

# **Informe sobre Git y GitHub**

## **Profesor**

Róbinson Coronado García

## **Alumno**

Oscar Darío Botero Vargas  
C.C. 71'764,308

Universidad de Antioquia  
Curso de Fundamentos de Sistemas de Información  
Medellín  
Septiembre 2020

## Git

Según la introducción del libro *Pro Git* (Chacon y Straub, sin fecha):

GIT es un *Sistema Controlador de Versiones*.

Un *Sistema Controlador de Versiones* (SCV o VCS, del inglés *Version Control System*) es un sistema que registra los cambios hechos a un archivo o a un conjunto de archivos a lo largo del tiempo de tal modo que uno puede recuperar versiones específicas posteriormente.

Un *Sistema Controlador de Versiones* (VCS) nos permite revertir archivos seleccionados a un estado previo, revertir el proyecto entero a un estado previo, comparar cambios a lo largo del tiempo, ver quien modificó, por última vez, algo que podría estar causando problemas, quien introdujo un problema y cuando, y más. El uso de un VCS también significa, generalmente, que si usted arruina las cosas o pierde archivos, puede recuperarlos fácilmente. Además, usted obtiene todo esto por muy poco costo.

Hay tres tipos de sistemas controladores de versiones:

Los ***Sistemas Locales para el Control de Versiones*** (*Local Version Control Systems*, LVCS) los cuales mantienen el historial de cambios realizados a los archivos bajo su dirección, pero no llevan registro de que cambios fueron realizados por cada autor.

Los ***Sistemas Centralizados para el Control de Versiones*** (*Centralized Version Control Systems*, CVCS) los que permiten, al igual que los LVCS, llevar un historial de los cambios realizados a cada archivo bajo su dominio pero, además, permiten que varios usuarios realicen los cambios, lo que facilita que los usuarios trabajen de manera colaborativa en las modificaciones a cada archivo. Estos sistemas llevan todo el historial de cambios en un único servidor al cual los clientes pueden acceder. La ventaja de este tipo de servidores es que permite tener un control de grado fino sobre quien puede hacer que, y es más fácil administrar un CVCS que tratar con varios LVCS o bases de datos locales en cada cliente.

La desventaja principal de los CVCS es que si el servidor deja de funcionar por un tiempo, durante ese lapso nadie podrá colaborar ni guardar cambios a los archivos bajo la regulación de él (del CVCS). La situación se agrava si los archivos en el servidor se pierden, en cuyo caso absolutamente todo se pierde (la historia completa del proyecto exceptuando las copias de las instantáneas individuales que las personas tengan en sus máquinas locales). Los sistemas VCS locales sufren el mismo problema: siempre que tenga el historial completo del proyecto en un solo lugar, corre el riesgo de perderlo todo.

Los ***Sistemas Distribuidos para el Control de Versiones*** (DVCS, *Distributed Version Control Systems*) mantienen una copia de todo el repositorio en cada cliente (incluyendo la historia completa). Por lo tanto, si algún servidor muere y estos sistemas colaboran a través de ese servidor, el repositorio de cualquiera de los clientes puede copiarse en el servidor para restaurarlo. Cada clon es realmente una copia de seguridad completa de todos los datos.

## GitHub

Según el artículo sobre GitHub en Wikipedia (GitHub, 9 de septiembre de 2020):

GitHub, Inc. es una corporación multinacional estadounidense que ofrece alojamiento para el desarrollo de software y el control de versiones mediante Git. Ofrece el control distribuido de

versiones y la funcionalidad de gestión de código fuente (SCM) de Git, además de sus propias características. Proporciona control de acceso y varias funciones de colaboración, como seguimiento de errores, solicitudes de funciones, gestión de tareas y wikis para cada proyecto. Con sede en California, ha sido una subsidiaria de Microsoft desde 2018.

Según el propio GitHub (What is GitHub?, s.f.):

GitHub es una plataforma de alojamiento de código para el control de versiones y la colaboración. Le permite a usted y a otros trabajar juntos en proyectos desde cualquier lugar.

## Referencias

Chacon, Scott y Straub, Ben (sin fecha). *1.1 Getting Started - About Version Control*. Git-Scm.Com. Recuperado desde <https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>

GitHub. (9 de septiembre de 2020). En *Wikipedia*. Recuperado el 10 de septiembre de 2020 desde <https://en.wikipedia.org/w/index.php?title=GitHub&oldid=977524200>

What is GitHub? (sin fecha). En *GitHub*. Recuperado el 10 de septiembre de 2020 desde <https://guides.github.com/activities/hello-world/>