



**Instituto Tecnológico de Costa Rica**  
**Área Académica de Ingeniería Mecatrónica**

**MT-7003 Microprocesadores y Microcontroladores**  
**Prof. Rodolfo Piedra**

**Tarea 1**  
**GitHub, Pytest y Flake 8**

**Estudiantes:**

<b>Juan Pablo Jiménez Anderson</b>	<b>2018099431</b>
<b>José Ignacio Pauly Llobet</b>	<b>2018092536</b>

**I Semestre – 2021**

### 1) ¿Diferencie la herramienta Git de Github?

Git es un sistema de control de versión, estos permiten mantener un registro de todos los cambios realizados a un programa, y volver a cualquiera de ellos en el caso que se encuentre un problema en el código causado por alguna edición errónea.

Además, Git es completamente local al sistema, es decir, es totalmente offline y tiene el feature de “branches,” o ramas.

Por otro lado, Github es un repositorio online para programas de github, donde los usuarios pueden compartir y discutir sobre sus proyectos. [1]

### 2) ¿Qué es un branch?

Una rama, o “**Branch**” es una línea de ediciones que se realiza en paralelo al programa principal o maestro, estas permiten organizar de manera sencilla cosas como debug, y updates a una parte específica de la programación, sin comprometer la estabilidad del código ya probado, Además de permitir observar en qué punto se realizaron los cambios. [1]

### 3) ¿Qué es un commit?

Si los Branches son comparables a las ramas de un árbol, entonces los **commits** son el tronco, estos representan los cambios cronológicos del programa principal, son como los “Save points” o “checkpoints” del mismo. [1]

### 4) ¿Qué es la operación cherry-pick?

El **Cherry-Pick** permite escoger un commit y moverlo a otra rama, o adjuntarlo al HEAD. Es útil para mover un commit que se colocó en un lugar incorrecto, y moverlo a donde debe estar. En general, no se considera la mejor práctica, pero es útil para cuando se está trabajando en equipo y para realizar hotfixes. [1]

### 5) ¿Qué hace el comando git stash?

El **Stash** permite guardar de manera temporal los cambios que se están realizando en una rama, sin realizar un commit, para poder realizar cambios a otras ramas, o para corregir la selección actual del punto donde se quiere guardar este cambio, en caso de que no sea correcto.

Este proceso es equivalente a poner la página de un documento físico a un lado, y buscar entre las otras páginas donde se quiere colocar, antes de grapar las hojas. [1]

### 6) ¿Compare las operaciones git fetch y git pull

Ambos comandos, pull y fetch son utilizados para descargar nuevos datos de un repositorio remoto. El comando git fetch, por su parte únicamente descarga nueva información de repositorios remotos, pero sin integrarla, por lo cual es muy útil al para dar un vistazo a repositorios remotos y sus nuevas mejoras, sin manipular, destruir o dañar el repositorio local. Por su parte, el comando git pull realiza prácticamente la misma función, sin embargo, este comando se usa para realizar updates al head branch con los últimos cambios que se han realizado en el repositorio remoto. Esta integración puede provocar un conflicto de integración o “merge conflict”. [2]

Es posible decir que el comando git pull es una combinación entre el comando git fetch y el comando git merge, ya que descarga la información de los repositorios remotos y lo integra. [3]

**7) Asumiendo que usted está en un Branch llamado “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 resultado esperado del comando \$git rebase master si se está en el Branch llamado secundario es que este ahora contiene todos los cambios del él mismo así como los cambios del Branch denominado master, es una acción similar a la que hace el comando merge, sin embargo, el “commit tree” es un poco distinto, ya que este se rescribe por lo que el Branch master es parte del commit history ahora. [4]

El comando \$git rebase origin/master lo que quiere decir que realiza un rebase del Branch actual sobre el master [5]

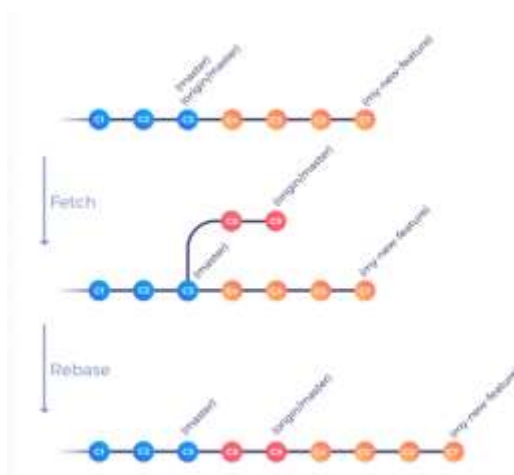


Figura 1. Diagrama del comando “\$git rebase origin/master” [6]

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

La prueba unitaria es un método de prueba de software mediante el cual las unidades individuales de código fuente se someten a varias pruebas para determinar si son aptas para su uso (fuente). Determina y comprueba la calidad de su código.

Generalmente, cuando se completa el proceso de desarrollo, el desarrollador codifica los criterios, o los resultados que se sabe que son potencialmente prácticos y útiles, en el script de prueba para verificar la exactitud de una unidad en particular. Durante la ejecución del caso de prueba, varios marcos registran las pruebas que fallan en cualquier criterio y las informan en un resumen.

Se espera que los desarrolladores escriban scripts de prueba automatizados, lo que asegura que todas y cada una de las secciones o unidades cumplen con su diseño y se comportan como se espera.

Aunque escribir pruebas manuales para su código es definitivamente una tarea tediosa y que requiere mucho tiempo, el marco de pruebas unitarias incorporado de Python ha hecho la vida mucho más fácil.

El marco de pruebas unitarias en Python se llama unittest, que viene empaquetado con Python. [7]

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

Pytest le permite usar la afirmación estándar de Python para verificar las expectativas y los valores en las pruebas de Python. Bajo el contexto de pytest los assert statements son el núcleo de lo que es la prueba, para los unittest, cada función o método de prueba debe tener solo un assert statement. La idea general es configurar todos los datos requeridos, llamar a la función o método del código fuente y confirmar mediante assert statements que se obtienen los resultados conforme a lo esperado. [8]

### **10) ¿Qué es Flake 8?**

Flake8 es una biblioteca de Python que envuelve PyFlakes, pycodestyle y el script McCabe de Ned Batchelder. Es un gran juego de herramientas para comparar su base de código con el estilo de codificación (PEP8), errores de programación (como “biblioteca importada pero no utilizada” y “Nombre no definido”) y para verificar la complejidad ciclomática. Flake 8 ejecuta todas las herramientas utilizando un único comando y muestra las advertencias en una salida combinada por archivo.

Como parte de las características que presentan este comando los archivos que contienen la línea

```
#flake8: noqa
```

Y las líneas que tienen el comentario #noqa al final no emitirán advertencias. [9]

## Referencias

- [1] Git Immersion, «Git Immersion,» Git Immersion, 2020. [En línea]. Available: <https://gitimmersion.com>.
- [2] Git tower, «Git FAQ,» Git tower, 2021. [En línea]. Available: <https://www.git-tower.com/learn/git/faq/difference-between-git-fetch-git-pull/>. [Último acceso: 24 Febrero 2021].
- [3] S. Hossain, «Git Fetch vs Git Pull,» Sabbir Hossain, 18 Diciembre 2019. [En línea]. Available: [https://medium.com/@sabbirhossain\\_70520/git-fetch-vs-git-pull-691823ed4239](https://medium.com/@sabbirhossain_70520/git-fetch-vs-git-pull-691823ed4239). [Último acceso: 24 Febrero 2021].
- [4] Git Immersion, «Lab 34. Rebasing,» [En línea]. Available: [https://gitimmersion.com/lab\\_01.html](https://gitimmersion.com/lab_01.html). [Último acceso: 27 Febrero 2021].
- [5] V. Vachhar, «Git Rebase,» 1 Marzo 2017. [En línea]. Available: <https://varun.ca/rebase/>. [Último acceso: 28 Febrero 2021].
- [6] A. Seure, «Master the Rebase (and the Other Way Around),» algalia, 6 Diciembre 2017. [En línea]. Available: <https://www.algalia.com/blog/engineering/master-git-rebase/>. [Último acceso: 28 Febrero 2021].
- [7] A. Sharma, «Unit Testing in Python,» Datacamp, 1 Mayo 2020. [En línea]. Available: <https://www.datacamp.com/community/tutorials/unit-testing-python>. [Último acceso: 24 Febrero 2021].
- [8] H. Krekel, «Assert statements,» pytest, 2015. [En línea]. Available: [https://docs.pytest.org/en/reorganize-docs/new-docs/user/assert\\_statements.html#:~:text=Assert%20statements%20are%20at%20the%20core%20of%20what%20a%20test%20is.&text=The%20plain%20Python%20assert%20statement,output%20to%20give%20more%20context](https://docs.pytest.org/en/reorganize-docs/new-docs/user/assert_statements.html#:~:text=Assert%20statements%20are%20at%20the%20core%20of%20what%20a%20test%20is.&text=The%20plain%20Python%20assert%20statement,output%20to%20give%20more%20context). [Último acceso: 24 Febrero 2020].
- [9] F. Vitor, «How to Use Flake8,» Simple is better than complex, 5 Agosto 2016. [En línea]. Available: <https://simpleisbetterthancomplex.com/packages/2016/08/05/flake8.html>. [Último acceso: 25 Febrero 2021].