

Manual de Instalación de Git: Paso a Paso

Este manual te guiará a través del proceso de instalación de Git en los sistemas operativos más comunes: Windows, macOS y Linux. Git es una herramienta fundamental para el control de versiones, y tenerla correctamente instalada es el primer paso para dominarla.

1. Instalación de Git en Windows

La forma recomendada de instalar Git en Windows es a través del instalador oficial **Git for Windows**, que incluye Git Bash (una terminal de estilo Unix) y Git GUI.

Paso 1: Descargar el Instalador de Git para Windows

1. Abre tu navegador web y visita la página oficial de Git: <https://git-scm.com/download/win>
2. La descarga debería comenzar automáticamente para la versión más reciente compatible con tu sistema. Si no es así, haz clic en el enlace para la versión de 64-bit o 32-bit según tu arquitectura.

Paso 2: Ejecutar el Instalador

1. Una vez finalizada la descarga, localiza el archivo **.exe** (ej., **Git-2.xx.x-64-bit.exe**) y ejecútalo.
2. Aparecerá una ventana de seguridad preguntando si deseas permitir que la aplicación haga cambios en tu dispositivo. Haz clic en **Sí**.

Paso 3: Aceptar la Licencia

1. Lee la licencia GNU General Public License y haz clic en **Next >**.

Paso 4: Seleccionar Componentes

1. Deja los componentes por defecto seleccionados (incluyendo Git Bash y Git GUI). Haz clic en **Next >**.

Paso 5: Ajustar tu PATH (¡Importante!)

1. En la pantalla "Adjusting your PATH environment", selecciona la opción recomendada: **Git from the command line and also from 3rd-party software**. Esta opción añade Git a tu variable de entorno PATH, permitiéndote usar los comandos de Git desde cualquier terminal de Windows (CMD, PowerShell) y desde otros programas.

2. Haz clic en **Next** >.

Paso 6: Configurar el Salto de Línea

1. En la pantalla "Configuring the line ending conversions", mantén la opción por defecto: **Checkout Windows-style, commit Unix-style line endings**. Esto es importante para evitar problemas de compatibilidad al colaborar entre diferentes sistemas operativos.
2. Haz clic en **Next** >.

Paso 7: Seleccionar el Emulador de Terminal (MinTTY por defecto)

1. Mantén la opción por defecto: **Use MinTTY (the default terminal of MSYS2)**. MinTTY es la terminal predeterminada para Git Bash y ofrece una mejor experiencia de usuario que la consola CMD estándar.
2. Haz clic en **Next** >.

Paso 8: Seleccionar Opciones Adicionales

1. Las opciones por defecto suelen ser adecuadas. Haz clic en **Install**.

Paso 9: Finalizar la Instalación

1. Una vez completada la instalación, puedes desmarcar "View Release Notes" y haz clic en **Finish**.

Paso 10: Verificar la Instalación

1. Abre una nueva ventana de **Git Bash** (busca en el menú de inicio) o tu **Símbolo del Sistema (CMD)** o **PowerShell**.
2. Escribe el siguiente comando y presiona **Enter**:

Unset

```
git --version
```

3. Si la instalación fue exitosa, verás la versión de Git instalada (ej., **git version 2.xx.x.windows.x**).

2. Instalación de Git en macOS

En macOS, puedes instalar Git de varias maneras. Las más comunes son a través de las Herramientas de Línea de Comandos de Xcode o usando Homebrew.

Opción 1: Instalar Git a través de Herramientas de Línea de Comandos de Xcode (Recomendado y Fácil)

Esta es la forma más sencilla, ya que Apple preinstala Git como parte de las herramientas de desarrollo.

Paso 1: Abrir la Terminal

1. Abre la aplicación **Terminal** (puedes encontrarla en **Aplicaciones > Utilidades > Terminal**, o buscarla con Spotlight **Cmd + Espacio**).

Paso 2: Ejecutar el Comando de Instalación de Xcode

1. En la terminal, escribe el siguiente comando y presiona **Enter**:

Unset

```
xcode-select --install
```

Paso 3: Seguir el Asistente de Instalación

1. Aparecerá una ventana de diálogo preguntándote si deseas instalar las herramientas de línea de comandos. Haz clic en **Instalar**.
2. Acepta los términos y condiciones si se te solicita.
3. Espera a que la descarga e instalación se completen. Esto puede tardar unos minutos.

Paso 4: Verificar la Instalación

1. Una vez que el proceso finalice, puedes verificar la instalación de Git escribiendo en la misma terminal:

Unset

```
git --version
```

2. Si la instalación fue exitosa, verás la versión de Git instalada (ej., `git version 2.xx.x` (Apple Git-xxx)).

Opción 2: Instalar Git a través de Homebrew (Alternativa Potente)

Homebrew es un popular gestor de paquetes para macOS que simplifica la instalación de software. Si ya lo tienes, es una excelente opción.

Paso 1: Instalar Homebrew (si no lo tienes)

1. Si no tienes Homebrew, abre la terminal y ejecuta:

Unset

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

2. Sigue las instrucciones en pantalla, incluyendo la introducción de tu contraseña de usuario si se te solicita.

Paso 2: Instalar Git con Homebrew

1. Una vez que Homebrew esté instalado, simplemente ejecuta en la terminal:

Unset

```
brew install git
```

Paso 3: Verificar la Instalación

1. Verifica la instalación de Git:

Unset

```
git --version
```

2. Verás la versión de Git instalada (ej., `git version 2.xx.x`).

3. Instalación de Git en Linux

La instalación de Git en Linux es generalmente muy sencilla y se realiza a través del gestor de paquetes de tu distribución.

Paso 1: Abrir la Terminal

1. Abre tu aplicación de Terminal.

Paso 2: Instalar Git según tu Distribución

- **Para distribuciones basadas en Debian (Ubuntu, Mint, Debian):**
 1. Primero, actualiza la lista de paquetes para asegurar que tienes acceso a las últimas versiones:

Unset

```
sudo apt update
```

2. Luego, instala Git:

Unset

```
sudo apt install git
```

- **Para distribuciones basadas en Fedora (Fedora, CentOS, RHEL):**

Unset

```
sudo dnf install git # Para Fedora 22 y posteriores  
# 0 para Fedora 21 y anteriores: sudo yum install git
```

- **Para otras distribuciones (ej., Arch Linux, OpenSUSE, etc.):** Consulta la documentación de tu distribución o la página oficial de Git para Linux: <https://git-scm.com/download/linux>. Por lo general, será un comando `sudo <gestor_paquetes> install git`.

Paso 3: Verificar la Instalación

1. Una vez que el comando de instalación ha finalizado, verifica que Git se ha instalado correctamente:

Unset

```
git --version
```

2. Verás la versión de Git instalada (ej., `git version 2.xx.x`).

4. Configuración Inicial Global de Git (para todos los Sistemas Operativos)

Después de instalar Git, es fundamental realizar una configuración inicial global. Esto le dirá a Git quién eres cuando realices commits, y esta información se adjuntará a cada cambio que registres.

Paso 1: Configurar tu Nombre de Usuario

1. En tu terminal (Git Bash en Windows, Terminal en macOS/Linux), escribe:

Unset

```
git config --global user.name "Tu Nombre Completo"
```

2. Reemplaza `"Tu Nombre Completo"` con el nombre que deseas que aparezca en tus commits (ej., `"Ana García"`).

Paso 2: Configurar tu Dirección de Correo Electrónico

1. A continuación, configura tu dirección de correo electrónico:

Unset

```
git config --global user.email "tu.email@example.com"
```

2. Reemplaza "`tu.email@example.com`" con tu dirección de correo real. Es recomendable usar el mismo correo que utilizarás en plataformas como GitHub o GitLab.

Paso 3: Verificar la Configuración

1. Para ver tu configuración global de Git (y otras configuraciones predeterminadas), ejecuta:

Unset

```
git config --list
```

2. Deberías ver `user.name` y `user.email` en la lista.

Paso 4: Configurar el Editor de Texto por Defecto (Opcional, pero Recomendado)

Git a veces necesita abrir un editor de texto (para mensajes de commit largos, resolución de conflictos, etc.). Por defecto, suele usar `Vim`, que puede ser complejo. Puedes configurarlo para usar tu editor de código preferido, como Visual Studio Code.

- **Para Visual Studio Code (asegúrate de que el comando `code` esté disponible en tu terminal):**

Unset

```
git config --global core.editor "code --wait"
```

- El `--wait` es importante para que Git espere a que cierres el editor antes de continuar.
- **Para Nano (un editor de terminal más sencillo):**

Unset

```
git config --global core.editor "nano"
```

Conclusión

¡Felicidades! Has instalado y configurado Git correctamente en tu sistema. Ahora estás listo para empezar a versionar tus proyectos, colaborar con otros y explorar todas las potentes funcionalidades que Git ofrece. Recuerda que la práctica es clave para dominar esta herramienta fundamental en el mundo del desarrollo de software y la ciencia de datos.