Tarea 1

GitHub, Pytest y Flake 8

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

Preguntas Teóricas (20 pts, 2pts c/u)

1) ¿Explique que es git y su relación con github?
2) ¿Qué es un branch? ¿Qué es un fork?
3) En el contexto de github. ¿Qué es un Pull Request?
4) ¿Qué es un commit?
5) 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.
6) ¿Qué es una Prueba Unitaria o Unittest en el contexto de desarrollo de software?
7) Bajo el contexto de pytest. ¿Cuál es la utilidad de un "assert"?
8) ¿Explique que son github-actions y su utilidad para el desarrollo continuo de código?
9) ¿Qué es Flake 8?
10) Explique la funcionalidad de parametrización de pytest.

Sección Práctica (55 pts, puntos varían según la sección)

La sección práctica busca ejercitar las tres herramientas vistas en la sección teórica para esto realice los siguientes pasos:

1) Cree un nuevo repositorio público en github. El nombre debe de ser: TareasPrimerApellidoEstudiante1PrimerApellidoEstudiante2. Dentro de este repositorio cree una carpeta llamada "Tarea 1", todos los documentos de esta tarea mencionados en los siguientes puntos deben ir en esta carpeta. (**1.5 pts**)
2) Suba todas las respuestas de la parte teórica al repositorio como un primer commit, asegúrese de subir las respuestas como un archivo PDF. Asegure usar una descripción intuitiva al realizar su commit. (**1.5 pts**)
3) Utilice pytest para probar completamente una serie de funciones de python. El código por probar debe de ser el siguiente (**Total: 30 pts**):

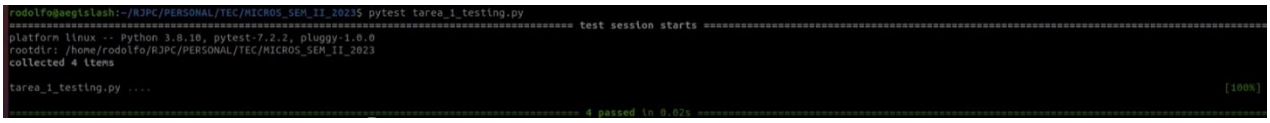
   Dos métodos distintos **(5 pts c/u)**:
   1. Un método "**count_char**" recibe dos parámetros: Cadena y Caracter. Y posee el siguiente comportamiento.
      a. Debe verificar que el parámetro cadena sea un string. En caso de no cumplir esta limitación el código retorna un **código de error único.**
      b. Debe verificar que el parámetro cadena solo contenga letras del abecedario y números del 0 al 9. En caso de no cumplir esta limitación el código retorna un **código de error único.**

   c. Debe verificar que el parámetro carácter sea un único carácter del abecedario o números del 0 al 9. Caso contrario debe retornar un **código de error único.**
   d. Debe contar la cantidad de veces que el parámetro carácter aparece en el parámetro cadena. En caso de hacer esto debe de retornar **un código de éxito único.**
   e. El método siempre devuelve dos valores. El código de error/éxito, la cantidad de veces que aparece el carácter en la cadena en ese orden específico. En los casos con código de error en vez de retornar la cantidad de apariciones se retorna None.
2. Un método "**multiplo_2**" que recibe dos parámetros de entrada "base" y "múltiplo" y debe de:
   a. Debe verificar que los parámetros de entrada no sean de tipo string, sino enteros positivos. En caso de no cumplirse retorna un **código de error único**.
   b. Debe verificar que el valor de múltiplo sea uno entre: 1, 2, 4, 8, 16. En caso de no cumplirse retorna un **código de error único**.
   c. Calcula la operación $multiple * base$. Sin usar sumas, multiplicaciones o ciclos for. Cualquier uso de estas operaciones sea en librerías o en operandos de Python se traduce a un resultado de 0 en esta función. En caso de hacer esto debe de retornar **un código de éxito único.**
   d. El método siempre devuelve dos valores. El código de error/éxito y el resultado de la operación en ese orden específico. En los casos con código de error en vez de retornar el resultado se retorna None.

Este código debe de ser su propio archivo .py.

Adjunto a este enunciado usted puede encontrar un archivo de Python llamado **tarea_1_testing.py.** Este archivo contiene los test de pytest que van a ser utilizados para evaluar el código, por cada test que el código pase exitosamente se darán 5pts hasta un total de **20 pts**. El archivo de pruebas debe de poder ejecutar su código sin necesidad de ser modificado, para esto contemple las siguientes indicaciones:

- El archivo de pruebas contiene la forma en que se debe de llamar el archivo con su código.
- El archivo de pruebas explica los códigos de error que se esperan obtener de las funciones
- El archivo de pruebas asume que el archivo con el código que define los métodos se encuentra en la misma carpeta.
- Usted está en libertad de descargar el archivo de pruebas y correrlo localmente para asegurar el funcionamiento de su código.
- Se espera ver lo siguiente al ejecutar el archivo de pruebas:



4) Use flake8 y asegure que su código no presenta errores de formato. (**5 pts**)
5) Suba el código de funciones y el código de pruebas a su repositorio en github como un commit.
6) Cree un branch del master de su repositorio e introduzca un cuarto commit en el cual se agrega un nuevo archivo con 3 errores identificables por flake 8. No puede reutilizar los archivos de los puntos 4 y 5 para esto. (**5 pts**)
7) En el mismo Branch del paso 6 agregue otro commit arreglando los errores de flake 8 del paso 6 (**5pts**)
8) Suba el branch a su repositorio de github. NOTA: Esto implica que el master de su repositorio va a estar dos commit por detrás del Branch.
9) Realice un pull request para poder llevar el commit de su branch al master. El pull request puede estar mergeado o solo creado. (**2 pts**)

Instrucciones Generales

1) La tarea debe ser realizada en parejas
2) Fecha de entrega viernes 15 de agosto.
3) Entregables: Link al repositorio de github con todos los archivos y branches solicitados.
4) El repositorio deberá nombrarse como: TareasPrimerApellidoEstudiante1PrimerApellidoEstudiante2, de no seguir esta indicación se le descontarán 10 puntos.
5) Ausencia de comentarios en el código se traducirá a una reducción de 10 puntos de la nota final. Comentarios sustanciales explican los parámetros de entrada y salida de un método, una explicación del método como tal, comentarios dentro del código que orienten al lector para entender la solución.
6) Al TEC digital solo entregan un archivo de texto con el link al repositorio

Recomendaciones

Usar el tutorial de git conocido como http://gitimmersion.com/ esto provee el conocimiento básico suficiente para realizar la tarea.

Para entender el concepto de "un código de error" se sugiere observar el estándar de errores de sistema de Linux: https://mariadb.com/kb/en/operating-system-error-codes/.

En ocasiones conviene descargar una máquina virtual e instalar Linux. Esto solamente para facilitar el uso de herramientas como pytest desde la terminal.

Hacer uso de la documentación oficial de pytest: https://docs.pytest.org/en/latest/getting-started.html