Tarea 1
GitHub, Pytest y Flake 8

A continuación, se presentan 2 asignaciones para entender y aplicar herramientas de desarrollo deproyectos de programación.

Preguntas Teóricas (16 pts, 2 pts c/u)

1) *¿Diferencie la herramienta Git de GitHub?*

En primer lugar, Git es un software de control de versiones utilizado por desarrolladores, donde se permite almacenar distintos archivos en diversas etapas de un proyecto; es decir, la herramienta posee la función de dar acceso a las versiones previamente guardadas para modificar su contenido o verificarlo.

Con respecto a GitHub, consiste en almacenamiento basado en la nube, lo cual permite tener diversos repositorios en la plataforma. Estos repositorios pueden ser accedidos por cualquier persona, ya que se encuentran de manera pública dentro de la plataforma. De esta forma, se permite copiar el código de un desarrollador para modificarlo, y a su vez se brinda la opción de compartir las modificaciones con el propietario. También, la plataforma permite la gestión de proyectos; por tanto, si se está desarrollando un código en conjunto se podrán dividir las asignaciones desde la plataforma para una mejor organización.

Finalmente, la diferencia entre las herramientas corresponde a que Git permite almacenar archivos en diversas etapas del proyecto, pero sus funciones son más de uso individual. Con respecto a GitHub, es almacenamiento basado en la nube y se puede incorporar las funciones de Git, con la diferencia de que se permite un trabajo colaborativo en tiempo real mediante la gestión de proyectos. [1]

2) *¿Qué es un branch?*

Un Branch (o una Rama) es un espacio destinado para que los desarrolladores puedan trabajar sobre un proyecto sin modificar el conjunto de archivos originales de este último. Existe un Branch “master” que se genera por defecto a partir de la ejecución de un comando, y su función puede ser la de recuperar el proyecto a su estado inicial en caso de ocurrir algún error.

Al crear un nuevo repositorio usualmente se generan dos Branches: la “master” y la nueva. Esta última funciona para que los desarrolladores puedan trabajar sobre el proyecto sin hacer uso de la “master”; ambas empiezan apuntando al primer dato del proyecto. Luego, en los proyectos colaborativos es posible usar múltiples Branches para obtener entornos independientes en los cuales cada usuario pueda trabajar en distintos archivos sin alterar los de los demás colaboradores. [2]

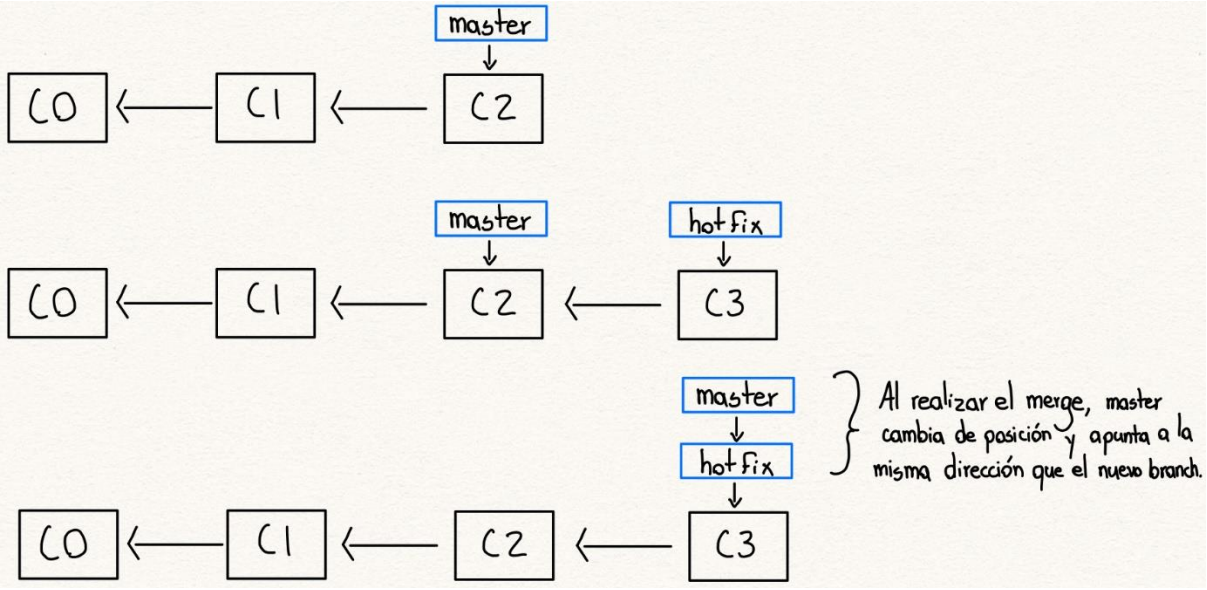
3) ¿Qué es un commit?

Un commit es un comando empleado para guardar o crear una copia del proyecto en cuestión; esta nueva copia se almacena en el repositorio. Al ser una especie de copia, permite modificar el proyecto, de modo que es posible regresar al estado donde se hizo el commit en caso de cometer algún error en las futuras versiones o ver cambios realizados antes de este. Asimismo, se pueden usar varios commits para editar partes diferentes de un proyecto, y estos no se sobrescribirán entre sí. Cuando este comando es llamado, se debe además incluir un mensaje corto (usualmente de longitud menor que 72 caracteres) acerca de los cambios a los que se está realizando el commit, denotado entre comillas. [3]

4) ¿Qué es la operación cherry-pick?

Es una operación que se basa en seleccionar un conjunto de archivos delimitados por el desarrollador. A manera de ejemplo, si en un Branch A existen X cantidad de archivos, pero solo se desea mover algunos de esos archivos a una Branch B (no en su totalidad), la operación cherry-pick permite seleccionar entre esos archivos la cantidad necesaria para ser trasladada a la Branch B. [4]

5) Explique de forma gráfica como cambia el “master” de un repositorio cuando se hace merge de un Branch.



[5]

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

Bajo este contexto, un Unittest se refiere a una prueba de software realizada mediante fragmentos de código para validar unidades de código fuente. Su propósito es asegurar que cada unidad de código (una clase, un método…) funciona de forma correcta, independientemente de otras unidades.

Los Unittests son creados por el desarrollador de un proyecto, y son las primeras pruebas

que se realizan con código escrito, con el fin de obtener menos errores y localizarlos fácilmente en el proyecto. Asimismo, mediante los Unittest se consigue facilitar la integración de unidades, reducir costos y conseguir una mejor compresión del código y su estructura. Sin embargo, cabe destacar que estas pruebas no demuestran por completo la ausencia de errores en el código completo. [6]

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

En el framework pytest, un “assert” es un comando usado para verificar los valores esperados en las pruebas de un código escrito en Python. Al emplear este comando, se asegura que una función retorne un valor con características específicas, de modo que en caso positivo el código continuará con su ejecución sin problema. No obstante, si el valor retornado por la función no es como se esperaba, el “assert” detendrá la ejecución del método y retornará un “AssertionError”, en donde se muestran los detalles de fallo de la función y un mensaje de error, el cual puede ser definido dentro del código. [7]

8) ¿Qué es Flake 8?

Flake 8 es una librería para el lenguaje de programación Python, la cual brinda al desarrollador diversas herramientas que le permiten corroborar el estado de su código fuente. Esto se puede usar para descubrir algún tipo de error de programación, o simplemente asegurarse que todo se encuentra de manera correcta (en especial con el número de caminos generado por las funciones “if” dentro de un código). Al implementar la librería se pueden encontrar operaciones como PyFlakes y pycodestyle. [8]

Bibliografía

[1] Kinsta, «Git vs Github: ¿Cuál es la Diferencia y cómo Empezar?,» 29 de diciembre, 2020. [En línea]. Disponible en: https://kinsta.com/es/base-de-conocimiento/git-vs-github/.

[2] Nube Colectiva, «Qué son las Ramas (Branches) en Git, cómo utilizarlas y otros detalles,» 21 de febrero, 2019. [En línea]. Disponible en: https://blog.nubecolectiva.com/que-son-las-ramas-branches-en-git-como-utilizarlas-y-otros-detalles/.

[3] J. Carrillo, «El Comando Git Commit Explicado,» 5 de febrero, 2021. [En línea]. Disponible en: https://www.freecodecamp.org/espanol/news/el-comando-git-commit-explicado/.

[4] F. Bozzo, «PlasticSCM: Qué es el Cherry Pick y cómo se usa,» 10 de octubre 2014. [En línea]. Disponible en: https://fdbozzo.blogspot.com/2014/10/plasticscm-que-es-el-cherry-pick-y-como.html.

[5] Git, «Git - Procedimientos Básicos para Ramificar y Fusionar,» [En línea]. Disponible en: https://git-scm.com/book/es/v2/Ramificaciones-en-Git-Procedimientos-B%C3%A1sicos-para-Ramificar-y-Fusionar.

[6] Ó. Moreno, «Pruebas unitarias: imprescindibles para programar,» 20 de agosto 2019. [En línea]. Disponible en: http://oscarmoreno.com/pruebas-unitarias/.

[7] PyTest Documentation, «How to write and report assertions in tests,» [En línea]. Disponible en: https://docs.pytest.org/en/latest/how-to/assert.html.

[8] G. Diaz, «Código de alta calidad en Python: linters,» 22 de junio, 2020. [En línea]. Disponible en: https://medium.com/@gonzaloandres.diaz/escribiendo-codigo-de-alta-calidad-en-python-parte-2-linters-64ffd8d2df91.