

Tarea 1. Preguntas teóricas

¿Diferencie la herramienta Git de Github?

Según [2], [8] y [10], Git es un sistema de control de versiones, o control de código fuente; es un software de código abierto que permite gestionar y rastrear cambios o modificaciones en el código fuente de uno o varios archivos localizados en un repositorio.

¿Qué es un branch?

Una rama o *branch* es una bifurcación de los archivos trabajados creada, lo que crea dos copias idénticas de estos para ser trabajados independientemente uno del otro, a partir del momento que se crea. [2][8]

¿Como se crea un nuevo Branch?

Para crear un nuevo branch de nombre *ramaAlt* a partir de la rama que se esté trabajando, se utiliza el comando “git branch ramaAlt” [2]

¿Qué es un commit?

Un *commit* es la acción que guarda los cambios realizados en el programa principal o en el *branch* que se esté trabajando. Es un punto de control al cual se puede regresar si cambios futuros realizados a los archivos no se desean por alguna razón. [2][6]

¿Qué es la operación “git stash”?

La operación *git stash* es un tipo de guardado temporal, que almacena los cambios hechos al código, mas no confirma esos cambios (no son cambios “oficiales”); realiza un guardado en una reserva o “stash”. [2]

Explique de forma gráfica: ¿Que sucede en mi Branch local cuando hago “git rebase master”?

Antes del rebase, se tienen los commits A, B, C, D, E en la rama principal (master, de color azul), u los commits F, G en una rama alternativa (alt, color rojo), la cual es la rama activa (se está trabajando en ella). Al ejecutar *git rebase master* estando en la rama *alt*, ocurre lo siguiente (de acuerdo con [2] y [4]):

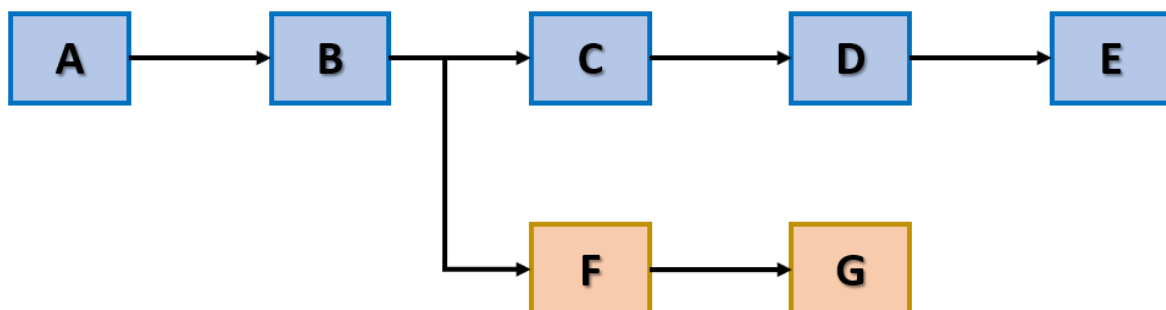


Figura 1. Línea temporal de “commits” antes de realizar un “rebase”. Fuente: propia

Todos los commits de *alt* (E, F) son reconstruidos después del último commit realizado en *master* (E), lo que genera nuevos commits E' y F', los cuales pertenecen a *master*, a diferencia de E y F, que pertenecen a *alt*; solo se diferencian en la rama a la cual pertenecen.

Después del rebase, los commits E y F aún están indicados, mas ya no son accesibles. La rama *master* tiene nuevos commits idénticos a los que estaban en *alt*.

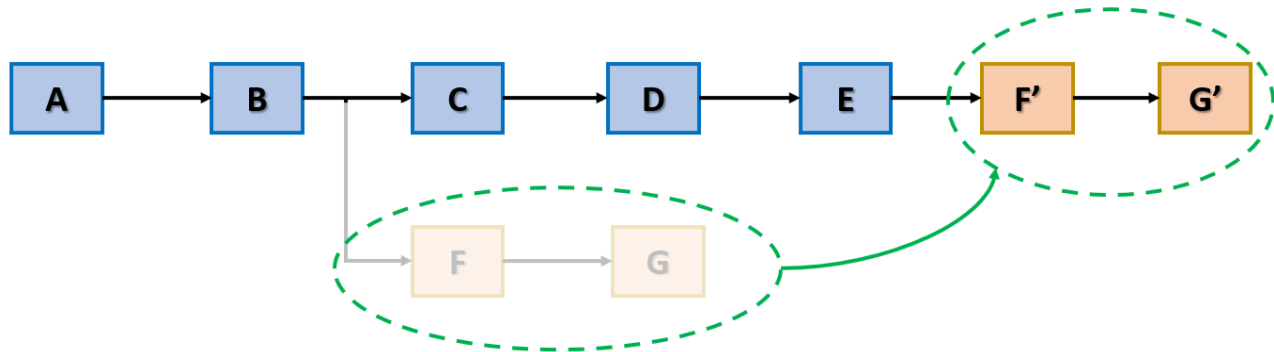


Figura 2. Línea temporal de “commits” después de realizar un “rebase”. Fuente: propia

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

Un Unittest es una forma de probar software en las que pequeñas porciones del código fuente de un programa son evaluadas, con el fin de determinar si la calidad del código es apta para su uso en aplicaciones específicas. [2][9]

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

Un *assert* es un comando de Python utilizado para verificar datos esperados, mediante test del código a analizar, con la función retornando un error si la comparación de datos no es la establecida o la deseada. [5]

¿Qué es Flake 8?

Flake8 es una librería dentro del Python Package Index (PyPI) que funciona como herramienta de “lint”, dicho de otra forma, ejecuta una herramienta de “control de calidad” que revisa el código y su sintaxis respecto a una serie de estándares (PEP8) y provee una guía para arreglar errores existentes. Además, Flake8 permite al usuario modificar los estándares sobre los que se revisa el código de forma sencilla mediante plugins. [3][11]

Explique la diferencia entre un “log de error” y un “valor de error de retorno”

Un log de error es un sistema que genera una gran cantidad de datos en forma de trazas textuales que usualmente no están visibles para el usuario. Técnicamente describen el comportamiento de los sistemas o programas. [1] Un valor de error de retorno es lo que la función o código retorna al usuario, ya sea un código específico de error definido por el programador o incluso un valor que indique el correcto funcionamiento del código. [7]

Referencias

- [1] A. Diaz. “¿Qué son los Logs y por qué deben interesarte?”. 19 de Enero, 2017. [En línea]. Disponible en: <https://dbibyhavas.io/es/blog/que-son-los-logs/> [Accesado el 15 de febrero de 2022]
- [2] Chacón, S & Straub, B. *ProGit*, 2da ed. Nueva York: Apress, 2014.
- [3] Dev Null, “What is Flake8 and why we should use it?”. 30 Enero, 2017. [En línea]. Disponible en: <https://medium.com/python-pandemonium/what-is-flake8-and-why-we-should-use-it-b89bd78073f2> [Accesado el 15 de febrero de 2022]
- [4] “Git rebase”. [En línea]. *Disponible* en: <https://www.atlassian.com/es/git/tutorials/rewriting-history/git-rebase> [Accesado el 15 de febrero de 2022]
- [5] H. Krekel. “pytest Documentation”. 25 de noviembre de 2020. [En línea] Disponible en: https://docs.pytest.org/_downloads/en/4.6.x/pdf/ [Accesado el 15 de febrero de 2022]
- [6] “Lineamientos para el uso de repositorios de código fuente”. [En línea]. México D.F. Dirección General de Cómputo y Tecnologías de Información y Comunicación. 2016. Disponible en: <https://www.red-tic.unam.mx/recursos/LineamientosRepositorioCodigoFuente.pdf> [Accesado el 16 de febrero de 2022].
- [7] Oracle. “Berkeley DB Programmer’s Reference Guide”. 16 de Junio, 2015. [En línea]. Disponible en: https://docs.oracle.com/cd/E17276_01/html/programmer_reference/program_errorret.html [Accesado el 15 de febrero de 2022]
- [8] S. Gómez. “Introducción a Git y Github-Día 1”. 17 de febrero de 2022. Córdoba, España: Universidad de Córdoba, 2015. Disponible en: <https://www.uco.es/aulasoftwarelibre/wp-content/uploads/2015/11/git-cosfera-dia-1.pdf> [Accesado el 17 de febrero de 2022]
- [9] S. Lukauskas. “Unit testing or what can we learn from tech startup.” 1 de abril de 2014. [En línea]. Disponible en: <http://www.theosysbio.bio.ic.ac.uk/wp-content/uploads/2013/10/unit-testing-presenter-notes.pdf>
- [10] “¿Qué es el control de versiones?”. [En línea]. Disponible en: <https://www.atlassian.com/es/git/tutorials/what-is-version-control> [Accesado el 15 de febrero de 2022]
- [11] V. Freitas, “How to Use Flake8,” 5 de Agosto, 2016. [En línea]. Disponible en: <https://simpleisbetterthancomplex.com/packages/2016/08/05/flake8.html#:~:text=Flake8%20is%20a%20Python%20library,and%20to%20check%20cyclomatic%20complexity.> [Accesado el 15 de febrero de 2022]