



Instituto Tecnológico de Costa Rica

Área Académica de Ingeniería en Mecatrónica

MT-7003 Microprocesadores y microcontroladores

Tarea #1

GitHub, Pytest y Flake 8

II Semestre 2023

Elaborado por:

Andrey Alfaro Hernández

2020124288

Noam Bermúdez Zamora

2019069843

Profesor:

Ing. Rodolfo Piedra Camacho

Preguntas Teóricas (20 pts, 2pts c/u)

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

La característica clave de Git es el desarrollo simultáneo de código, debido a que cada usuario tiene una réplica completa del proyecto en su máquina local, funcionando como un repositorio de control de versiones independiente. Esta capacidad ha convertido a Git en una herramienta esencial para el desarrollo de código, ya que permite a los diferentes desarrolladores trabajar en un proyecto de manera simultánea sin riesgo de pérdida de información. Esto es posible gracias a que Git registra y realiza un seguimiento de cada cambio realizado en el código fuente, lo que facilita la recuperación de versiones anteriores en caso de ser necesario. [1]

2) ¿Qué es un branch?

En el contexto de Git un “Branch” (Rama en español) es una línea de desarrollo alterna y aislada del código principal o “Master Branch”. Por lo que cada vez que se crea un Branch se crea una copia independiente del proyecto, lo que facilita el desarrollo de nuevas funciones y la corrección de errores, sin afectar el código principal. Además, posteriormente a la finalización del desarrollo sobre un Branch se puede combinar las nuevas características o cambios con la “Master Branch”. [2]

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

En el contexto de Github, un Pull Request, es una interfaz que permite a los desarrolladores de un repositorio solicitar cambios sobre el código fuente, lo que permite un debate integral sobre los cambios propuestos previamente a la integración sobre el proyecto. [3]

Así cuando un desarrollador desea cambiar el código principal debe crear una copia del repositorio (“Fork”) y realizar los cambios en esta. Posteriormente, debe solicitar una Pull Request que los propietarios del repositorio original revisarán y aceptarán si se considera adecuado. Así, la Pull Request debe poseer una descripción completa de los cambios solicitados y cualquier otra información relevante. Además, los propietarios pueden revisar la Pull Request y sugerir cambios extra o incluir comentarios en caso de ser necesario. [3]

4) ¿Qué es un commit?

Un commit es un comando que registra los cambios ejecutados sobre el código fuente de un proyecto. Además, un commit genera un historial de los cambios efectuados en el desarrollo del proyecto. El commit tradicionalmente incluye un mensaje detallado que denota los cambios realizados con el fin de simplificar la comprensión de los cambios entre desarrolladores. [4]

5) Describa lo que sucede al ejecutar las siguientes operaciones: “git fetch” “git rebase origin/master”

A) Git fetch:

La operación “git fetch” genera una descarga de commits, archivos y referencias desde un repositorio remoto hacia un repositorio local. Además, Git aísla el contenido descargado de cualquier versión local preexistentes de un repositorio local, ya que mantiene intacto el trabajo local y no lo sobrescribe de ninguna forma. Posteriormente, se pueden utilizar otros comandos para fusionar la versión descargada con la rama local. [5]

B) Git rebase origin/master:

El comando “Git rebase origin/master” genera una solicitud para replica de todos los commits presentes en una rama remota “origin/master” hacia la rama actual. Esto sobrescribe el historial de confirmación de forma que los cambios sobre la rama actual se vuelven a aplicar en la cima de la rama origen, como si estos se hubieran ejecutado sobre dicha rama de origen. [6]

Sin embargo, el proceso de ejecutar utilizando “get rebase origin/master” posee sus complicaciones, ya que Git pausa el proceso si encuentra algún conflicto entre los cambios locales y remotos. Por lo que si existe conflicto se debe resolver los problemas de forma manual para poder proseguir con el rebase. Esto genera que los commits locales se apliquen uno a la vez, creando una nueva secuencia de commits, que permiten eliminar commits insignificantes u obsoletos y así mantener un orden más claro en el proyecto. [6]

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.

Un merge conflict o rebase conflict sucede cuando en el intento de realizar un Pull Request o una operación git rebase hay dos commits que modifican la misma línea de código generando un conflicto pues Git no sabe cuál de los dos cambios aplicar. Se puede prevenir si se trabaja en distintos branches y cuando se realiza el merge se solucionan posibles ediciones que sean conflictivas. [7]

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

Las pruebas unitarias son una forma de comprobar que un fragmento de código funciona correctamente. Consisten en aislar una parte del código y comprobar que funciona a la perfección y validan el comportamiento de un objeto y la lógica. Con ellas se detectan antes errores que, sin las pruebas unitarias, no se podrían detectar hasta fases más avanzadas. [8]

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

Assert se usa para realizar comprobaciones en Python del cumplimiento de una expresión. Se pueden realizar distintas comprobaciones como sobre el tipo de variable, una expresión de igualdad o si se contiene algún dato dentro de una lista. En caso de que se cumpla devuelve el valor True, caso contrario el valor False. [9]

9) ¿Qué es Flake 8?

Es una librería Python que consiste en un kit de herramientas para comprobar el código base contra el estilo de codificación (PEP8), errores de programación y código que no está indentado. Ayuda a mejorar la legibilidad del código y vuelve los bugs más fáciles de detectar. [10]

10) Explique la funcionalidad de parametrización de pytest.

La funcionalidad de parametrización permite la evaluación de manera sencilla y potente de un código probando diferentes entradas y salidas con el mismo código de prueba, lo que parametriza las pruebas. Esto permite ejecutar una prueba con distintos parámetros, lo que hará que se realice esta prueba por cada set de parámetros. [11]

Referencias bibliográficas:

- [1] M. Jacobs., L. Casey., E. Kaim. (2023). “¿Qué es Git?” 2023 [Online]. Disponible en: <https://learn.microsoft.com/es-es/devops/develop/git/what-is-git>. [Accedido: Agosto 4, 2023].
- [2] GitHub, “About Branches”, 2023 [Online]. Disponible en: <https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-branches>. [Accedido: Agosto 4, 2023].
- [3] Atlassian. "Pull-requests | Atlassian Git Tutorial". [Online]. Disponible en: <https://www.atlassian.com/git/tutorials/making-a-pull-request>. [Accedido: Agosto 4, 2023].
- [4] GitHub, “Git Commit”, [Online]. Available: <https://github.com/git-guides/git-commit> [Accedido: Agosto 4, 2023].
- [5] Atlassian. "Git fetch | Atlassian Git Tutorial". [Online]. Disponible en: <https://www.atlassian.com/git/tutorials/syncing/git-fetch>. [Accedido: Agosto 4, 2023].
- [6] Atlassian. "Git rebase | Atlassian Git Tutorial". [Online]. Disponible en: <https://www.atlassian.com/git/tutorials/rewriting-history/git-rebase>. [Accedido: Agosto 4, 2023].
- [7] Simplilearn. “How to Resolve Merge Conflicts in Git?”. [Online]. Disponible en: <https://www.simplilearn.com/tutorials/git-tutorial/merge-conflicts-in-git> [Accedido: Agosto 4, 2023].
- [8] YeePLY. “¿Qué son las pruebas unitarias y cómo llevar una a cabo?”. [Online]. Disponible en: <https://www.yeeply.com/blog/que-son-pruebas-unitarias/> [Accedido: Agosto 4, 2023].
- [9] Pytest. “How to write and report assertions in tests”. [Online]. Disponible en: <https://docs.pytest.org/en/7.1.x/how-to/assert.html> [Accedido: Agosto 4, 2023].
- [10] freeCodeCamp. “Cómo autoformatear tu código Python con Black”. [Online]. Disponible en: <https://www.freecodecamp.org/espanol/news/como-autoformatear-tu-codigo-python-con-black/> [Accedido: Agosto 4, 2023].
- [11] El código Perfecto. “¿Pytest vs Unittest! La batalla definitiva: Descubre por qué Pytest conquista a los desarrolladores de Python”. [Online]. Disponible en: <https://elcodigoperfecto.blog/2023/05/29/pytest-vs-unittest-la-batalla-definitiva-descubre-por-que-pytest-conquista-a-los-desarrolladores-de-python/> [Accedido: Agosto 4, 2023].