

Tarea 1: GitHub, Pytest y Flake 8

Marco Flores Vargas – Sebastián Solano Montero

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

Git es un sistema de control de versiones, su utilidad principal es generar un repositorio de control de versiones. Esto permite que se pueda trabajar sin conexión o de manera remota, de manera que el desarrollador puede sincronizar su trabajo después de comprobarlo localmente. En otras palabras permite al desarrollador crear versiones antes de tener que sincronizarlas con el servidor. [1]

2) ¿Qué es un branch?

Es una copia independiente del código base del proyecto, de manera que cada desarrollador guarda cambios en el código en su propio repositorio local. La creación de ramas es la manera de administrar la separación que generada por los cambios en el proyecto principal. Tras finalizar el trabajo en una rama se puede reintegrar (merged) a la rama principal. [1]

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

Es la manera en la que un desarrollador notifica sobre la incorporación de cambios en un repositorio, estos cambios se proponen por rama y esto garantiza que la rama tenga solamente trabajo finalizado y aprobado. [2]

4) ¿Qué es un commit?

Es confirmar cambios aplicado al proyecto de forma permanente, registra una nueva versión del proyecto. Un commit no se debe realizar hasta que el proyecto sea probado y sea funcional. [3]

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

“git fetch” obtiene los cambios de una rama remota, no aplica cambios en el directorio de trabajo local. [4]

“git rebase origin/master” git rebase sirve para modificar el historial de reposiciones, se puede reordenar, editar o combinar confirmaciones. Si se usa junto con origin/master reordena el historial de confirmaciones en la rama actual a uno basado en la rama principal. [5]

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.

Los “merge conflict” ocurren al combinar ramas que tienen cambios en conflicto, donde Git necesita ayuda a manera de intervención manual para decidir cuáles cambios incorporar. Estos conflictos suelen presentarse cuando múltiples personas modifican la misma línea en un archivo o cuando uno edita mientras otro elimina el mismo archivo. Estos conflictos

deben resolverse antes de fusionar Pull Request en GitHub. [6] La operación git rebase puede introducir conflictos similares si dos commits alteran la misma línea. En este caso, Git no puede determinar cuál cambio aplicar, lo que se conoce como “rebase conflict”. [7]

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

Una Prueba Unitaria o Unittest es un método utilizado para verificar el correcto funcionamiento del código. Esto se logra mediante la creación y ejecución de pruebas individuales de unidades. Se les llama pruebas unitarias porque descomponen las funciones del programa en comportamientos discretos y comprobables, que pueden ser evaluados como entidades separadas. [8]

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

La utilidad de un "assert" radica en su capacidad para verificar expectativas y valores en pruebas de Python. Permite utilizar la declaración estándar "assert" de Python para comprobar, por ejemplo, que una función devuelve un valor específico. Si esta afirmación no se cumple, se mostrará el valor de retorno de la llamada a la función. [9]

9) ¿Qué es Flake 8?

Flake8 es una herramienta que realiza un análisis estático del código en Python. Esta herramienta fusiona diversas utilidades ampliamente reconocidas para verificar el código, como pycodestyle (anteriormente conocida como pep8), pyflakes y McCabe. Flake8 examina el código fuente en busca de cuestiones relacionadas con el estilo, errores y posibles fallas de lógica. Además, brinda advertencias y recomendaciones con el objetivo de mejorar la calidad del código. [10]

10) Explique la funcionalidad de parametrización de pytest.

La funcionalidad de parametrización en pytest permite ejecutar diversos escenarios de prueba con distintos conjuntos de datos y argumentos. Esto resulta particularmente útil cuando se pretende evaluar una función o método utilizando diversas entradas, verificando así si su comportamiento es correcto en cada caso. La parametrización simplifica la escritura de código de prueba y simplifica la administración y ejecución de múltiples situaciones de prueba. [11]

Además, pytest permite parametrizar fácilmente funciones de prueba. Esto se logra a través de la función `pytest.fixture()`. La anotación `@pytest.mark.parametrize` posibilita la definición de múltiples conjuntos de argumentos y fixtures en funciones de prueba o clases. La función `pytest_generate_tests` posibilita definir esquemas personalizados de parametrización o extensiones para la parametrización. [12]

Referencias:

- [1] M. Jacobs, L. Casey, E. Kaim. (2023, Jul. 26) ¿Qué es Git? [Online]. Available: <https://learn.microsoft.com/es-es/devops/develop/git/what-is-git>
- [2] GitHub. (2023). Crear una solicitud de incorporación de cambios, documentación de GitHub. Available: <https://docs.github.com/es/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/creating-a-pull-request>
- [3] S. Dauzon, *Git controle sus versiones*, 2nd. Ed: Editions ENI: <https://www.ediciones-eni.com/libro/git-controle-la-gestion-de-sus-versiones-conceptos-utilizacion-y-casos-practicos-2a-edicion-9782409033988>
- [4] Lopez-Pellicer, Francisco J [et al.]. GitHub como herramienta docente. A: JENUI 2015. "Actas de las XXI Jornadas de la Enseñanza Universitaria de la Informática". Universitat Oberta La Salle ed. Andorra la Vella: Universitat Oberta La Salle, 2015, p. 66-73.
- [5] GitHub. (2023). Acerca de la fusión mediante cambio de base de Git. Available: <https://docs.github.com/es/get-started/using-git/about-git-rebase>
- [6] GitHub Docs (2023). About merge conflicts. Available at: <https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/addressing-merge-conflicts/about-merge-conflict>
- [7] GitHub Docs (2023). Resolving merge conflicts after a Git Rebase. Available at: <https://docs.github.com/en/get-started/using-git/resolving-merge-conflicts-after-a-git-rebase>
- [8] Microsoft Learn (2023). Datos fundamentales de las pruebas unitarias - Visual Studio (Windows) | Microsoft Learn. Available at: <https://learn.microsoft.com/es-es/visualstudio/test/unit-test-basics?view=vs-2022>
- [9] Holger Krekel and Pytest-dev team (2015). How to write and report assertions in tests - pytest documentation. Available at: <https://docs.pytest.org/en/7.1.x/how-to/assert.html>
- [10] Flake8 (2016). Your tool for style guide enforcement. Available at: <https://flake8.pycqa.org/en/latest/>
- [11] Holger Krekel and Pytest-dev team (2015). Parametrizing tests - pytest documentation. Available at: <https://docs.pytest.org/en/7.1.x/example/parametrize.html>
- [12] Holger Krekel and Pytest-dev team (2015). How to parametrize fixtures and test functions - pytest documentation. Available at: <https://docs.pytest.org/en/7.1.x/how-to/parametrize.html>