Instituto Tecnológico de Costa Rica

Área Académica de Ingeniería Mecatrónica

MT-7003 Microprocesadores y microcontroladores

Tarea #1

GitHub, Pytest y Flake 8

Estudiantes:

Emmanuel Muñoz Peña - 2020026653

Sharon Vásquez Díaz – 2020023727

Profesor:

Ing. Rodolfo Piedra Camacho

Grupo 1

Fecha de entrega: viernes 17 de febrero del 2023.

I Semestre

2023

Preguntas Teóricas

1. ¿Explique la principal utilidad de git como herramienta de desarrollo de código?

Git consiste en una herramienta que se usa para almacenar distintas versiones de un archivo, esto de manera que cualquier versión se pueda recuperar en caso de ser necesario. Mundialmente Git se ha vuelto en el estándar para el control de versiones (conocidos como DVCS), este consiste en un sistema de control de versiones distribuido. Lo anterior quiere decir que se tiene un clon local del proyecto, que es un repositorio de control de versiones completo. Con los repositorios locales se les permite a los usuarios trabajar de forma remota y luego pueden sincronizar su copia con la del servidor. A diferencia del control de versiones centralizado, los clientes no tienen que sincronizar el código con un servidor antes de crear nuevas versiones. Además, se debe destacar que, al ser una herramienta de control de versiones, cuenta con un historial de los cambios que se realizan al código por los usuarios colaboradores. [1]

Esta herramienta cuenta con muchísimas ventajas, como el desarrollo simultaneo, ya que todos tienen su copia local del código y pueden trabajar al mismo tiempo en sus propias ramas. Por sus diferentes facilidades, Git ha sido integrado en la mayoría de las herramientas y productos, y funciona con cualquier equipo. [1]

2. ¿Qué es un branch?

Una branch o rama es una variación alterna y aislada del código que se está trabajando, se define como rama ya que estas surgen del tronco de los árboles, en el caso de los repositorios las branches se crean a partir de un código principal para el cual se quiere probar alguna función. Esto permite desarrollar nuevas funciones, corregir errores y probar nuevas ideas en el código sin afectar el código principal. [2]

3. En el contexto de github. ¿Qué es un Pull Request?

La función de Pull Request facilitan la colaboración entre los desarrolladores, estas ofrecen una interfaz que permite debatir los cambios propuestos antes de integrarlos al proyecto oficial. En otras palabras, consiste en realizar una solicitud para incorporar cambios, los cuales deben ser aprobados por los distintos colaboradores. Luego de que sea revisado el código de la rama, esta se debe fusionar con la rama *main*. [3]

4. ¿Qué es un commit?

Como se puede ver en (Referencia) un commit es una operación que logra enviar los últimos cambios realizados en el código trabajado al repositorio, en donde se guardan indefinidamente. Una vez hecho un commit igual se puede acceder a las versiones anteriores del código, pero este debe ser especificado, ya que por defecto se tiene la última versión "*commited*" disponible para revisión. [4]

5. Describa lo que sucede al ejecutar las siguientes operaciones: "git fetch" "git rebase origin/master"

Al utilizar el comando “git fetch” se le indica al git local que recupere la última información del archivo original, esto sin hacer transferencias de archivos. Se realiza más bien de manera que se compruebe si existe algún cambio disponible. Este comando generalmente es utilizado cuando se está trabajando con un repositorio clonado. [5]

Por otro lado, el comando “git rebase origin/master” implica que queremos rebasar una rama de la rama ascendente *master*. Por lo tanto “rebase origin/master” hace referencia que se desea rebasar una rama especifica de la rama ascendente *master*. Sin embargo, con este comando no se tendrán los nuevos commits en dicha rama, por lo cual, es necesario actualizar la rama local *master* antes de utilizar el comando rebase. Además, utilizar rebase permite que evitemos conflictos cuando se aplican commits en el repositorio local y no se han subido a uno remoto. [6]

Es importante destacar que si se utiliza el comando “*git fetch origin*” se van a descargar todas las ramas de dicho servidor, por lo cual, es necesario colocar en el comando rebase “*origin/master*” para indicar cuál rama se desea modificar. De manera que al utilizar ambos comandos los cambios locales se cargarán encima de los cambios remotos, dando como resultado un historial más limpio (a diferencia de cuando se utiliza el comando “*git pull*”), puesto que se tendrán menos confirmaciones de los commits que saturan el historial. [7]

6. Explique que es un “merge conflict” o “rebase conflict” en el contexto de tratar de hacer merge a un Pull Request o de completar una operación git rebase.

El *merge conflict* es un error que se da cuando git no puede resolver las diferencias entre dos *commits* de dos *branches* distintas a la hora de intentar fusionarlas con *merge*. Estos conflictos aparecen cuando dos o más personas hacen cambios en la misma línea del mismo código o una de estas intenta hacer un *pull* mientras otra persona haya eliminado este archivo. [8]

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

Una prueba unitaria, en inglés se conoce como Unittest, se utiliza para verificar el comportamiento de las unidades más pequeñas de la aplicación a utilizar, es decir, de los componentes individuales o fragmentos del código. En caso de los lenguajes orientados a objetos se trataría de una clase o un método de clase. Dichas pruebas funcionan aislando una parte determinada del código, con el fin de asegurar su funcionamiento. [9]

Tienen distintos beneficios, ya que la calidad del código mejorará y permiten que se reduzca el tiempo de depuración, de manera que el cliente se sienta más satisfecho. Estas pruebas sirven como documentación, por lo cual, nos permiten entender de mejor manera cada módulo del código. También evitan que se acumulen los errores en el código al identificarse con tiempo. [9]

8. Bajo el contexto de pytest. ¿Cuál es la utilidad de un “assert”?

Esta función es útil para el *debugging* de los códigos realizados, la función assert en pytest permite verificar los resultados esperados de una función mediante en el testeo de estas. [10]

9. ¿Qué es Flake 8?

Flake8 consiste en una librería de Python, esta consiste en un conjunto de herramientas para verificar su código fuente contra el PEP8, errores de programación y para verificar la complejidad ciclomática. El termino anterior hace referencia a una métrica de software para medir el número de rutas independientes a través del código fuente, es decir, a mayor cantidad de *ifs* en la función, más rutas tendrá y eso genera mayor complejidad ciclomática. [11]

Esta herramienta permite que el código final quedé limpio, puesto que revisa distintas estandarizaciones, las cuales permiten que se revise errores de lógica, complejidad, entre otros. [12] Cuenta con tres paquetes, estos son:

- PyFlakes: verifica la lógica del código. [12]
- Pycodestyle: este agrega verificación de estilos, anteriormente se conocía como PEP8. [12]
- Ned Batchelder's McCabe: este es el encargado de verificar la complejidad del código. [12]

10. Explique la funcionalidad de parametrización de pytest.

La parametrización en pytest logra reducir la cantidad de código necesaria para el testeo de una función y la recompilación del mismo, ya que parametrizando se puede lograr, por ejemplo, que se comprueben 3 valores de inputs serialmente una vez se empiece el testeo y de esta forma realizar de manera más dinámica el *debugging* del código.[13]

Referencias

[1] M. Jacobs *et al*., (2023, febrero 9). "*¿Qué es Git? - Azure DevOps*" [Online]. Disponible en: https://learn.microsoft.com/es-es/devops/develop/git/what-is-git

[2] GitHub Docs. *"About branches - GitHub Docs"* [Online]. Disponible en: https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-branches

[3] Atlassian. "*Pull-requests | Atlassian Git Tutorial*". [Online]. Disponible en: https://www.atlassian.com/es/git/tutorials/making-a-pull-request#:~:text=Las%20pull%20requests%20son%20una,integrarlos%20en%20el%20proyecto%20oficial

[4] GitHub Docs. *"About commits - GitHub Docs"* [Online]. Disponible en: https://docs.github.com/en/pull-requests/committing-changes-to-your-project/creating-and-editing-commits/about-commits

[5] J. Carrillo, (2021, enero 23). "*Git Fetch vs Pull: ¿Cuál es la diferencia entre los comandos Git Fetch y Git Pull*?" [Online]. Disponible en: https://www.freecodecamp.org/espanol/news/git-fetch-vs-pull-cual-es-la-diferencia-entre-los-comandos-git-fetch-y-git-pull/

[6] V. Vachhar, (2017, marzo 1). "*Git Rebase*" [Online]. Disponible en: https://varun.ca/rebase/#:~:text=git%20rebase%20origin%2Fmaster%20rebases,things%20such%20as%20squashing%20commits

[7] J. Wachira, (2022, agosto 16). "*Git Rebase Origin/Branch vs. Git Rebase Origin Branch*" [Online]. Disponible en: https://www.delftstack.com/howto/git/git-rebase-origin-master/

[8] GitHub Docs. "*About merge conflicts - GitHub Docs*" [Online]. Disponible en: https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/addressing-merge-conflicts/about-merge-conflicts

[9] KeepCoding, (2022, agosto 1). *"¿Qué son las pruebas unitarias de software?"* [Online]. Disponible en: https://keepcoding.io/blog/que-son-las-pruebas-unitarias-de-software/

[10] Pytest (2015). "*How to write and report assertions in tests — pytest documentation*" [Online]. Disponible en: https://docs.pytest.org/en/7.1.x/how-to/assert.html

[11] G. Díaz, (2020, junio 22). *"Escribiendo código de alta calidad en Python (2020) — Parte 2: linters"* [Online]. Disponible en: https://medium.com/@gonzaloandres.diaz/escribiendo-codigo-de-alta-calidad-en-python-parte-2-linters-64ffd8d2df91#:~:text=Flake8%20es%20una%20librer%C3%ADa%20de,para%20verificar%20la%20complejidad%20ciclom%C3%A1tica

[12] Manre's Universe, (2020, diciembre 12). *"Calidad de código en Python"* [Online]. Disponible en: https://manre-universe.net/calidad-de-codigo-en-python/

[13] Pytest (2015). "*Parametrizing tests"* [Online]. Disponible en: https://docs.pytest.org/en/7.1.x/example/parametrize.html