



# Fundamentos de Git y GitHub

**Abraham Zamudio Chauca**

<https://www.linkedin.com/in/abraham-zamudio/>

$$U^{n+1}$$



$$\frac{\partial v}{\partial t} + V \cdot \nabla v =$$

$$\nabla \cdot (k \nabla v) + g(v)$$

$$\rho \left( \frac{\partial u}{\partial t} + V \cdot \nabla u \right)$$

$$- \frac{\partial p}{\partial x} + \frac{\partial}{\partial t}$$

# Contenido

## 1 Parte I

- Introducción
- Características
- Instalación
- configuración

## 2 Parte II

## 3 Conclusiones



# Parte I

# "FINAL".doc



FINAL.doc!



FINAL\_rev.2.doc



FINAL\_rev.6.COMMENTS.doc



FINAL\_rev.8.comments5.  
CORRECTIONS.doc



FINAL\_rev.18.comments7.  
corrections9.MORE.30.doc



FINAL\_rev.22.comments49.  
corrections.10.#@\$%WHYDID  
ICOMETOGRADSCHOOL?????.doc

# Todo inicia con una historia :



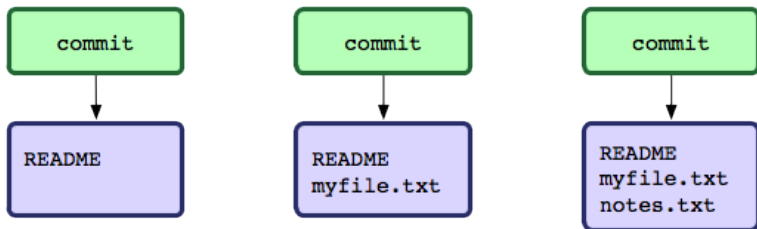
El Hombre Lobo y Drácula han sido contratados por Universal Missions para investigar si es posible enviar su próximo explorador planetario a Marte. Ellos quieren poder trabajar al mismo tiempo en los planes, pero ya han experimentado ciertos problemas anteriormente al hacer algo similar. Si se rotan por turnos entonces cada uno gastará mucho tiempo esperando a que el otro termine, pero si trabajan en sus propias copias e intercambian los cambios por email, las cosas se perderán, se sobrescribirán o se duplicarán.



# la historia continua ...

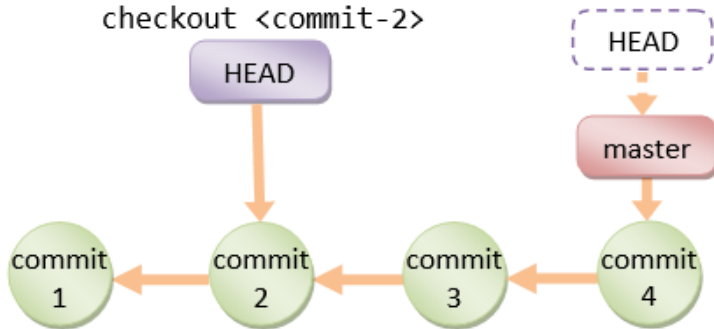
Un colega sugiere utilizar control de versiones para lidiar con el trabajo. El control de versiones es mejor que el intercambio de ficheros por email:

- Nada se pierde una vez que se incluye bajo control de versiones, a no ser que se haga un esfuerzo sustancial. Como se van guardando todas las versiones precedentes de los ficheros, siempre es posible volver atrás en el tiempo y ver exactamente quién escribió qué en un día en particular, o qué versión de un programa fue utilizada para generar un conjunto de resultados en particular.



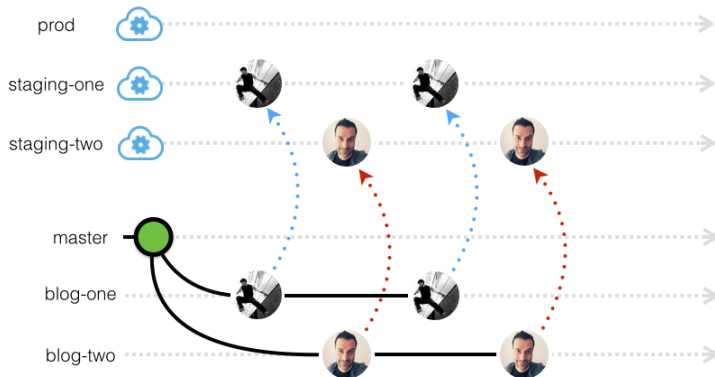
# la historia continua ...

- Como se tienen estos registros de quién hizo qué y en qué momento, es posible saber a quién preguntar si se tiene una pregunta en un momento posterior y, si es necesario, revertir el contenido a una versión anterior, de forma similar a como funciona el comando “deshacer” de los editores de texto.



# la historia continua ...

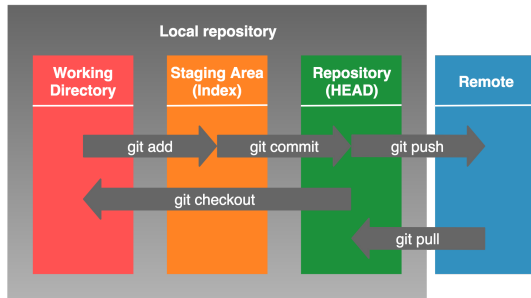
- Cuando varias personas colaboran en el mismo proyecto, es posible pasar por alto o sobrescribir de manera accidental los cambios hechos por otra persona. El sistema de control de versiones notifica automáticamente a los usuarios cada vez que hay un conflicto entre el trabajo de una persona y la otra.





# Versionamiento para todo :

Los equipos no son los únicos que se benefician del control de versiones: los investigadores independientes se pueden beneficiar en gran medida. Mantener un registro de qué ha cambiado, cuándo y por qué es extremadamente útil para todos los investigadores si alguna vez necesitan retomar el proyecto en un momento posterior (e.g. un año después, cuando se ha desvanecido el recuerdo de los detalles).



# Características : Acerca del Control de versiones

¿Qué es un control de versiones, y por qué debería importarte?

Un control de versiones es un sistema que registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante. Aunque en los ejemplos de esta charla usarás archivos de código fuente como aquellos cuya versión está siendo controlada, en realidad puedes hacer lo mismo con casi cualquier tipo de archivo que encuentres en una computadora.



# Características : Historia de Git

## Linux



Desde su nacimiento en el 2005, Git ha evolucionado y madurado para ser fácil de usar y conservar sus características iniciales. Es tremendamente rápido, muy eficiente con grandes proyectos y tiene un increíble sistema de ramificación (branching) para desarrollo no lineal

git

# Características de git

## Git

- Es software libre
- Es muy potente y rápida.
- Fue diseñada por Linus Torvalds
- No depende de un repositorio central
- Con ella podemos mantener un historial completo de versiones
- Podemos movernos, como si tuviéramos un puntero en el tiempo, por todas las revisiones de código y desplazarnos una manera muy ágil.
- Tiene un sistema de trabajo con ramas que lo hace especialmente potente
- En cuanto a la funcionalidad de las ramas, las mismas están destinadas a provocar proyectos divergentes de un proyecto principal.
- Las ramas pueden tener una línea de progreso diferente de la rama principal donde está el core de nuestro desarrollo.

git

# Instalación

## Windows

Debido a que el Command Prompt (o consola de Windows) no es muy amigable y no soporta comandos de Unix, la web de Git nos proporciona una herramienta muy buena llamada Git Bash y que nos permitirá escribir comandos como si estuviésemos en Linux o Mac OS X.

► Descarga

 **git 2.8.0 is here!**

# Crear una cuenta en

<https://github.com/>



# Contenido

## 1 Parte I

- Introducción
- Características
- Instalación
- configuración

## 2 Parte II

## 3 Conclusiones

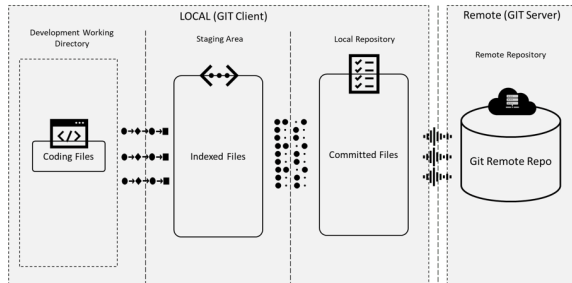
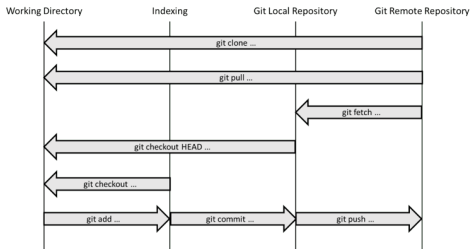


# Parte II





# GIT Basics and Beginners Guide



# Contenido

## 1 Parte I

- Introducción
- Características
- Instalación
- configuración

## 2 Parte II

## 3 Conclusiones



# Conclusiones

# Conclusiones

- Git tiene la clave para ser un mejor desarrollador, ya que te ayuda en algo tan primordial como a mantener y mejorar el orden en el desarrollo de software.
- Git y Github son herramientas fundamentales para el proceso de desarrollo de tu proyecto o producto.
- Para terminar, hoy en día el uso de software de control de versiones como Git es super esencial para el desarrollo de pequeños, medianos y grandes proyectos. Si queremos ser organizados, tener control de nuestro código en cualquier momento y lugar, contribuir con la comunidad programadora, Git se convierte en nuestra herramienta ideal.



Gracias  
por su  
participacion