Tarea 1 GitHub, Pytest y Flake8
MT-7003 Microprocesadores y microcontroladores
Cristofher Solís Jiménez y Jose David Soto Zúñiga

Preguntas Teóricas (24 pts, 2pts c/u)

1) ¿Qué es Git?

Consiste en un software diseñado para el control de versiones (lleva el registro de cambios en el código fuente) en proyectos de desarrollo de programas. Permite la coordinación entre integrantes de un trabajo que involucre archivos compartidos y una mejor organización al tener información sobre cambios realizados.

2) ¿Que es Github?

Github es una plataforma de red social para desarrolladores, de gestión de proyectos y control de versiones de código. Su principal función es trabajar en comparación con personas de todo el mundo así como planificar proyectos y realizar un seguimiento de trabajo.

3) ¿Qué es un branch?

Una rama consiste en una manera de continuar con el desarrollo de un proyecto sin afectar la sección principal de este. Permite la construcción de software, estableciendo una rama principal, sobre la cual “crecen” (se construyen) las ramas que consisten en una serie de commits, viéndose como un añadido a la rama principal.

4) ¿Qué es un commit?

En github, el comando git commit captura o guarda la versión actual del proyecto, realizando así versiones “seguras” del proyecto. El comando abrirá un editor de texto el cual pedirá un mensaje de confirmación. Una vez escrito el mensaje, guarda el archivo y cierra el editor.

5) ¿Qué es la operación cherry-pick?

Consiste en agarrar un commit particular de una rama y aplicarlo a otra rama, se dice que añade a la historia.

6) ¿Qué hace el comando git checkout?

El comando git checkout permite desplazar las ramas creadas anteriormente con el comando branch. Al realizar ese cambio de rama, se actualizan los archivos en esa rama y se le indica a Git que registre todas las confirmaciones realizadas en dicha rama, se puede ver como seleccionar la línea de desarrollo para trabajar.

7) ¿Qué hace el comando git stash?

Este comando permite “reservar”, en una pila de cambios inacabados, el estado en el que se encuentra el directorio de trabajo. De esta manera se puede retomar el trabajo incluso desde otra branch y sin verse en la obligación de realizar un commit de una sección inconclusa.

8) Compare las operaciones git fetch y git pull

Git fue diseñado para que el cliente y el servidor no necesariamente se encuentren conectados siempre, por lo que es posible trabajar desconectado, para que esto sea posible, Git mantiene dos repositorios locales, uno con el código en el servidor y otro con el repositorio remoto. Git puede descubrir los cambios necesarios incluso cuando no se puede acceder al repositorio remoto. Más tarde, cuando se necesite enviar los cambios a otra persona, git puede transferirlos como un conjunto de cambios desde un punto en el tiempo conocido por el repositorio remoto.

git fetch es el comando que dice "actualizar mi copia local del repositorio remoto".

git pull se encarga de decir "llevar los cambios en el repositorio remoto a donde guardo mi propio código".

Por lo tanto es importante reconocer que siempre hay al menos tres copias de un proyecto. Una copia es su propio repositorio con su propio historial de confirmación. La segunda copia es su copia de trabajo en la que está editando y construyendo. La tercera copia es su copia local "en caché" de un repositorio remoto.

9) ¿Qué hace el comando git reset ~HEAD?

Primeramente, es importante rescatar lo que realiza git reset; este comando permite deshacer cambios realizados en el proyecto, es flexible pues es posible elegir hasta qué punto se desea regresar. Ahora bien “HEAD~” es una forma corta de referirse al antecesor directo del commit que se encuentre en master. Por lo tanto, git reset HEAD~ consiste en deshacer los cambios hasta el padre del commit ubicado en master branch.

10) ¿Qué es Pytest?

Pytest es una librería de python que permite realizar pruebas simples durante el desarrollo de código, para utilizarlo, se utilizan archivos con el prefijo «test_», entonces al utilizar el siguiente código se puede realizar la prueba de que funcione y a la vez recibir un mensaje de error en caso contrario. Por ejemplo:
def test_”nombre de la función”
assert “nombre de la función” == “resultado esperado”

A su vez, pytest brinda opciones útiles como lo son el fixture y el patching, que permiten reutilizar el mismo fragmento de código en múltiples funciones o modificar el

funcionamiento de la llamada de una función respectivamente. Asimismo, permite dar a conocer tanto los casos de éxito como de error.

11) Bajo el contexto de pytest ¿Que es un “assert”?

Un assert es una forma de definir un tipo de error en el código desarrollado, con lo cual se pueden detectar problemas de manera temprana y controlada. Es posible modificar el mensaje de error mostrado según se desee así como la condición para que este se muestre.

12) ¿Qué es Flake 8?

Cuando se programa en equipo, es importante tener un código lo más legible posible, por lo que hay estándares como PEP8 o pycodestile los cuales dictan reglas para lograr códigos más sencillos de leer.

La herramienta flake8 permite no solo revisar que se cumpla con la norma de PEP8 sino que también permite revisar si existen variables no utilizadas, imports no necesarios entre otras líneas de código que no son relevantes para la función del programa, lo que permite tener un código más limpio. Además existe el plugin pep8_naming, el cual revisa la nomenclatura de las funciones, variables, etc.