

Estudiantes: Tristan Bodra Jiménez y Jesús Fallas Barboza.

Tarea 1

GitHub, Pytest y Flake 8

Preguntas teóricas:

1) ¿Diferencie la herramienta Git de Github?

La principal diferencia entre Git y Github es que Git es una herramienta de código abierto que los desarrolladores instalan localmente, que permite al usuario gestionar el historial de código fuente de un proyecto en desarrollo. Por otro lado, Github es más como una “red social” para desarrolladores, construida alrededor de la herramienta Git. Aquí los desarrolladores que utilizan Git pueden conectarse y cargar o descargar recursos.

2) ¿Qué es un branch?

Branch de “rama”, es una función que presentan la mayoría de herramientas de control de versiones. Las ramificaciones hacen referencia a tomar la rama principal de desarrollo del código (rama master) y de ahí continuar trabajando sin seguir esa rama principal, si no en otra distinta. Esto puede ser muy útil cuando se quiere añadir una nueva función o solucionar un error; hacerlo por una nueva rama puede evitar inestabilizar el código base principal y da la oportunidad de limpiar el historial futuro antes de fusionarlo con la rama principal.

3) ¿Qué es un commit?

En términos de Git, el comando git commit captura una instantánea de los cambios preparados en ese momento del proyecto. Estas instantáneas son prácticamente versiones “seguras” de un proyecto. En pocas palabras un commit define el final de una transacción ejecutada con éxito.

4) ¿Qué es la operación cherry-pick?

Este es un potente comando que permite, por medio de referencias, añadir al actual “head” de trabajo las confirmaciones arbitrarias de Git. La ejecución de este comando es prácticamente elegir una confirmación de una rama y aplicarla a otra. Hay que tener ciertas consideraciones a la hora de aplicar esta operación, ya que Cherry-pick puede generar duplicaciones de confirmaciones, por lo que se suelen preferir funciones tradicionales.

5) ¿Qué hace el comando git stash?

Este comando permite tomar cambios realizados en el código sin confirmar y guardarlos aparte para poder utilizarlos más adelante. Estos cambios son guardados en “stashes” y son muy prácticos ya que permiten a la persona que este realizando ese cambio en el código cambiar rápidamente de contexto si es el caso y ponerse a hacer otras cosas, aún cuando dichos cambios no están listos para ser confirmados.

6) Compare las operaciones git fetch y git pull

El comando git fetch se usa más que todo para ver el progreso del trabajo de otras personas, puesto que este hace referencia a descargar commits, archivos y referencias de un repositorio remoto al repositorio local, esto sin la necesidad de fusionar los cambios en el nuestro repositorio. Por otro lado, git pull también extrae y descarga contenido desde repositorios remotos pero este por el contrario actualiza el repositorio local al instante con dicho contenido. En realidad el comando git pull termina siendo una combinación entre los comandos git fetch y git merge, donde este último es el que fusiona ambas partes como tal.

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?

Cuando ejecutamos el comando git rebase, propiamente en una rama principal, lo que esperamos es prácticamente integrar cambios de otra rama en esta. Respecto al comando git rebase origin/master se da algo similar, pues ambas formas usan la sintaxis de git revisions para nombrar una confirmación específica, pero esta hace referencia con a cualquier rama, es decir, este comando es usado típicamente cuando hay varias personas trabajando en el mismo main o master, donde cabe la posibilidad de que se hayan efectuado algunos cambios mientras se trabajaba en una rama aparte, entonces con este comando se esperaría reorganizar la rama principal tomando en cuenta esos posibles commits efectuados, realizando la reorganización de forma más segura para evitar incompatibilidades. En general estos comandos nos permiten mantener un historial del proyecto de forma lineal al cambiar la base de una rama de una confirmación a otra de forma que parezca que se creó desde una confirmación distinta.

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

La prueba unitaria o unit test hace referencia a un método para probar una unidad estructural de código. Dicho método tiene las características de probar pequeñas cantidades código, ser independiente al no depender ni afectar otras pruebas unitarias, probar métodos públicos para que la prueba no sea pues frágil a cambios en la implementación, automatizable (sin necesidad de intervención manual), repetible y predecible, y por último profesionales (con las mismas exigencias del código en general como documentación, profesionalidad, etc.). Esta práctica se suele querer esquivar por parte de los programadores por cuestiones de tiempo aunque es de suma importancia llevarla a cabo para evitar deficiencias en el código.

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

Bajo el contexto de pytest, el comando assert hace referencia a un método de verificación para el resultado de un método. En otras palabras, con este comando se busca corroborar por medio de una “igualdad” (masomenos) que un método este retornando el valor esperado según suposiciones del desarrollador.

10) ¿Qué es Flake 8?

La herramienta Flake 8 nos permite obtener códigos más limpios y puntuales. Esta es un librería para el software python que contiene un gran conjunto de herramientas como PyFlakes, pycodestyle y el script

McCabe de Ned Batchelder que nos ayudan a verificar el cumplimiento de la PEP-8, que no hayan errores como importación de librerías que no se estén utilizando, así como aspectos de la complejidad ciclomática del código.