



Carlos Yael De Los Santos Zorrilla. 2019-8150

Programacion 3 | Kelyn Tejada Belliard

Tarea 3

C1-2023

1-Que es Git?

Git es un *software* de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora incluyendo coordinar el trabajo que varias personas realizan sobre archivos compartidos en un repositorio de código.

Al principio, Git se pensó como un motor de bajo nivel sobre el cual otros pudieran escribir la interfaz de usuario o front end como Cogito o StGIT. Sin embargo, Git se ha convertido desde entonces en un sistema de control de versiones con funcionalidad plena. Hay algunos proyectos de mucha relevancia que ya usan Git, en particular, el grupo de programación del núcleo Linux.

El mantenimiento del software Git está actualmente (2009) supervisado por Junio Hamano, quien recibe contribuciones al código de alrededor de 280 programadores. En cuanto a derechos de autor Git es un software libre distribuible bajo los términos de la versión 2 de la Licencia Pública General de GNU.

2-Para que funciona el comando Git init?

El comando git init se utiliza para crear un nuevo repositorio vacío en la dirección donde le hayamos previamente indicado. Se creará una carpeta oculta llamada .git en la dirección indicada y en esta es donde se manejará el repositorio recién creado. Es decir, ahí es donde git guardará el proyecto y todas las versiones creadas.

4-Que es una rama? 3-Como saber es que rama estoy?

Podemos ver las ramas como espacios dentro de un repositorio, cuando creas un proyecto nuevo y este lo almacenas en un repositorio, se guarda en un espacio principal (main Branch). De este espacio o rama principal se pueden crear ramificaciones que parten del estado actual del proyecto, pero pueden ser modificados y guardado en otro espacio (secondary branch) sin que estos se reflejen en la rama principal.

En resumen, una rama es cada espacio que puede tener un repositorio el cual nos permite tener versiones distintas e independientes del software.

5-Quien creo git?

Git es una herramienta que inventó Linus Torvalds, el mismo que hizo el núcleo de Linux, en el 2005. Lo hizo porque quería una forma mejor de guardar y controlar los cambios en el código de Linux, que es un proyecto muy grande y complicado. Git es un programa que cualquiera puede usar y modificar porque es libre y de código abierto, y ha mejorado mucho con el tiempo gracias a la ayuda de mucha gente.

6-Cuales son los comandos más esenciales de Git?

start a working area (see also: git help tutorial)

clone Clone a repository into a new directory

init Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)

add Add file contents to the index

mv Move or rename a file, a directory, or a symlink

restore Restore working tree files

rm Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)

bisect Use binary search to find the commit that introduced a bug

diff Show changes between commits, commit and working tree, etc

grep Print lines matching a pattern

log Show commit logs

show Show various types of objects

status Show the working tree status

grow, mark and tweak your common history

branch List, create, or delete branches

commit Record changes to the repository

merge Join two or more development histories together

rebase Reapply commits on top of another base tip

reset Reset current HEAD to the specified state

switch Switch branches

tag Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)

fetch Download objects and refs from another repository

pull Fetch from and integrate with another repository or a local branch

Update remote refs along with associated objects

7-Que es git Flow?

Git flow es un modelo de trabajo con Git que define cómo se deben crear y gestionar las ramas de un repositorio. Fue propuesto por Vincent Driessen en el año 2010. Git flow utiliza dos ramas principales: master y develop, y varias ramas auxiliares: feature, release y hotfix. Cada una de estas ramas tiene un propósito específico y unas reglas de fusión. Git flow es útil para proyectos que tienen un ciclo de publicación programado y que requieren una organización clara del código.

Master: es la rama principal que contiene el código que está listo para ser publicado o desplegado. Solo recibe código de las ramas release y hotfix.

Develop: es la rama principal que contiene el código que está en desarrollo y que se espera que funcione. Solo recibe código de las ramas feature.

Feature: son ramas auxiliares que se crean para desarrollar nuevas funcionalidades o requisitos. Se crean a partir de la rama develop y se fusionan con ella cuando la funcionalidad está terminada y probada.

Release: son ramas auxiliares que se crean para preparar una nueva versión del código. Se crean a partir de la rama develop y se fusionan con la rama master y con la rama develop cuando la versión está lista y depurada.

Hotfix: son ramas auxiliares que se crean para corregir errores críticos que se detectan en la rama master. Se crean a partir de la rama master y se fusionan con la rama master y con la rama develop cuando el error está solucionado.

8-Que es trunk-based development?

En otras palabras, el desarrollo basado en troncos es una forma de trabajar con Git donde los desarrolladores hacen cambios pequeños y los unen a una rama principal muchas veces al día . Es una forma de hacer DevOps, que es una manera de desarrollar software más rápido y mejor. El desarrollo basado en troncos hace que sea más fácil integrar el código y evitar conflictos entre las ramas . Los desarrolladores usan ramas cortas que duran poco tiempo, no como otras formas de usar ramas largas que duran mucho .