



# Git commands: los comandos de Git más importantes

Con los comandos de Git adecuados, puedes utilizar el sistema de control de versiones de forma realmente eficaz. Aquí te mostramos los comandos más importantes. Los comandos de Git se ejecutan a través de la terminal.

## Índice

1. Los comandos de Git te facilitan el trabajo
2. Antes de trabajar con Git
3. Configurar el nombre y el correo electrónico con comandos de Git
4. Visión general y cambios
5. Comandos de Git para repositorios
6. Comandos de Git para las ramas
7. Comandos de Git para las etiquetas
8. Utilizar Git de la mejor forma

---

## Dominios web baratos

Dominios tan originales como tus ideas.

Registra tu dominio con IONOS y disfruta de las funciones integrales que tenemos para ofrecerte.

- ✓ Correo incluido
- ✓ Certificado SSL
- ✓ Asistencia 24/7

Continúa

## Los comandos de Git te facilitan el trabajo

Como **sistema de control de versiones**, Git es una herramienta importante para todos los desarrolladores. Optimiza el flujo de trabajo para **pequeños y grandes equipos que quieren colaborar en un proyecto al mismo tiempo**, y proporciona la seguridad y estabilidad necesarias. Git desempeña un papel especialmente importante cuando desarrollas en equipo y participas en la creación de diferentes ramas y repositorios, ya que permite que cada persona pueda hacer un seguimiento de las incorporaciones y cambios. Los comandos de Git son esenciales para trabajar con el sistema, porque con ellos puedes seguir los pasos necesarios para utilizar Git de forma óptima. Aquí encontrarás los comandos de Git más importantes.

## Antes de trabajar con Git

Antes de empezar un nuevo proyecto, es buena idea comprobar si ya has instalado Git y qué versión tienes. El comando apropiado es:



```
1 | git --version
```

## Digital Guide



Si no te muestran ningún número de versión, debes instalar Git manualmente. Para ello utiliza el gestor de paquetes en Linux (<https://www.ionos.es/digitalguide/servidores/know-how/linux-una-alternativa-a-windows-de-bajo-coste/>). En Mac la instalación funciona a través de la terminal y en Windows has de descargar Git manualmente y usar el programa de instalación.

Para iniciar un nuevo proyecto para Git, abre una carpeta en la terminal y configura un nuevo repositorio con el siguiente comando:

```
1 | git init
```

Si quieres copiar un repositorio ya creado o eliminado, y pegarlo en la carpeta, usa el comando git clone (<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/git-clone/>):

```
1 | git clone /ruta/a/repositorio/original
```

```
1 | git clone https://ejemplogit.es/usuario/repositorio.git
```

Si ya has creado una clave SSH, también puedes usar:

```
1 | git clone usuario@servidor:/ruta.git
```

## Configurar el nombre y el correo electrónico con comandos de Git

Para trabajar en un proyecto, necesitas un nombre de usuario y una dirección de correo electrónico válida. Para ello, usa los siguientes comandos de Git.

Configura tu nombre de usuario:

```
1 | git config --global user.name "Nombre"
```

Comprueba tu nombre de usuario:

```
1 | git config --global user.name
```

Escribe tu dirección de correo electrónico:

```
1 | git config --global user.email "correo@ejemplo.es"
```

Para comprobar tu dirección de correo electrónico, teclea:

```
1 | git config --global user.email
```



Para ver todas tus entradas en la vista general, usa este comando:

```
1 | git config --global --list
```

# Visión general y cambios

## Digital Guide



Algunos comandos de Git importantes te facilitan trabajar en un repositorio de Git. Para una visión general del repositorio usa este comando:

```
1 | git clone ejemploit@ejemplo.es:repositorio.git
```

Puedes ver el estado local y de los cambios que aún no se han aceptado:

```
1 | git status
```

Los cambios aparecen resaltados en rojo.

Comprueba las diferencias entre los archivos modificados pendientes de confirmar mediante commit y la última versión actual con git diff (<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/git-diff/>):

```
1 | git diff HEAD
```

## Comandos de Git para repositorios

Si quieres guardar los cambios en el repositorio local directamente, necesitas los siguientes comandos de Git.

Añade todos los archivos nuevos, modificados o eliminados con este comando:

```
1 | git add
```

Si solo quieres aplicar ciertos cambios a tu commit, ponlos entre corchetes después del comando:

```
1 | git add [ archivo_1 archivo_2 archivo_3 ]
```

Para finalizar un git commit (<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/git-commit/>) usa:

```
1 | git commit
```

Si quieres añadir explicaciones adicionales sobre el paso actual, puedes hacerlo así:

```
1 | git commit -m "Tu mensaje"
```

Con git log (<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/git-log/>) puedes ver los commit actuales en tu repositorio local:

```
1 | git log
```



## Comandos de Git para las ramas

Para trabajar con una rama, necesitas los siguientes comandos de Git.

Enumerar todas las ramas:

Digital Guide

```
1 | git branch
```



(<https://www.ionos.es/digitalguide/>)

Informar sobre las diferentes ramas:

```
1 | git fetch
```

Enumerar todas las ramas del repositorio local:

```
1 | git branch -a
```

Crear una nueva rama:

```
1 | git branch nuevo-nombre-de-rama
```

Cambiar a una rama específica con git checkout (<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/git-checkout/>):

```
1 | git checkout nombre-de-otra_rama
```

Crear una nueva rama y pasar a trabajar en ella:

```
1 | git checkout -b nueva-rama
```

Mover la nueva rama del repositorio local al repositorio del commit:

```
1 | git push nombre_remoto nueva_rama
```

Eliminar una rama en el repositorio local si solo contiene los cambios adoptados:

```
1 | git branch -d nombre_de_rama
```

Añadir los cambios de una rama a la rama actual:

```
1 | git merge nombre_de_otra_rama
```

Recuperar los cambios de un repositorio remoto con git pull (<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/git-pull/>):

```
1 | git pull otra_rama
```

## Comandos de Git para las etiquetas



Si utilizas etiquetas, estos comandos de Git te facilitan el trabajo.

Lista de todas las etiquetas:

Recuperar todas las etiquetas del repositorio de commit (<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/git-fetch>):

1 | git fetch --tags

Mostrar una etiqueta específica:

1 | git show nombre\_de\_etiqueta

Mover una etiqueta específica al repositorio de commit con git push (<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/git-push/>):

1 | git push nombre\_remoto ejemplo\_de\_etiqueta

Mover todas las etiquetas en el repositorio de commit:

1 | git push nombre\_remoto --tags

Eliminar la etiqueta en el repositorio local:

1 | git tag -d ejemplo\_de\_etiqueta

## Utilizar Git de la mejor forma

Con estos comandos de Git trabajarás de forma rápida y eficaz en el sistema de control de versiones. En nuestra Digital Guide también encontrarás un tutorial de Git ([https://www.ionos.es/digitalguide/paginas-web/tutorial-de-git/](https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/tutorial-de-git/)) para dar los primeros pasos, así como un resumen de Git en PDF para descargar (<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/git-cheat-sheet/>). Si buscas una alternativa al sistema de Linus Torvalds, nuestra comparación de Git y SVN (<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/git-vs-svn-una-comparativa-del-control-de-versiones/>) te resultará interesante.

### 💡 Consejo

Usa Deploy Now (<https://www.ionos.es/alojamiento/deploy-now?ac=OM.WE.WEo50K361688T7073a&itc=5L4C4XEV-5ICVBT-38NS5G8>) de IONOS para hacer que tu proyecto esté en línea en solo tres pasos. Tanto las webs estáticas como las apps de una sola página estarán disponibles para tus clientes aún más rápido.

