



Microprocesadores y Microcontroladores

Ing. Rodolfo Piedra Camacho

Kenneth G. Gutiérrez Bravo

Christopher J. Morales Acosta

Tarea 1 GitHub, Pytest y Flake 8

Entrega: 5 de marzo

II Semestre

2020

1. ¿Diferencie la herramienta Git de Github?

Git es una herramienta de control de versiones distribuida que puede gestionar el historial de código fuente de un proyecto de desarrollo, mientras que GitHub es una plataforma basada en la nube construida alrededor de la herramienta Git. La diferencia principal entre Git y GitHub es que Git es una herramienta de código abierto que los desarrolladores instalan localmente para gestionar el código fuente, mientras que GitHub es un servicio en línea al que los desarrolladores que utilizan Git pueden conectarse y cargar o descargar recursos y realizar trabajo colaborativo.

2. ¿Qué es Branch?

En Git, las Branch son espacios o entornos independientes para que un desarrollador pueda usar y así trabajar sobre un mismo proyecto sin chancar o borrar el conjunto de archivos originales del proyecto, dándonos flexibilidad para desarrollar nuestro proyecto de manera más organizada y ramificada.

3. ¿Qué es un commit?

Realizar una operación de commit significa que se está realizando una “confirmación de cambios”, de esta forma queda una versión actualizada. Los commit se realizan cuando se tienen documentos con contenido preparado (staged content) y permite agregar información sobre el cambio realizado (como los principales cambios realizados, nombre y correo de aquel que realiza el commit, etc.)

4. ¿Qué es la operación cherry-pick?

git cherry-pick es un potente comando que permite que los commit arbitrarios de algún branch se elijan por referencia y se añadan al actual HEAD de trabajo. La ejecución de cherry-pick es el acto de elegir cierto commit de cierto branch y aplicarlo a otro Branch (incluyendo el principal). git cherry-pick puede ser útil para deshacer cambios. Por ejemplo, supongamos que una confirmación se aplica accidentalmente en la rama equivocada. Puedes cambiar a la rama correcta y ejecutar cherry-pick en la confirmación para aplicarla a donde pertenece. Hay que tener cuidado pues es posible que se generen duplicate commits, que pueden afectar la integridad del repositorio.

5. ¿Qué hace el comando git stash?

El comando git stash almacena temporalmente (o guarda en un stash) los cambios que hayas efectuado en el código en el que estás trabajando para que puedas trabajar en otra cosa y, más tarde, regresar a aplicar los cambios más tarde. Guardar los cambios en stashes resulta práctico si tienes que cambiar rápidamente de contexto y enfocarse con otra cosa, pero estás en medio de un cambio en el código y no lo tienes todo listo para realizar un commit.

6. Compare las operaciones git fetch y git pull.

El comando Fetch descarga commits de un repositorio remoto al repositorio personal local de forma aislada, este comando es útil para hacer revisiones del trabajo colaborativo de otros. Cuando se requiere confirmar la incorporación del contenido descargado del Fetch se utiliza la función de Checkout. De forma similar el comando Pull descarga algún commit de un repositorio remoto, pero a diferencia de Fetch este realiza un commit directamente al

repositorio personal realizando un merge. Pull se suele utilizar cuando se esta completamente seguro de que se quiere incorporar el commit seleccionado en el repositorio local, sin embargo, hay que tener cuidado de no tener progreso sin guardar pues se pueden generar conflictos a la hora de hacer el merge.

7. Asumiendo que usted está en un Branch llamada “secundario” y su Branch principal se llama “master” ¿Qué resultado espera de hacer git rebase master? ¿Qué resultado espera de hacer git rebase origin/master?

El comando rebase es una forma de incorporar un cambio o “patch” sobre la rama principal. Se diferencia de la función Merge, de la siguiente forma: con merge se añaden los cambios en una Branch anteriormente creada y luego se une esta con la rama principal, sin embargo, el comando Rebase se capturan todos los commit de una rama y se aplican directamente sobre la principal. De esta forma es más clara y directa la incorporación de los patches en el repositorio.

Ahora cuando se realizan clonaciones de un repositorio remoto, git automáticamente nombra la rama remota como origin/master que es una copia local de la rama master del repositorio remoto origin. Por lo tanto, en el repositorio local se generarán dos ramas al realizar una clonación. La rama origin/master que es la copia local del repositorio remoto y una rama master que es la rama principal del repositorio local.

Entonces cuando se realiza un rebase de la rama master, la operación se realiza sobre la rama principal del repositorio local, a diferencia del caso donde se realiza un rebase sobre la rama origin/master que se realiza sobre la copia del repositorio remoto.

8. ¿Qué es una Prueba Unitaria o Unittest en el contexto de desarrollo de software?

Se trata de un método para determinar si un módulo o un conjunto de módulos de código funciona correctamente. El concepto de Unit testing no se limita a ningún lenguaje específico, sino que es una herramienta de la programación en general. Las pruebas unitarias se implementan con el desarrollo de un módulo o proyecto, y se ejecutan cuando este último sufre modificaciones para garantizar su funcionamiento. Si bien el código mismo de la prueba unitaria puede contener errores, la clave está en la separación del código de un módulo de su respectiva prueba unitaria, de modo que puedan correr independientemente.

En otras palabras, es una forma de comprobar que un conjunto de funciones o clases (tantas como queramos) funcionan como esperamos. Lógicamente, las pruebas unitarias nunca pueden garantizar completamente el correcto funcionamiento de una porción de código. No obstante, ello, serán capaces de detectar gran cantidad de anomalías y de ahorrarnos tiempo de depuración.

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

Un assert son pequeñas pruebas de errores que retornan un valor lógico True or False, son partes de un método más grande de pruebas de errores. Cuando un assert falla en la secuencia de pruebas el test se detiene y no se ejecuta el resto del código que se prueba.

10. ¿Qué es Flake 8?

Flake8 es un linter de código de Python. Esto significa que Flake8 es una herramienta de revisión y corrección de código que permite revisar la sintaxis y estructura básica del código de Python. Este tipo de herramientas son útiles para mantener un código limpio, documentado y organizado. Además, permite ahorrar tiempo de depuración, identificar advertencias y errores de programación. Entre sus ventajas están la facilidad de uso con el usuario, fácil configuración e instalación.