Instituto Tecnológico de Costa Rica

Área Académica de Ingeniería Mecatrónica

Carrera de Ingeniería Mecatrónica

Microprocesadores y Microcontroladores

Tarea #1: GitHub, Pytest y Flake 8

Profesor:

Rodolfo Piedra Camacho

Estudiantes:

Alejandra Sousa Leal

David Wu Cen

II Semestre 2021

Preguntas teoricas

1) ¿Diferencie la herramienta Git de Github?

Git es un sistema de control que se instala en el sistema local donde se utilizara y contiene un historial de las versiones de nuestros códigos. Se puede utilizar incluso sin servicio a internet a excepción de su descarga [1].

GitHub por otro lado es un servicio en la nube, que se trabaja de manera en-linea. Es una base de datos que permite mantener un historial y compartir la versión Git de nuestros proyectos incluso fuera de nuestros sistemas locales o servidores. GitHub opera totalmente online [1].

La principal diferencia entonces sería que Git nos permite tener un historial de nuestro código fuente, mientras que GitHub nos permite tener un repositorio de Git, además que Git es local mientras que GitHub trabaja en la nube. GitHub nos da la facilidad de poder compartir, ver las ediciones y comentarios de otros con los que hayamos compartido nuestro código [1].

2) ¿Qué es un branch?

Una branch es una línea de desarrollo independiente. Es una forma de aislar trabajo, esto nos permite desarrollar funciones, arreglar problemas y experimentar con ideas en un entorno aislado. Tenemos nuestra default branch, que es nuestra línea principal de desarrollo, y de esta branch por defecto, creamos más branches [2].

3) ¿Qué es un commit?

Commit es un comando que salva los cambios en el repositorio local de Git. Es una forma de tener una snapshot, una prueba no definitiva del producto final. Se recomienda hacer commits bastante seguido. Los commits deberían marcar la historia de nuestro repositorio y como llegó a ser como es [3].

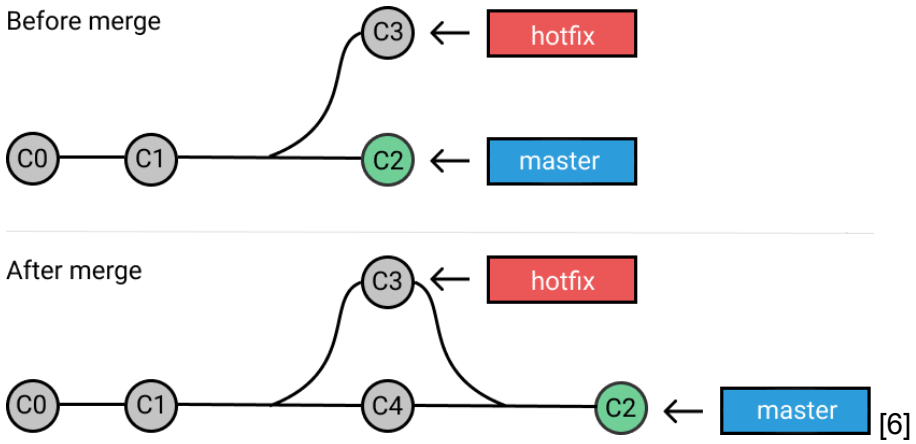
4) ¿Qué es la operación cherry-pick?

Cherry-pick es comando de Git que permite tomar uno o varios commits de otra rama sin tener que hacer un merge completo. Para usar cherry-pick solo hay que conocer el commit específico que se quiere aplicar en la rama. Para aplicarla se usa con este comando: "(master) $ git cherry-pick" seguido del commit, por ejemplo "(master) $ git cherry-pick 13f03ab". Se debe tomar en cuenta que este comando crea un nuevo commit, por lo que antes de usar cherry-pick no debemos de tener ningún archivo modificado que no haya sido incluido en un commit. Cherry-pick puede ser utilizado con una variedad de opciones para mejorar el flujo, por ejemplo para usar más de un commit, se pueden anotar subsecuentemente y añadir opciones como:

- -x : modifica el mensaje para añadir una referencia al commit original y así poder saber de donde se ha traído este commit.
- -e : permite editar el mensaje del commit en lugar de usar el original.
- -n: aplicará cambios pero no hará ningún commit [4].

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

El término master se refiere a la versión principal o primaria de código fuente de un repositorio. El merge es un comando que permite tomar líneas o branches de desarrollo creados independientemente e integrarlos en otra branch [5]. Por ejemplo, en el master o branch principal.



[6]

La imagen anterior muestra el primer ejemplo de un merge, por ejemplo, acá tenemos un branch donde se hizo el desarrollo de un “hotfix” o arreglo que algún problema, pero no se verá reflejado en la línea principal o branch principal. Pero una vez aplicamos el merge, ya se vería reflejado el hotfix en el branch principal [6].

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

La Prueba Unitaria o Unittest es un método de prueba de software donde unidades individuales de código fuente de uno o más módulos de programación de computadora (datos de control asociado, procedimientos de uso y procedimientos de operación), se prueban para determinar si se pueden usar. Existen tres métodos: test_double, test_square y test_divide. Y luego de correrlo, tiene dos outputs “OK” que indica que todas las pruebas se corrieron de forma exitosa y “FAILED (failures=n)” que indica las n pruebas que fallaron y luego imprime la información asociada a las fallas [7].

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

Los assert or assertions en PyTest son revisiones que retornan o True o False como estatus. En este contexto de PyTest, si se falla el assert, se para la ejecución, es decir, el resto del código no se ejecuta. A continuación se muestra una imagen del uso de asserts:

```
assert "hello" == "Hai" is an assertion failure.
assert 4==4 is a successful assertion
assert True is a successful assertion
assert False is an assertion failure.
```

[8]

8) ¿Qué es Flake 8?

Flake 8 es una herramienta de linter de código invocada en Python. "Linting" significa ejecutar una herramienta contra el código que comprueba la sintaxis de de este y da instrucciones sobre cómo limpiarlo. Por lo tanto, utilizar Flake 8, previene errores y ahorra tiempo. Para implementarlo, se corre: "python<version> -m pip install flake8". Después, se ejecuta: "flake8 path/to/your_code/main.py #check particular file" "flake8 path/to/your_project/ #check the entire project repo" [9].

Referencias

[1] "Git vs. GitHub: What's the Difference?", Devmountain. [Online]. Available: https://blog.devmountain.com/git-vs-github-whats-the-difference/. [Accessed July 31, 2021].

[2] "About Branches", Github.com. [Online]. Available: https://docs.github.com/en/github/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-branches. [Accessed July 31, 2021].

[3] "Git Commit", GitHub. [Online]. Available: https://github.com/git-guides/git-commit. [Accessed July 31, 2021].

[4] "Seleccionando commits: Cherry-pick", *Runroom.com*, June 15, 2017. [Online]. Available: https://www.runroom.com/realworld/seleccionando-commits-cherry-pick [Accessed July 31, 2021].

[5] "Git Merge", *atlassian.com*. [Online]. Available: https://www.atlassian.com/git/tutorials/using-branches/git-merge#:~:text=Merging%20is%20Git's%20way%20of,them%20into%20a%20single%20branch. [Accessed July 31, 2021].

[6] N. Zoric, "How to Use Git Merge [the Correct Way]", *dev.to*, April 30, 2018. [Online]. Available: https://dev.to/neshaz/how-to-use-git-merge-the-correctway-25pd. [Accessed July 31, 2021].

[7] S. Mwangi, "Working with Pythons Unittest", March 22, 2021. [Online]. Available: https://www.section.io/engineering-education/working-with-pythons-unittest/. [Accessed July 31, 2021].

[8] "PyTest Tutorial: What is, How to Install, Framework, Assertions", *guru99.com*. [Online]. Available: https://www.guru99.com/pytest-tutorial.html#:~:text=assertion%20in%20PyTest.-,Assertions%20in%20PyTest,with%20the%20next%20test%20method. [Accessed July 31, 2021].

[9] "¿Qué es Flake8 y por qué deberíamos usarlo?", *ichi.pro*. [Online]. Available: https://ichi.pro/es/que-es-flake8-y-por-que-deberiamos-usarlo-202979474961394 [Accessed July 31, 2021].