit está integrada en el internet explorer de Windows, así que es más cómodo de usar en Windows pero peor en cualquier otro SO. Git Cola es ligero y de código abierto, su interfaz es más simple pero menos visual

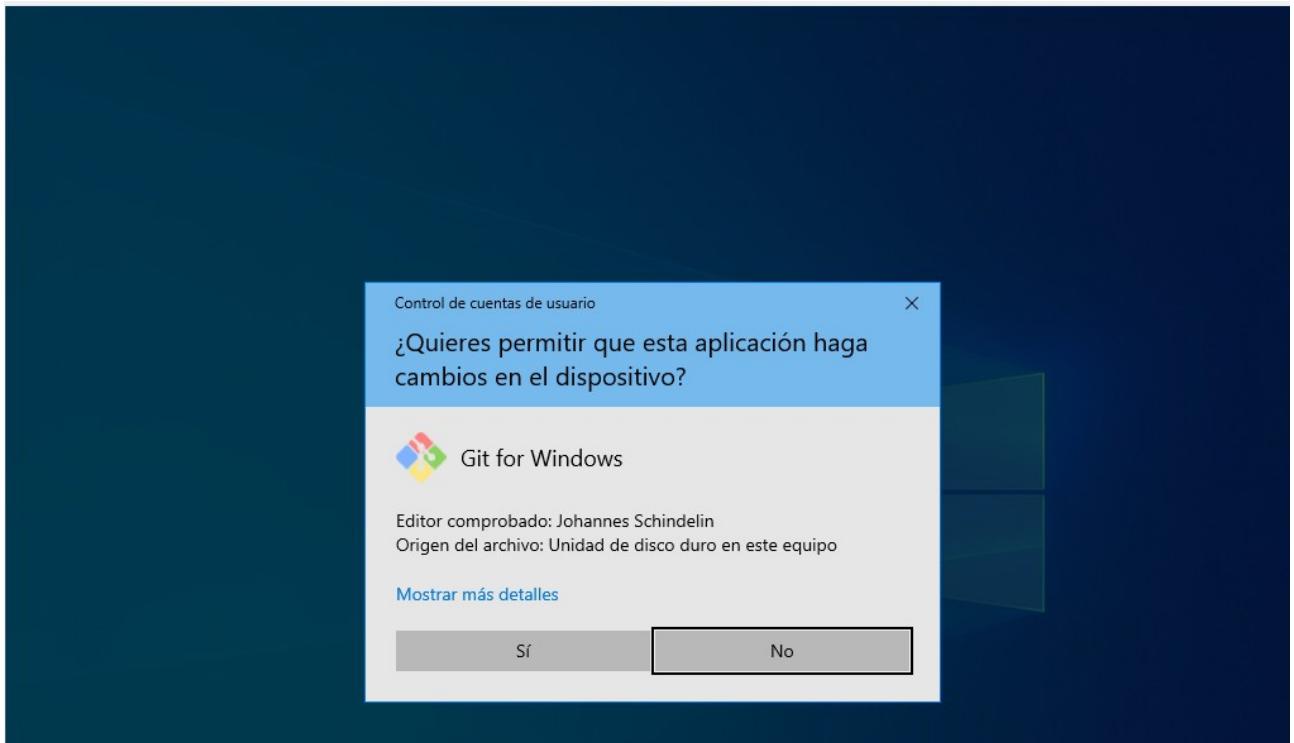
3



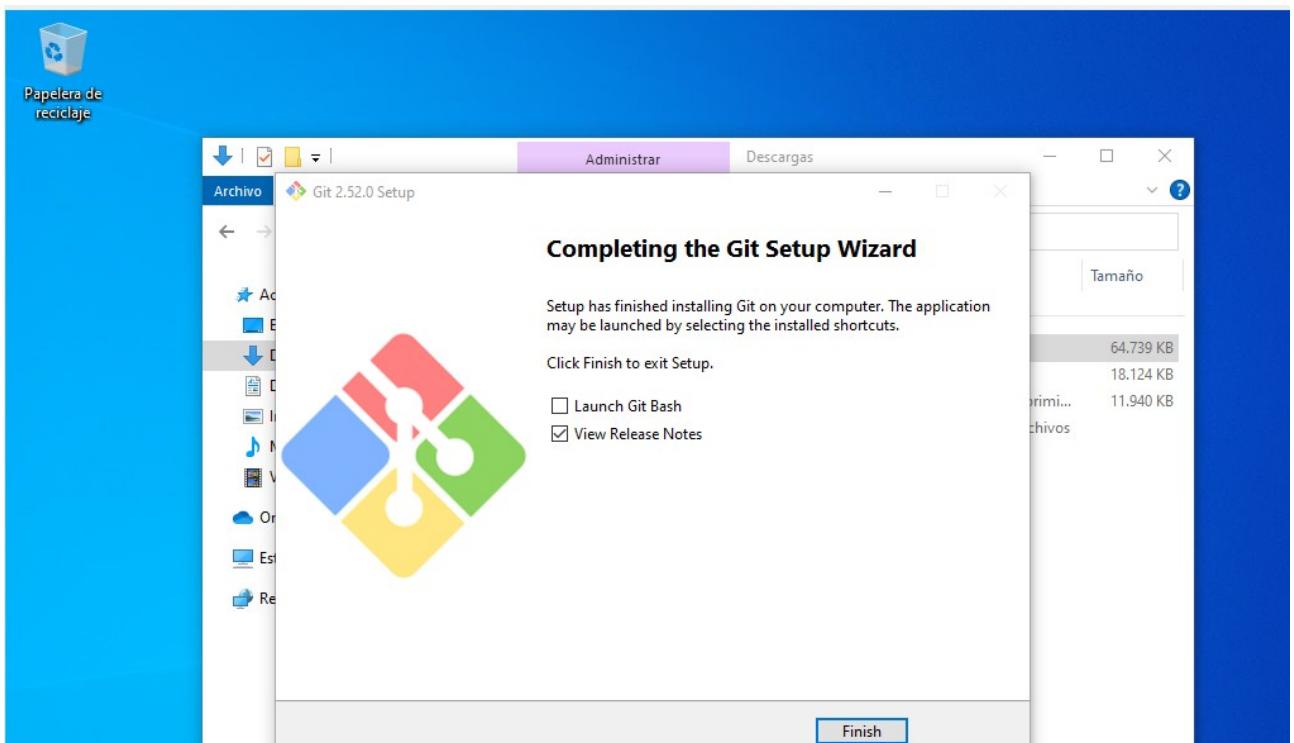
Buscamos en internet la página de Git y clickamos en download for Windows, a la derecha



Después en clik here to download



Después de descargarse hay que ejecutarlo y darle a que si, darle a next todas las veces que te lo pida, sin modificarlo y darle a instalar



Después se le da a finish sin marcar ni desmarcar nada

## 4

Las ramas de Git son líneas de desarrollo que permiten trabajar en nuevas funciones o correcciones sin afectar el código principal. Cada rama mantiene su propio historial, y puedes moverte entre ellas sin perder trabajo, después puedes fusionar una rama con otra para integrar todo el historial

## 5

Primero trabajas localmente, editando los archivos y registrando los avances, estos avances se guardan en tu repositorio local hasta que lo envíes al repositorio remoto, donde otros pueden verlo también. También puedes llevar cosas del repositorio remoto al local. El trabajo se hace en ramas, que después se integran

## 6

Los ficheros tienen 3 estados principales:

Modificado → Indica que el archivo ha cambiado en un repositorio pero aún no está listo para ser guardado.

Preparado → Git lo ha registrado en el área de preparación y está preparada para incluirse en el siguiente sommit

Confirmado o Committed → Registrado permanentemente en el historial del repositorio

## 7

Repository → Es el contenedor del proyecto, donde se almacena el código, los archivos y el historial de los cambios

Commit → Es un registro permanente de los cambios realizados en los archivos. Cada uno guarda una snapshot para poder volver a este punto en cualquier momento

Stage → Donde colocas los cambios que quieras meter en el próximo commit. Es para elegir qué modificaciones e van a guardar y cuáles no

Fork → Una copia completa de un repositorio que se crea en tu propia cuenta. Es para ayudar en proyectos de otra gente sin afectar al repositorio original

Master → Es la rama principal de un repositorio. Es la versión estable del proyecto

Checkout → Es un comando para moverse entre ramas o versiones del repositorio actualizando tu directorio de trabajo con los archivos que correspondan a ese punto del historial

Merge → Es la acción de combinar los cambios de una rama con otra, interando sus historiales. Es para unir el trabajo en paralelo de dos ramas dentro del mismo proyecto

## 8

git config --global → Configura valores globales para Git en tu sistema. Estos se aplican para todos los repositorios del equipo

git config --list → Muestra una lista completa de las configuraciones de Git activas en el momento. Es para verificar cómo está configurado el entorno

git status → Muestra el estado del repositorio, incluyendo que archivos han sido modificados

git push origin master → Envía los commits de tu rama master local al repositorio remoto. Es para compartir los cambios con los demás colaboradores

git init → Crea un nuevo repositorio dentro de la carpeta actual

git add <file> → Añade un archivo al área de preparación para incluirlo en el próximo commit

git add . → Añade todos los archivos nuevos o modificados al área de preparación

git commit → Guarda en el repositorio los cambios que están en preparación creando una nueva versión del proyecto

git commit -m "comment" → Funciona igual que el anterior pero te deja añadirle un comentario al commit

git push → Envía los commits locales al repositorio remoto

git pull → Descarga los cambios del repositorio remoto y los fusiona con la rama local actual

git clone → Crea una copia completa de un repositorio remoto en tu máquina

git checkout → Para cambiar entre ramas o restaurar archivos a versiones específicas

git diff → Muestra las diferencias entre archivos

git branch → Crea, lista o elimina ramas en el repositorio