

Tarea 1: GitHub, Pytest y Flake 8

Preguntas Teóricas:

1) ¿Explique que es git y su relación con github?

- Git es un sistema de control de versiones de código, gratuito y de código abierto, el cual puede ser utilizado para gestionar proyectos pequeños o grandes. Por su parte, GitHub es una plataforma en la nube que usa Git como su base para gestionar repositorios de código. Git se encarga de revisar los cambios realizados a los archivos, mientras que GitHub proporciona una interfaz web que permite colaborar, compartir código y gestionar proyectos.

2) ¿Qué es un branch? ¿Qué es un fork?

- Branch: es una versión paralela de un repositorio. Este está incluido en el repositorio, pero no afecta a la rama primaria o principal, lo que permite trabajar libremente sin interrumpir la versión “activa”. Una vez que se hayan realizado los cambios deseados, puede volver a combinar la rama en la rama principal para publicarlos.

- Fork: es una copia personal del repositorio de otro usuario que vive en tu cuenta. Los *forks* te permiten hacer cambios libremente en un proyecto sin afectar el repositorio original (*upstream*). También puedes abrir una *pull request* en el repositorio *upstream* y mantener tu *fork* sincronizado con los cambios más recientes, ya que ambos repositorios siguen estando conectados.

3) En el contexto de github. ¿Qué es un Pull Request?

- Un Pull Request es una solicitud para efectuar cambios a un repositorio de GitHub. Permite que otros colaboradores del repositorio revisen, comenten y aprueben los cambios antes de integrarlos en la rama principal u otra rama objetivo.

4) ¿Qué es un commit?

- Un *commit*, o “revisión”, es un cambio individual en un archivo (o en un conjunto de archivos). Cuando realizas un *commit* para guardar tu trabajo, Git crea un identificador único (también llamado “SHA” o “hash”) que permite llevar un registro de los cambios específicos realizados, junto con quién los hizo y cuándo. Los *commits* suelen incluir un mensaje de *commit*, que es una breve descripción de los cambios realizados.

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.

- Un *merge conflict* o *rebase conflict* ocurre cuando Git no puede combinar automáticamente cambios entre ramas debido a modificaciones incompatibles en las mismas líneas o áreas de un archivo. Esto puede pasar en un *merge* o durante un *rebase*. El desarrollador debe resolver manualmente el conflicto editando los archivos afectados y luego continuar con la operación.

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

- Una prueba unitaria es una verificación automatizada que comprueba que una unidad mínima de código (como una función o método) funciona según lo esperado. Se ejecuta de forma aislada para garantizar que el comportamiento sea correcto sin depender de otras partes del sistema.

7) Bajo el contexto de pytest. ¿Cuál es la utilidad de un “assert”?

- Un *assert* es una herramienta de depuración y prueba de código. Este permite verificar ciertas condiciones dentro de las pruebas y en caso de que una condición no se cumpla, pytest genera un error indicando que la prueba falló.

8) ¿Explique que son github-actions y su utilidad para el desarrollo continuo de código?

- GitHub Actions es una plataforma integrada dentro de GitHub diseñada para automatizar flujos de trabajo de desarrollo de software directamente en tu repositorio. Permite automatizar tareas como construcción, pruebas y despliegue del código mediante pipelines definidas.

Este tipo de herramienta es útil para el desarrollo continuo de código debido a que permite implementar prácticas de CI/CD (Integración Continua y Entrega/Despliegue Continuo) directamente desde tu código, sin depender de herramientas externas.

- Integración Continua (CI): automatiza el proceso de compilar y ejecutar pruebas cada vez que se integran cambios, garantizando que el código siempre se mantenga en estado funcional.
- Entrega/Despliegue Continuo (CD): tras validarse, el mismo pipeline puede encargarse de desplegar el software a entornos de producción o preparación de manera automatizada.

9) ¿Qué es Flake 8?

- Flake8 es una herramienta de análisis de código Python que combina PyFlakes, pycodestyle y otras utilidades. Permite detectar errores de sintaxis, errores de formato y algunos posibles errores lógicos, ayudando a que el código cumpla con las guías de formato PEP 8 (cuyas siglas en inglés significan Python Enhancement Proposal) y sea más consistente.

10) Explique la funcionalidad de parametrización de pytest.

- La parametrización es una funcionalidad que permite ejecutar el mismo caso de prueba varias veces con diferentes conjuntos de datos, sin necesidad de duplicar el código del test.

Esto se realiza por medio de `@pytest.mark.parametrize`, que recibe:

1. El nombre o nombres de los parámetros que recibirá la función de prueba.
2. Una lista de valores o tuplas que representan los diferentes casos de entrada.

Luego, al realizar la prueba, este se realiza la cantidad de veces que tenga de casos dentro de la parametrización.