# Git tutorial

*Publicado el sábado 9 de abril de 2016.*
*Actualizado el sábado 28 de marzo de 2020.*

Por medio de Twitter uno se entera de muchas cosas, dejando a un lado el maligno cotilleo, claro está. Una de esas cosas buenas es conocer que existe "Pro Git Book" el cual fue escrito por Scott Chacon (empleado hoy en día en la empresa GitHub) y Ben Straub (quien ahora trabaja en la empresa Zendesk) sobre el famoso **sistema de administración (de versiones) de código fuente** que tal como su nombre pomposo indica, es un software orientado a mantener en orden nuestros "apuntes" (ea , que estamos en el siglo XXI y ya debemos dejar de hacer esto, *que yo también lo hacía, perdonadme Romero*):

https://twitter.com/romero/status/716935957656813568

## Introducción

Esta entrada tiene como objetivo la instalación de Git en nuestros ordenadores y su aprendizaje básico teniendo como guía de escritura al famoso libro "Pro Git" 2° edición, osea un Git tutorial en toda regla. Luego veremos las opciones avanzadas, de las cuales muchos prescindiréis, especialmente la instalación de nuestros propio servidores de repositorios, pero no os quiero abrumar con más detalles, entremos en materia y recordad que si queréis ir más lentamente descargad y leed el libro (está escrito bajo licencia *GNU General Public License*) ***o mejor aún,*** compradlo ***y colaborad con los autores y además contribuid a otra buena causa.***

### Mini tutorial Git y chuleta de comandos

Para aquellos de vosotros que os pasáis apurados todo el tiempo os recomiendo un resumen realizado por "El Baúl del Programador" el cual contiene, basado en el libro ProGit, los comandos básicos agrupados por similaridad de funciones junto con ejemplos prácticos de los escenarios

que regularmente se nos presentan, he aquí el enlace al artículo y la cuenta en Twitter por si queréis seguirlos:

https://twitter.com/ElBaulP/status/722357570967572480

## Brevísima historia de Git.

Git logo.

Ya en 1998 el equipo de desarrollo del sistema operativo GNU/Linux, encabezado por Linus Torvalds, observó lo difícil que era llevar bajo control el crecimiento (debido a que cada día se unían más entusiastas a la causa del software libre) del novedoso proyecto. Tanto es así que en 2002 se decide encomendar a la empresa BitMover -recién fundada dos años antes- la tarea de

mantener "en custodia", por medio de su producto estrella BitKeeper, las valiosísimas líneas de código del mayor emprendimiento hecho sobre software libre hasta la fecha. Al padre Richard Stallman por supuesto que no le gustó para nada este software "semi-privativo" (el adjetivo es inventado por nosotros, si es que no existía o existe ese concepto) ya que, si a ver vamos, era sin valor monetario para la comunidad en desarrollo *a costa de comprometerse a no utilizar ninguna otro software para control de versiones -privativo o libre- mientras utilizaran BitKeeper e incluso un años más allá después de haberlo dejado de usar (menos mal tenían acceso al código fuente de BitKeeper).* **Tampoco podían participar en el desarrollo de algún software para control de versiones nuevo *y como guinda para el pastel debían ser informados de manera automática de los cambios al GNU/Linux (con ayuda de los propios servidores de BitMover).***

A muchos programadores esto no fue para nada de su agrado, **una de las principales características de la filosofía del conocimiento abierto es precisamente la LIBERTAD.** A pesar que el propio Linus Torvalds hizo llamados públicos y notorios a quien tuviera una alternativa totalmente bajo licencia GNU, nadie respondió ni ofreció ninguna herramienta *-de hecho, hay que tener* cojones *para desarrollar software para creadores de software: estaís en el* ojo del huracán *todo el tiempo-*. Mientras tanto intentaron edulcorar la agria relación con parches que hicieran compatible los servidores de BitMover con "Concurrent Version Systems" y "Subversion" pero el terrible cotilleo por internet que menciono al principio de este artículo echó por la borda cualquier intento de progreso.

En el año de 2005 la empresa BitMover amagó con empezar a cobrar por la licencia de uso pero Andrew Tridgell (padre también del rsync y Samba Server) ya estaba "prevenido al bate": en aquella época él era empleado de "Open Source Development Labs" (OSDL) -predecesora de lo que hoy es "The Linux Foundation"- y tenía un as bajo la manga: "SourcePuller".

Como mencionamos Andrew Tridgell fue padre del software Samba, el cual es basado -y ampliado- en el protocolo de recursos compartidos de la empresa Microsoft bajo su sistema operativo Windows *-empresa que, por demás, nunca hizo queja alguna sobre ese invento hecho bajo software libre-* y aplicó la misma técnica sobre el BitKeeper e inventó el "SourcePuller". Pero se extralimitó ya que contrato leonino impuesto por BitMover sobre OSDL prohibía expresamente cualquier mejora (hasta el propio Linus Torvalds quedó salpicado en el asunto, al ser empleado también de OSDL tampoco podía participar en ningún desarrollo de software de control de versiones). **Es de nuestra opinión que BitMover ya estaba ansiosa de obtener dividendos sobre su inversión y vieron en el trabajo de Andrew la excusa perfecta -violación de contrato- para cobrar por la licencia de uso *y no al contrario -hay una extensa discusión sobre el tema (ver abajo enlaces de las fuentes) pero escapa de este humilde tutorial-.***

Es por tanto que de todo lo anterior nace Git en el año 2005, y evidentemente que Linus Torvalds fue el titiritero detrás de todo el asunto: una vez le fue consultado sobre el origen del nombre "Git" a lo cual respondió (recordad que él es finlandés y por cercanía geográfica aprendió el inglés británico antes de ser ciudadano estadounidense; las fuentes coloreadas son mis observaciones):

- Una combinación de tres letras impronunciables y que no son ningún comando en Unix. El hecho que provenga de una mala pronunciación de "get" (obtener) es irrelevante (deriva del inglés antiguo; "get" se pronuncia rápidamente mientras que "git" se alarga, es casi lo mismo pero a diferentes velocidades).
- Estúpido. Vil y despreciable. Sencillo. Escoja usted del diccionario de jergas (de hecho en el inglés antiguo significaba "bastardo").
- "Información de Seguimiento Global": si usted tiene un buen estado de ánimo, y realmente funciona para usted. Los ángeles cantan, y un resplandor llena de repente su habitación.
- "Maldito camión cargado de mierda": cuando se echa a perder (por las iniciales en idioma inglés "Goddamn Idiotic Truckload of shit").

Como vemos el señor Linus Torvalds es lo que nosotros por acá llamamos ***una verdadera chupeta de ajo. Es por ello que tomamos aire y decimos a todo pulmón "VERGACIÓN LINUS ERES UNA PERSONA REALMENTE JODIDA" ja ja ja. ;-)***

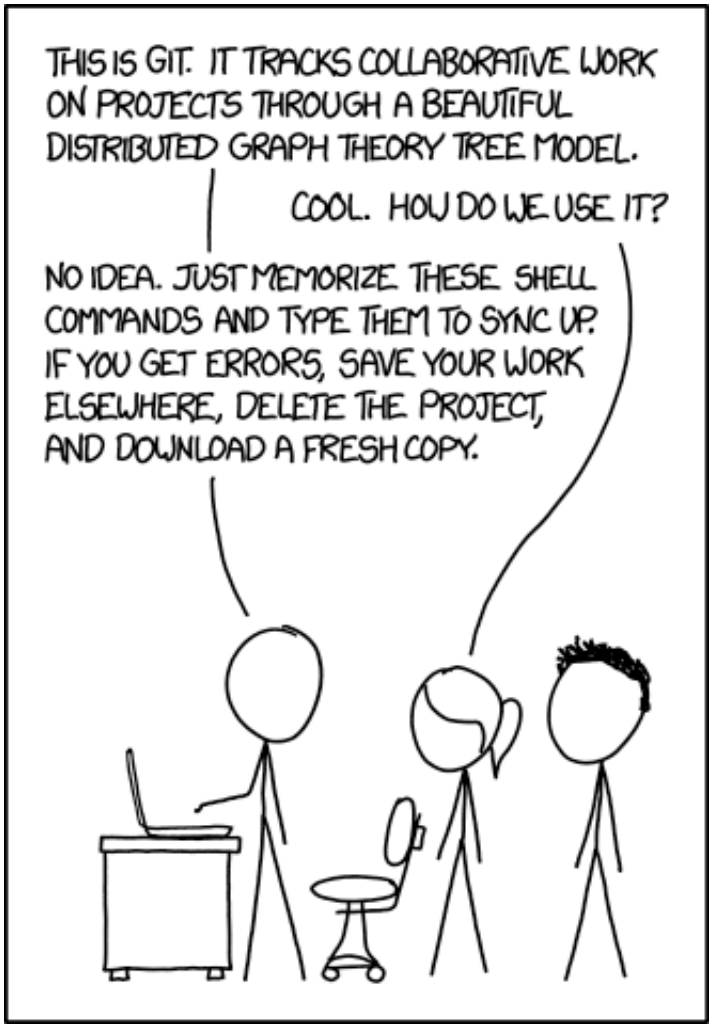
Para que podáis entender el panorama informático al momento del nacimiento de Git, en este mensaje de Linus Torvalds a al comunidad, fechado el miércoles 6 de abril de 2005 (en idioma inglés) están los detalles del punto de inflexión en la herramienta para desarrollar el kernel del GNU/Linux -y fue para mejorar, indudablemente-.

A partir del nacimiento de Git, Linus Torvalds designó al japonés e ingeniero en software Junio C. Hamano (?? ?) el mantenimiento y desarrollo de Git. Hamano, quien es empleado de la empresa Google llevaba un blog hasta 2011 donde regularmente informaba amablemente al público en general sobre su trabajo pero luego decidió, para su paz mental, mudarlo al principal rival de Wordpress: Blogspot (empresa comprada por Google en el año 2003).

https://twitter.com/jch2355/status/586754865465991171

El útimo anuncio en dicho blog describe el lanzamiento de la versión 2.5 de Git en julio de 2015 aunque a la fecha de escribir estas lineas están trabajando en la versión 2.8:

https://twitter.com/jch2355/status/895709811672662017




This is Git. It tracks collaborative work on projects on through a beautiful distributed graph theory tree model.

De la propia página personal de Hamano observamos una referencia a la anterior imagen, *lo cual es una clara señal de que el hombre se toma las cosas con mucha, muchísima calma, a diferencia mayúscula de su jefe Linus Torvalds.*

A la fecha se espera la llegada de la versión 2.9.1, según reseña la página web tutorial de LXALinuxAdictos y os adelanto, si queréis leer el reportaje, que allí expresan una muy resumida historia del Git, directo y al grano. Es bueno el artículo, pero nosotros aquí, como veis, somos de muchas más palabras ;-) .

https://twitter.com/LXALinuxAdictos/status/753537530906210304

Por otra parte, el protocolo **Git** se ha extendido a nivel mundial, tanto es así que el libro que vamos a estudiar ¡ha sido traducido al idioma coreano!

https://twitter.com/benstraub/status/755466002771636224

## Presente y futuro de Git.

El primer caso de éxito para el protocolo y programa Git -y que ya lo hemos nombrado y referenciado bastante- es el caso de GitHub. Es tanto así que en el libro por el cual nos guiamos para esta entrada tiene su capítulo aparte y dedicado. La cuestión es simple: ellos dan alojamiento a proyectos cuyo control de versiones es manejado por medio de Git -por supuesto y lógico- *y dicho alojamiento es libre y gratuito* ***mientras sea público, allí está la "pega" del asunto.*** Si vosotros queréis llevar un proyecto privado y en azul celeste, pues que vuestro dinero os cueste, y por eso cobra y vive GitHub. Claro hay otros privilegios adicionales y se decide comprar alojamiento privado: total confidencialidad del proyecto, respaldos, páginas web, etcétera y cobran de acuerdo al tamaño del proyecto, esto es así porque nadie trabaja gratis, todos tenemos nuestros gastos.

Pero el asunto del GitHub va mucho más allá del ámbito de la programación. Desde escribir un libro (como el que comenzaremos a estudiar aquí) y gestionar su traducción a otros idiomas hasta leyes, control de planificación de ciudades o gente que programa drones, **hay de todo un poco.** Si queréis ahondar en los usos actuales (y tal vez veáis el futuro) os recomiendo leer el excelente reportaje en este enlace, gracias al sitio web "El Baúl del Programador".

## Instalación de Git en Ubuntu.

Superada la jocosa personalidad del padre de Linux, vamos a la parte seria del artículo: Git

tutorial. Estamos aquí por instalar en nuestro ordenador con Ubuntu el programa Git. Esto vale también para la mayoría de distribuciones basadas en Debian. Abrimos una ventana terminal (si necesitan ayuda con esto pueden leer nuestro tutorial sobre bash) y tecleamos lo siguiente:

```
sudo apt-get install git-all
```

Introducimos nuestra contraseña para ganar acceso como usuario maestro "**root**" y podremos observar algo como esto (si desea haga clic en la imagen para ampliarla):


sudo apt-get install git-all

Y listo, eso es todo, luego de descargar aproximadamente 30 megabytes y ocupar 139 megabytes en disco duro a la final se instalará un "gitdaemon", es decir un servicio que estará siempre al pendiente de nuestro proyectos. Más adelante, en la sección avanzada, veremos y ahondaremos en este punto.

## Características de Git.

## Nuestro primer proyecto con Git.

## Ramificaciones (avanzado de aquí en adelante).

## Git en nuestro(s) propio(s) servidore(s).

Hemos realizado traducciones de los artículos publicados por DigitalOcean:

- «Cómo configurar un servidor Git privado en un VPS» por Brian Rogers.
- Cómo configurar un repositorio para ser alojado en nuestros servidores Git privados y público (GitHub).

## Git en entornos distribuidos.

## GitHub.

## Herramientas en Git.

## Personalización de Git.

## Git en otros sistema de control de versiones.

## Asuntos internos en Git.

## Git en otros medioambientes.

## Incrustando Git en tus aplicaciones por medio de la línea de comandos.

## Humor.

## Fuentes consultadas.

### Artículos en idioma castellano:


- ""Una breve historia del Git", 2° edición, escrito por Scott Chacón y Ben Straub (pueden contribuir con sus observaciones y mejoras -en inglés- a la futura 3° edición en este enlace patrocinado por GitHub).
- "Git 2.9.1 ya está listo y renovado con esta nueva versión" por Isaac P.E. en LXALinuxAdictos.
- "La generación GitHub: Por qué ahora todos estamos en el opensource"en El Baúl del Progamador.com.

- "GIT: entrevista a su creador en su décimo aniversario" traducción del artículo original publicado en abril de 2015 en Linux.com
- "21 alias útiles para git" por Alejandro Alcalde.
- «Cifrado de repositorios Git» por Lorenzo "Atareao".
    - «Como Administrar tus Claves GPG en Linux con Seahorse» por Oriol.
- «Borrar archivos de Git» por Lorenzo "Atareao"
- Asegura tu trabajo con «git stash»

## Artículos en idioma inglés:

- «How to Run Your Own Git Server» by Swapnil Bhartiya.
- "BitKeeper" en Wikipedia.
- "Git Software" en Wikipedia.
- "Concurrent Versions System", página oficial.
- "Subversion", página oficial (preguntas frecuentes).
- Discusión sobre BitMover, SourcePuller y Git:
    - "What are some well-written accounts of the Linux-Bitkeeper-Git story?" por Oscar Bonilla, año 2011.
    - "Why a free BitKeeper clone was a bad idea (which time had come)" por Oscar Bonilla, año 2005.
    - "Tridgell drops Bitkeeper bombshell" por Andrew Orlowski.
    - "Tridge speaks" en Grooklaw.
- "Git 2.9 has been released" at Github.
- "Kernel SCM saga..." by Linus Torvalds at mirror LKLM.org
- "«Monotone» is an open source software tool for distributed revision control".
- "The GitHub generation: why we're all in open source now" by Mikeal Rogers.
- "10 Years of Git: An Interview with Git Creator Linus Torvalds" via Linux.com, april 2015.
- December 22, 2005 "GIT 1.0 released" via Gmane.org
- May 28, 2014 "Git 2.0 Officially Released" via Phoronix.com
- "Git: A Tool for Learning Puppet" by Tiffany Longworth.
- "Debian Wheezy Local Git Server With Git Lite" by HowToForge.com
- "How to Run Your Own Git Server" by Swapnil Bhartiya
- Learn git concepts, not commands - a git tutorial
- «Install hub to make your Git command-line as fully featured as GitHub».
- «Manage your dotfiles with Git».

### Cifrado y protección de repositorios

- «Protect secret data in git repo» by Marco Vito Moscaritolo (WayBack Machine).

### Enlaces mediante Youtube (en inglés):

https://www.youtube.com/watch?v=QGKTdL7GG24

### Enlaces mediante Twitter (en inglés):

https://twitter.com/EtnasSoft/status/727455725811814405
https://twitter.com/SaraMG/status/733155433163022340
https://elbauldelprogramador.com/21-aliases-utiles-para-git/
https://elbauldelprogramador.com/como-usar-los-filtros-smudge-y-clean-en-git/
https://elbauldelprogramador.com/sincronizacin-de-proyectos-en-git-con-hooks-ganchos/
https://scotch.io/tutorials/contributing-to-open-source-software-with-git
https://twitter.com/github/status/736325309587296261
https://twitter.com/opensourceway/status/771045139452088320