



Área Académica de Ingeniería Mecatrónica Carrera de Ingeniería
Mecatrónica

MT7003 - Microprocesadores y Microcontroladores

Profesor: Ing Rodolfo José Piedra Camacho, Lic.

Tarea 1: GitHub, Pytest y Flake 8

Joshua Corvera Almendarez

Omar Gabriel Fuentes Aguilar

13/08/2021

II Semestre

Preguntas Teóricas (16 pts, 2pts c/u)

1) ¿Diferencie la herramienta Git de Github?

La herramienta Git es un sistema de control de versiones distribuido (DVCS), se usa para guardar diferentes versiones de un archivo para que cualquier versión sea recuperable cuando se desee. Git también permite comparar las distintas versiones, o sea, es capaz de indicar los detalles sobre los cambios realizados y quien los modificó en cualquier momento. Otra cosa asombrosa es que Git permite que las personas trabajen juntas en un mismo proyecto en simultáneo sin perturbar los archivos de los demás. [1].

GitHub es un servicio basado en la web para el control de versiones usando Git. Principalmente Github permite por medio de una plataforma mantener repositorios de código en un almacenamiento en la nube, para que varios usuarios puedan tener acceso y trabajar en conjunto en un único proyecto, visualizando los cambios en tiempo real. Ver [2].

Las diferencias entre ambas radican en que Git es una herramienta de control de versiones que detalla en su historial los cambios realizados al código, mientras que Github es un sitio web basado en el almacenamiento en la nube que funciona en ayuda de Git. [3].

2) ¿Qué es un branch?

Los branch o ramas corresponden a una bifurcación del estado del código que crea un nuevo camino para la evolución del mismo. Un branch puede desarrollarse en paralelo junto a otras Git Branch, ver [4]. Se utilizan a menudo cuando se requiere resolver problemas o crear nuevas funcionalidades.

3) ¿Qué es un commit?

Commit es un comando que define el final de una transacción ejecutada con éxito, este comando permite asegurar que todas las modificaciones realizadas durante la transacción se conviertan en parte permanente de la base, significa que guarda en un punto y tiempo dado el contenido y estado del repositorio [5].

4) ¿Qué es la operación cherry-pick?

La operación cherry-pick permite escoger commits de otra rama sin tener que hacer un merge total [6]. Es decir, permite aplicar cambios realizados en otros commits al branch en la que se está trabajando, cabe destacar que el branch no debe tener cambios, o sea “clean status”.

- 5) Explique de forma gráfica cómo cambia el “master” de un repositorio cuando se hace merge de un Branch.

Git permite el uso de distintos branches o ramas para evitar afectar el código principal, ya sea para generar modificaciones o para corregir errores sin afectar el trabajo completo, estas ramas se pueden fusionar por medio de un merge, durante el proceso del merge se pueden generar 2 casos un fast-forward merge (merge de avance rápido) o un 3-way merge (merge a tres bandas) [7].

En el fast-forward se hace un merge con la rama “master” y no se ha añadido ningún commit luego de crear el branch secundaria (feature), como se muestra en la Figura 1 a la izquierda el antepasado común es “head master”, como no se realizó ningún commit del branch feature, el master se actualiza en el feature sin que el merge cree un commit, por eso su nombre de avance rápido [7].

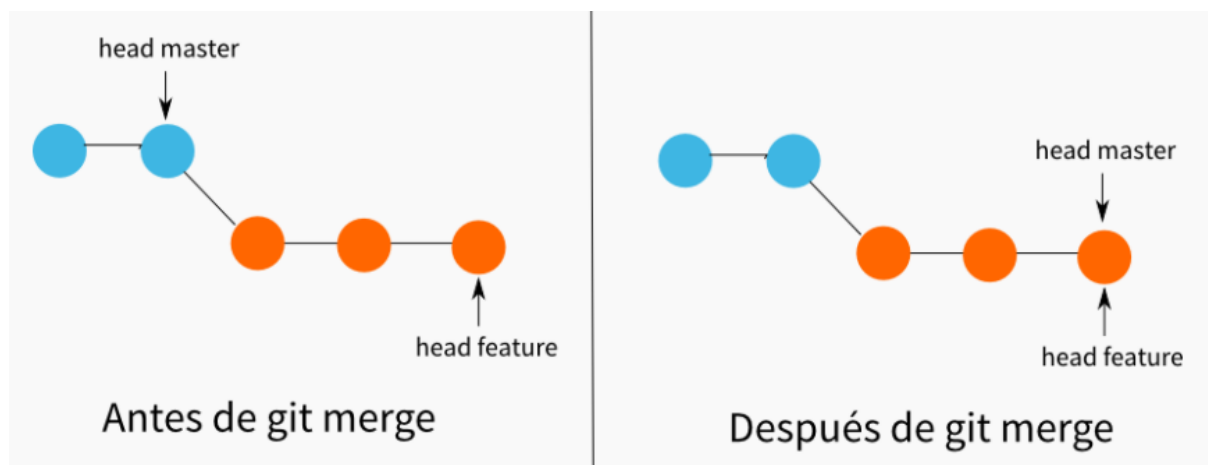


Figura 1. Diagrama de un fast-forward merge.

Por otro lado, si la rama master ha divergido después de haber creado la rama feature ya no es posible un fast-forward merge, debido a que el commit de la rama donde actualmente se está (master) no es un antepasado directo de la rama a fusionar (feature) por tanto, git realiza un merge a tres bandas, es decir, que genera un commit para fusionar las dos ramas, tomando en cuenta el HEAD de cada una de ellas y el antepasado común de las dos

En caso de querer hacer un commit al realizar el merge y si la rama del master diverge, ya no se puede realizar un fast-forward merge, ya que master no es el antepasado común, sino que disponen de otro como se muestra a la izquierda de la Figura 2, en este caso se debe de hacer un merge a tres bandas o 3-way merge. En este merge se crea un nuevo commit que va a

fusionar las dos ramas (master y feature), usando su antepasado común y las cabezas de ambas ramas, de esta manera como la imagen de la derecha de la Figura 2 [7].

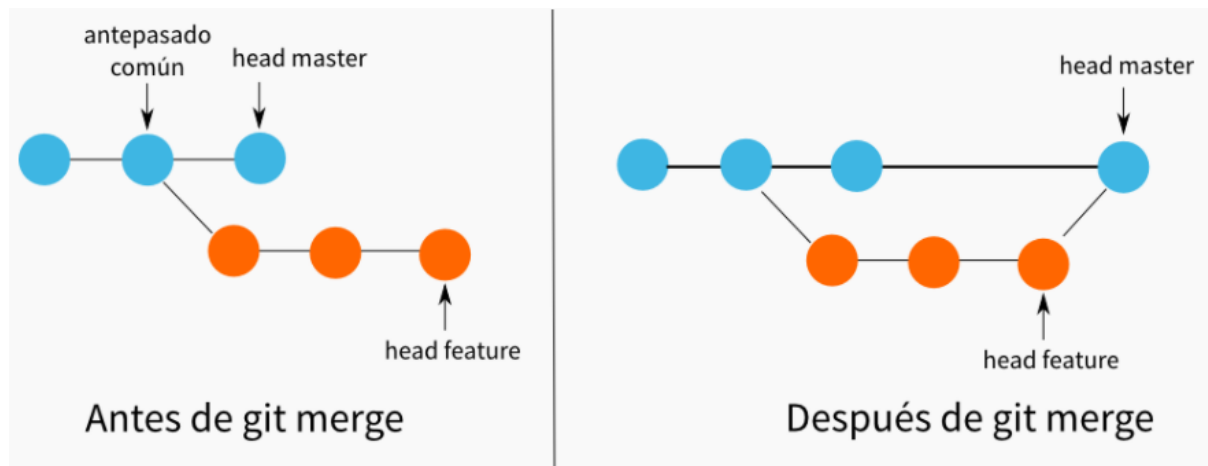


Figura 2. Diagrama de un 3-way merge

6) ¿Qué es una Prueba Unitaria o Unittest en el contexto de desarrollo de software?

Un Unittest, en español prueba unitaria, se puede utilizar para comprobar los componentes individuales de los códigos en lenguajes de programación. La ventaja que presentan los Unittest es que pueden verificar fácilmente y de manera rápida el funcionamiento de cada elemento antes de incorporarlos de forma general al programa. [8]

7) Bajo el contexto de pytest. ¿Qué es un “assert”?

En pytest un “assert” permite realizar comprobaciones, de modo que si la expresión contenida dentro del mismo es falsa, se va a lanzar un mensaje de “AssertionError”. Su función puede ser reemplazada con el uso de “if”, no obstante usar “assert” es más practicado de usar, además de poder modificar el mensaje entregado por alguno más específico [9].

8) ¿Qué es Flake 8?

Flake8 se trata de una herramienta de Python para detectar si el código de Python está estandarizado, ver [10].

Flake8 se caracteriza en que posee reglas de inspección flexibles, admite la incorporación de complementos adicionales y tiene amplia escalabilidad.

Flake8 es un paquete con 3 herramientas, las cuales revisan estáticamente la convención estilística de codificación PEP8, además de que verifican estáticamente errores de lógica,

analizan la complejidad del código y revisa si hay variables declaradas no utilizadas, así como imports y otras. Mediante esta herramienta el código queda más limpio.

Referencias

- [1] E. Castellanos, (2021, Feb. 14). “Git vs GitHub – ¿Qué es el Control de Versiones y Cómo Funciona?”. [Online]. Disponible en: <https://www.freecodecamp.org/espanol/news/git-vs-github-what-is-version-control-and-how-does-it-work/>
- [2] Kinsta (2020). [Online]. Disponible en: <https://kinsta.com/es/base-de-conocimiento/git-vs-github/>
- [3] aprendeIA (2020). [Online]. Disponible en: <https://aprendeia.com/que-es-git-y-github/#comments>
- [4] D. Albornoz. (2021, Jul 29). Cómo usar una Git Branch. [Online]. Disponible en: <https://www.hostinger.es/tutoriales/como-usar-git-branch>.
- [5] TechTarget Contributor (2005). [Online]. Available: <https://searchsqlserver.techtarget.com/definition/commit>
- [6] Seleccionando commits: Cherry-pick. (2017, junio 15). [Online]. Disponible en: <https://www.runroom.com/realworld/seleccionando-commits-cherry-pick>.
- [7] C. Rondón. (2015, Oct 3). Ramas y resolución de conflictos en git. [Online]. Disponible en: <https://styde.net/ramas-y-resolucion-de-conflictos-en-git/>
- [8] Unit testing: la prueba de calidad para software. (2019, marzo 14). [Online]. Disponible en: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-papel-del-unit-test-en-el-desarrollo-de-software/>.
- [9] pytest (s.f). [Online]. Disponible en: <https://docs.pytest.org/en/6.2.x/assert.html>
- [10] Programador clic. (s.f.). [Online]. Disponible en: <https://programmerclick.com/article/40861407461/>.