

Tarea 1 - GitHub, Pytest y Flake 8

MT-7003 Microprocesadores y microcontroladores

Estudiantes: Andrey Astorga Bogantes y Kevin Chinchilla Gómez

Preguntas Teóricas.

1. ¿Diferencie la herramienta Git de Github?

Git corresponde a un sistema de control de versiones distribuida, como tal constituye a una gran herramienta para desarrolladores, puesto que permite tener un registro y control de las versiones de un determinado proyecto o código. Por su parte GitHub, corresponde una comunidad orientada hacia los desarrolladores donde se tiene acceso a los distintos repositorios, como tal es una red social se utiliza para gestión y control de proyectos.

2. ¿Qué es un branch?

Se habla de un *Branch* o bien ramificación, cuando se toma la rama principal de desarrollo (o también conocida como máster) y a partir de esta se realizan distintos cambios o trabajos sin seguir la rama principal de desarrollo. Esto es de gran utilidad pues permite desarrollar nuevas funciones, o la corrección de errores, sin afectar la rama principal.

3. ¿Qué es un commit?

El comando *git commit* se utiliza para capturar una “instantánea” de los cambios preparados en ese momento del código como tal, estas capturas se consideran como versiones seguras del proyecto, este comando se encarga de enviar los cambios al repositorio local.

4. ¿Qué es la operación cherry-pick?

Esta operación nos permite tomar uno o varios *commits* de una rama y aplicarlo en otra, esto es de gran utilidad tanto en temas de funcionalidad, como para ejecutar un commit en una rama errónea y se desea corregir.

5. ¿Qué hace el comando git stash?

Este comando permite guardar de manera temporal los distintos cambios (los cuales se encuentran normalmente sin acabar o no están listos para aplicarse) realizados sin la necesidad de aplicarlos de manera inmediata, es de gran ayuda, por ejemplo, si se desea trabajar en distintas partes del proyecto sin ejecutar los cambios realizados de manera inmediata.

6. Compare las operaciones git fetch y git pull.

Operación Git Fetch: únicamente descarga nuevos datos desde un repositorio remoto(branch) y los almacena en un branch oculto llamado origin/master; sin embargo, no integra ninguno de estos nuevos datos en sus archivos de trabajo (git

local). Es decir, no hace ninguna transferencia de archivos, sino que se utiliza para obtener una vista nueva de todos los cambios o modificaciones que sucedieron en un repositorio remoto.

Operación Git Pull: actualiza el branch actual o git local con los últimos cambios del repositorio remoto(branch). Lo que implica que esta operación no solo descarga nuevos datos; sino que también los integra directamente en sus archivos de copia de trabajo actuales(git local o branch actual). Cabe destacar que la operación git pull no es más que la suma de las operaciones git fetch y git merge.

7. Asumiendo que usted está en un Branch llamado “secundario” y su Branch principal se llama “master” ¿Qué resultado espera de hacer git rebase master? ¿Qué resultado espera de hacer git rebase origin/master?

Git rebase master: Esta operación mueve todo el Branch “secundario” y lo coloca en el último commit del Branch “master”, manteniéndose como Branch “secundario”

Git rebase origin/master: Esta operación mueve todo el Branch “secundario” y lo coloca de forma que todos los commits que contiene este Branch “secundario” se conviertan ahora en los últimos commits del Branch “master” combinando todos los cambios del Branch “master” en un solo “parche” y luego integrarlo junto al Branch “secundario”. De esta forma, todos los commits se almacenan en el Branch “master”.

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

Una Prueba Unitaria o Unittest es una prueba que permite examinar, de forma rápida y fácil, el correcto funcionamiento de cada parte del código según lo previsto, antes de que todas las partes del mismo ocupen su lugar correspondiente en el código o programa general. De igual forma, con estas pruebas es posible comprobar los componentes individuales de los programas informáticos.

Su objetivo consiste en la comprobación frecuente de diversos componentes, por lo que estas pruebas suelen realizarse de forma automática. Por otro lado, el Unittest le ahorra tiempo al desarrollador al facilitar hacer cambios grandes de forma rápida para mejorar el código.

Las pruebas unitarias mejoran la capacidad de administración en relación con términos de visibilidad, informes, control, corrección, eficiencia, velocidad, planificación y satisfacción del cliente. También, ayudan a la comprensión del código que se está desarrollando. Además, ayudan al desarrollador a comprender el diseño del código en el que está trabajando. En lugar de escribir el código para hacer algo, comienza describiendo todas las condiciones a las que está sometiendo el código y las salidas que espera de eso.

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

El comando “assert”, así como en Python, se utiliza para verificar las expectativas y los valores en las pruebas, es decir, comprobar si las funciones retornan los valores

esperados. De igual forma, permite verificar los valores de las subexpresiones más comunes, incluidas llamadas a variables, atributos, comparaciones y operadores binarios y unarios.

10. ¿Qué es Flake 8?

Es una buena herramienta de análisis estático del código fuente como el código Python. La cual verifica “pep8”, “pyflakes” y “complejidad circular”. Donde “pep8” es una guía que facilita la lectura del código en Python y la consistencia entre programas de distintos usuarios; “pyflakes” es una herramienta de análisis de código Python, y por último, la “complejidad circular” es una métrica de software que se utiliza para medir el número de rutas independientes a través del código fuente.