



--distributed-even-if-your-workflow-isnt



- [About](#)
 - [Branching and Merging](#)
 - [Small and Fast](#)
 - [Distributed](#)
 - [Data Assurance](#)
 - [Staging Area](#)
 - [Free and Open Source](#)
 - [Trademark](#)
- [Documentation](#)
 - [Reference](#)
 - [Book](#)
 - [Videos](#)
 - [External Links](#)
- [Downloads](#)
 - [GUI Clients](#)
 - [Logos](#)
- [Community](#)

This book is available in [English](#).

Full translation available in

[azərbaycan dili](#),
[български език](#),
[Deutsch](#),
[Español](#),
[Français](#),
[Ελληνικά](#),
[日本語](#),
[한국어](#),
[Nederlands](#),
[Русский](#),
[Slovenščina](#),
[Tagalog](#),
[Українська](#),
[简体中文](#),

Partial translations available in

[Čeština](#),
[Македонски](#),
[Polski](#),
[Српски](#),
[Ўзбекча](#),
[繁體中文](#),

Translations started for

[Беларуская](#),
[فارسی](#),
[Indonesian](#),
[Italiano](#),
[Bahasa Melayu](#),
[Português \(Brasil\)](#),
[Português \(Portugal\)](#),
[Svenska](#),
[Türkçe](#).

The source of this book is [hosted on GitHub](#).
Patches, suggestions and comments are welcome.
[Chapters](#) ▼

1. [1. Inicio - Sobre el Control de Versiones](#)

- 1. 1.1 [Acerca del Control de Versiones](#)
- 2. 1.2 [Una breve historia de Git](#)
- 3. 1.3 [Fundamentos de Git](#)
- 4. 1.4 [La Línea de Comandos](#)
- 5. 1.5 [Instalación de Git](#)
- 6. 1.6 [Configurando Git por primera vez](#)
- 7. 1.7 [¿Cómo obtener ayuda?](#)
- 8. 1.8 [Resumen](#)

2. [2. Fundamentos de Git](#)

- 1. 2.1 [Obteniendo un repositorio Git](#)
- 2. 2.2 [Guardando cambios en el Repositorio](#)
- 3. 2.3 [Ver el Historial de Confirmaciones](#)
- 4. 2.4 [Deshacer Cosas](#)
- 5. 2.5 [Trabajar con Remotos](#)
- 6. 2.6 [Etiquetado](#)
- 7. 2.7 [Alias de Git](#)
- 8. 2.8 [Resumen](#)

3. [3. Ramificaciones en Git](#)

- 1. 3.1 [¿Qué es una rama?](#)
- 2. 3.2 [Procedimientos Básicos para Ramificar y Fusionar](#)
- 3. 3.3 [Gestión de Ramas](#)
- 4. 3.4 [Flujos de Trabajo Ramificados](#)
- 5. 3.5 [Ramaz Remotas](#)
- 6. 3.6 [Reorganizar el Trabajo Realizado](#)
- 7. 3.7 [Recapitulación](#)

4. [4. Git en el Servidor](#)

- 1. 4.1 [Los Protocolos](#)
- 2. 4.2 [Configurando Git en un servidor](#)

- 3. 4.3 [Generando tu clave pública SSH](#)
- 4. 4.4 [Configurando el servidor](#)
- 5. 4.5 [El demonio Git](#)
- 6. 4.6 [HTTP Inteligente](#)
- 7. 4.7 [GitWeb](#)
- 8. 4.8 [GitLab](#)
- 9. 4.9 [Git en un alojamiento externo](#)
- 10. 4.10 [Resumen](#)

5. [Git en entornos distribuidos](#)

- 1. 5.1 [Flujos de trabajo distribuidos](#)
- 2. 5.2 [Contribuyendo a un Proyecto](#)
- 3. 5.3 [Manteniendo un proyecto](#)
- 4. 5.4 [Resumen](#)

1. [6. GitHub](#)

- 1. 6.1 [Creación y configuración de la cuenta](#)
- 2. 6.2 [Participando en Proyectos](#)
- 3. 6.3 [Mantenimiento de un proyecto](#)
- 4. 6.4 [Gestión de una organización](#)
- 5. 6.5 [Scripting en GitHub](#)
- 6. 6.6 [Resumen](#)

2. [7. Herramientas de Git](#)

- 1. 7.1 [Revisión por selección](#)
- 2. 7.2 [Organización interactiva](#)
- 3. 7.3 [Guardado rápido y Limpieza](#)
- 4. 7.4 [Firmando tu trabajo](#)
- 5. 7.5 [Buscando](#)
- 6. 7.6 [Reescribiendo la Historia](#)
- 7. 7.7 [Reiniciar Desmitificado](#)
- 8. 7.8 [Fusión Avanzada](#)
- 9. 7.9 [Rerere](#)
- 10. 7.10 [Haciendo debug con Git](#)
- 11. 7.11 [Submódulos](#)
- 12. 7.12 [Agrupaciones](#)
- 13. 7.13 [Replace](#)
- 14. 7.14 [Almacenamiento de credenciales](#)
- 15. 7.15 [Resumen](#)

3. [8. Personalización de Git](#)

- 1. 8.1 [Configuración de Git](#)
- 2. 8.2 [Git Attributes](#)
- 3. 8.3 [Puntos de enganche en Git](#)
- 4. 8.4 [Un ejemplo de implantación de una determinada política en Git](#)
- 5. 8.5 [Recapitulación](#)

4. [9. Git y Otros Sistemas](#)

- 1. 9.1 [Git como Cliente](#)

- 2. 9.2 [Migración a Git](#)
- 3. 9.3 [Resumen](#)

5. **10. Los entresijos internos de Git**

- 1. 10.1 [Fontanería y porcelana](#)
- 2. 10.2 [Los objetos Git](#)
- 3. 10.3 [Referencias Git](#)
- 4. 10.4 [Archivos empaquetadores](#)
- 5. 10.5 [Las especificaciones para hacer referencia a... \(refspec\)](#)
- 6. 10.6 [Protocolos de transferencia](#)
- 7. 10.7 [Mantenimiento y recuperación de datos](#)
- 8. 10.8 [Variables de entorno](#)
- 9. 10.9 [Recapitulación](#)

1. **A1. Apéndice A: Git en otros entornos**

- 1. A1.1 [Interfaces gráficas](#)
- 2. A1.2 [Git en Visual Studio](#)
- 3. A1.3 [Git en Eclipse](#)
- 4. A1.4 [Git con Bash](#)
- 5. A1.5 [Git en Zsh](#)
- 6. A1.6 [Git en Powershell](#)
- 7. A1.7 [Resumen](#)

2. **A2. Apéndice B: Integrando Git en tus Aplicaciones**

- 1. A2.1 [Git mediante Línea de Comandos](#)
- 2. A2.2 [Libgit2](#)
- 3. A2.3 [JGit](#)

3. **A3. Apéndice C: Comandos de Git**

- 1. A3.1 [Configuración](#)
- 2. A3.2 [Obtener y Crear Proyectos](#)
- 3. A3.3 [Seguimiento Básico](#)
- 4. A3.4 [Ramificar y Fusionar](#)
- 5. A3.5 [Compartir y Actualizar Proyectos](#)
- 6. A3.6 [Inspección y Comparación](#)
- 7. A3.7 [Depuración](#)
- 8. A3.8 [Parcheo](#)
- 9. A3.9 [Correo Electrónico](#)
- 10. A3.10 [Sistemas Externos](#)
- 11. A3.11 [Administración](#)
- 12. A3.12 [Comandos de Fontanería](#)

2nd Edition

1.6 Inicio - Sobre el Control de Versiones - Configurando Git por primera vez

Configurando Git por primera vez

Ahora que tienes Git en tu sistema, vas a querer hacer algunas cosas para personalizar tu entorno de Git. Es necesario hacer estas cosas solamente una vez en tu computadora, y se mantendrán entre actualizaciones. También puedes cambiarlas en cualquier momento volviendo a ejecutar los comandos correspondientes.

Git trae una herramienta llamada `git config`, que te permite obtener y establecer variables de configuración que controlan el aspecto y funcionamiento de Git. Estas variables pueden almacenarse en tres sitios distintos:

1. Archivo `/etc/gitconfig`: Contiene valores para todos los usuarios del sistema y todos sus repositorios. Si pasas la opción `--system` a `git config`, lee y escribe específicamente en este archivo.
2. Archivo `~/.gitconfig` o `~/.config/git/config`: Este archivo es específico de tu usuario. Puedes hacer que Git lea y escriba específicamente en este archivo pasando la opción `--global`.
3. Archivo `config` en el directorio de Git (es decir, `.git/config`) del repositorio que estés utilizando actualmente: Este archivo es específico del repositorio actual.

Cada nivel sobrescribe los valores del nivel anterior, por lo que los valores de `.git/config` tienen preferencia sobre los de `/etc/gitconfig`.

En sistemas Windows, Git busca el archivo `.gitconfig` en el directorio `$HOME` (para mucha gente será `C:\Users\%USER\`). También busca el archivo `/etc/gitconfig`, aunque esta ruta es relativa a la raíz `MSys`, que es donde decidiste instalar Git en tu sistema Windows cuando ejecutaste el instalador.

Tu Identidad

Lo primero que deberás hacer cuando instales Git es 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:

```
$ git config --global user.name "John Doe"
$ git config --global user.email johndoe@example.com
```

De nuevo, 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.

Muchas de las herramientas de interfaz gráfica te ayudarán a hacer esto la primera vez que las uses.

Tu Editor

Ahora que tu identidad está configurada, puedes elegir el editor de texto por defecto que se utilizará cuando Git necesite que introduzcas un mensaje. Si no indicas nada, Git usará el editor por defecto de tu sistema, que generalmente es Vim. Si quieres usar otro editor de texto como Emacs, puedes hacer lo siguiente:

```
$ git config --global core.editor emacs
```

Vim y Emacs son editores de texto frecuentemente usados por desarrolladores en sistemas basados en Unix como Linux y Mac. Si no estás familiarizado con ninguno de estos editores o estás en un sistema Windows, es posible que necesites buscar instrucciones acerca de cómo configurar tu editor favorito con Git. Si no configuras un editor así y no conoces acerca de Vim o Emacs, es muy factible que termines en un estado bastante confuso en el momento en que sean ejecutados.

Comprobando tu Configuración

Si quieres comprobar tu configuración, puedes usar el comando `git config --list` para mostrar todas las propiedades que Git ha configurado:

```
$ git config --list
user.name=John Doe
user.email=johndoe@example.com
color.status=auto
color.branch=auto
color.interactive=auto
color.diff=auto
...
```

Puede que veas claves repetidas, porque Git lee la misma clave de distintos archivos (`/etc/gitconfig` y `~/.gitconfig`, por ejemplo). En estos casos, Git usa el último valor para cada clave única que ve.

También puedes comprobar el valor que Git utilizará para una clave específica ejecutando `git config <key>`:

```
$ git config user.name
John Doe
```

[prev](#) | [next](#)

[About this site](#)

Patches, suggestions, and comments are welcome.

Git is a member of [Software Freedom Conservancy](#).