



--distributed-is-the-new-centralized

- [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](#),  
[\\_\\_\\_\\_\\_](#),  
[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 [Rammas 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. [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

## 4.3 Git en el Servidor - Generando tu clave pública SSH

### Generando tu clave pública SSH

Tal y como se ha comentado, muchos servidores Git utilizan la autenticación a través de claves públicas SSH. Y, para ello, cada usuario del sistema ha de generarse una, si es que ya no la tiene. El proceso para hacerlo es similar en casi cualquier sistema operativo. Ante todo, asegúrate que no tengas ya una clave. Por defecto, las claves de cualquier usuario SSH se guardan en la carpeta `~/.ssh` de dicho usuario. Puedes verificar si ya tienes unas claves, simplemente situándote sobre dicha carpeta y viendo su contenido:

```
$ cd ~/.ssh
$ ls
authorized_keys2  id_dsa          known_hosts
config            id_dsa.pub
```

Has de buscar un par de archivos con nombres tales como *algo* y *algo.pub*; siendo ese "algo" normalmente *id\_dsa* o *id\_rsa*. El archivo terminado en *.pub* es tu clave pública, y el otro archivo es tu clave privada. Si no tienes esos archivos (o no tienes ni siquiera la carpeta *.ssh*), has de crearlos; utilizando un programa llamado *ssh-keygen*, que viene incluido en el paquete SSH de los sistemas Linux/Mac o en el paquete MSysGit en los sistemas Windows:

```
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/schacon/.ssh/id_rsa):
Created directory '/home/schacon/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/schacon/.ssh/id_rsa.
Your public key has been saved in /home/schacon/.ssh/id_rsa.pub.
The key fingerprint is:
d0:82:24:8e:d7:f1:bb:9b:33:53:96:93:49:da:9b:e3 schacon@mylaptop.local
```

Como se ve, este comando primero solicita confirmación de dónde van a guardarse las claves (*.ssh/id\_rsa*), y luego solicita, dos veces, una contraseña (passphrase), contraseña que puedes dejar en blanco si no deseas tener que teclearla cada vez que uses la clave.

Tras generarla, cada usuario ha de encargarse de enviar su clave pública a quienquiera que administre el servidor Git (en el caso de que éste esté configurado con SSH y así lo requiera). Esto se puede realizar simplemente copiando los contenidos del archivo terminado en *.pub* y enviándoselos por correo electrónico. La clave pública será una serie de números, letras y signos, algo así como esto:

```
$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAklOUpKDHrfHY17SbrmTIpNLTGK9Tjom/BWDSU
GPL+nafzlhDTYw7hdI4yZ5ew18JH4JW9jhbUFRviQzM7xLELEVf4h9lFX5QVkbPppSwg0cda3
Pbv7k0dJ/MTyB1WxFCR+HAo3FXRitBqxiX1nKhXpHAZsMcilQ8V6RjsNAQwdsdMFvSLVK/7XA
t3FaoJoAsncM1Q9x5+3V0Ww68/eIFmb1zuUFljQJKprX88XypNDvjYNby6vw/Pb0rwert/En
```

```
mZ+AW40ZPnTPI89ZPmVMLuayrD2cE86Z/il8b+gw3r3+1nKatmIkjn2so1d01QraTlMqVSsbx  
NrRFi9wrf+M7Q== schacon@mylaptop.local
```

Para más detalles sobre cómo crear unas claves SSH en variados sistemas operativos, consultar la correspondiente guía en GitHub: <https://help.github.com/articles/generating-ssh-keys>.

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

[About this site](#)

Patches, suggestions, and comments are welcome.

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