

# Git y Github

Git Command line



---

# Índice

Introducción	3
¿Qué es Git Bash?	4
¿Cómo instalar Git Bash?	4
Cómo usar Git Bash	4
Cómo navegar por las carpetas	4
Comandos Git Bash	5

---

# Introducción

En esencia, *Git* es un conjunto de programas de utilidades de líneas de comandos que están diseñados para ejecutarse en un entorno de líneas de comandos de estilo *Unix*. Los sistemas operativos modernos como *Linux* y *macOS* incluyen terminales de líneas de comandos *Unix* integrados. Esto convierte a *Linux* y a *macOS* en sistemas operativos complementarios cuando se trabaja con *Git*. En cambio, *Microsoft Windows* utiliza el símbolo del sistema de *Windows*, un entorno de terminal que no es *Unix*.

En entornos de *Windows*, *Git* normalmente se incluye en un paquete como parte de aplicaciones de interfaz gráfica de usuario de nivel superior. Las interfaces gráficas de usuario para *Git* podrían intentar abstraer y ocultar los lenguajes primitivos del sistema de control de versiones subyacente. Esto puede ser una ayuda excepcional para que los principiantes en *Git* contribuyan rápidamente a un proyecto. Una vez que los requisitos de colaboración de un proyecto aumentan con otros miembros del equipo, es fundamental ser consciente de cómo funcionan los métodos de *Git* de verdad sin procesar. En ese momento, puede ser beneficioso disponer de una versión de interfaz gráfica de usuario para las herramientas de líneas de comandos. Se ofrece *Git Bash* para proporcionar una experiencia de *Git* en el terminal.

## ¿Qué es Git Bash?

Bash es una interfaz de línea de comandos creada hace más de veintisiete años por Brian Fox como un reemplazo de software gratuito para Bourne Shell.

Un shell es un tipo específico de interfaz de línea de comandos.

Bash es de "código abierto", lo que significa que cualquiera puede leer el código y sugerir cambios. Desde sus inicios ha contado con el apoyo de una gran comunidad de ingenieros que han trabajado para convertirla en una herramienta increíble. Bash es el shell predeterminado para Linux y Mac. Por estas razones, Bash es el shell más utilizado y ampliamente distribuido.

Windows tiene una interfaz de línea de comandos diferente, llamada símbolo del sistema. Si bien esto tiene muchas de las mismas características que Bash, Bash es mucho más popular. Debido a la fortaleza de la comunidad de código abierto y las herramientas que brindan, dominar Bash es una mejor inversión que dominar el símbolo del sistema.

Para usar Bash en una computadora con Windows, necesitamos descargar e instalar un programa llamado Git Bash. Git Bash (Is the Bash for windows) nos permite acceder fácilmente a Bash así como a otra herramienta llamada Git, dentro del entorno de Windows.

*Git Bash* es un paquete que instala Bash, algunas utilidades comunes de bash y Git en un sistema operativo Windows.

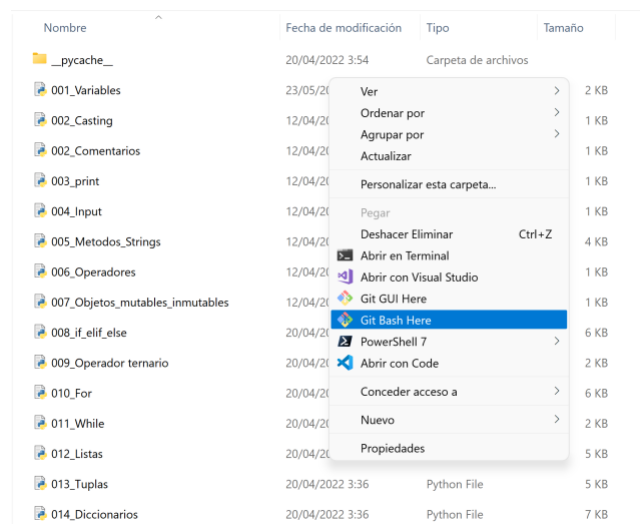
Resumiendo: Git bash, es una herramienta de tipo consola que básicamente nos permite manipular y gestionar todo el proceso de crear puntos de control en nuestro proyecto y manipularlos.

## ¿Cómo instalar Git Bash?

*Git Bash* viene incluido como parte del paquete [Git para Windows](#).

La descarga e instalación de *Git* para *Windows* es idéntica a la de cualquier otra aplicación de Windows.

Una vez descargado e instalado, tendremos disponibles varias herramientas de *GIT*, entre las que destaca la consola *Git Bash* a la que accederemos pulsando el botón derecho del ratón desde nuestro directorio de trabajo:



## Cómo usar Git Bash

*Git Bash* tiene las mismas operaciones que una experiencia *Bash* estándar. Será útil revisar el uso básico de *Bash*. Se verán numerosos comandos de la consola *GitBash* en posteriores temas.

### Cómo navegar por las carpetas

El comando *Bash pwdse* usa para imprimir el 'directorio de trabajo actual'. *pwdes* equivalente a ejecutar *cd* en un terminal *DOS* (host de consola de Windows). Esta es la carpeta o ruta en la que reside la sesión actual de *Bash*.

El comando *Bash ls* se usa para 'listar' el contenido del directorio de trabajo actual. *ls* es equivalente a *DIR* en terminal host de consola de Windows.

Tanto *Bash* como el host de la consola de *Windows* tienen un comando *cd*.

*cd* es un acrónimo de 'Cambiar directorio'. *cd* se invoca con un nombre de directorio adjunto. Ejecutar *cd* cambiará el directorio de trabajo actual de las sesiones de terminal al argumento del directorio pasado.

## Comandos Git Bash

*Git Bash* está empaquetado con comandos adicionales que se pueden encontrar en el directorio de la emulación de *Git Bash*, */usr/bin*.

*Git Bash* en realidad puede proporcionar una experiencia de *shell* bastante sólida en Windows. *Git Bash* viene empaquetado con los siguientes comandos de *Shell*: *Ssh*, *scp*, *cat*, *find*.

Además del conjunto de comandos *Bash* discutido anteriormente, *Git Bash* incluye el conjunto completo de comandos principales de *Git* discutidos a lo largo de este sitio.

Una vez instalado *Git Bash* lo podremos llamar también desde nuestro *IDE* en caso de que disponga de terminal integrada. Cualquiera de los comandos de *Git Bash* será también ejecutado desde la consola de nuestro *IDE*.

