

Preguntas Teóricas

- 1) **¿Explique la principal utilidad de git como herramienta de desarrollo de código?**
 - a. Git es el sistema de control de versiones más utilizado. Git rastrea los cambios que realizas en los archivos, por lo que tienes un registro de lo que se ha hecho y puedes volver a versiones específicas si alguna vez lo necesitas. Git también facilita la colaboración, permitiendo que los cambios de varias personas se integren en una única fuente.
- 2) **Explique la diferencia entre git y github**
 - a. GitHub facilita la colaboración con git. Es una plataforma que puede mantener repositorios de código en almacenamiento basado en la nube para que varios desarrolladores puedan trabajar en un solo proyecto y ver las ediciones de cada uno en tiempo real, además, también incluye funciones de organización y gestión de proyectos.
- 3) **¿Qué es un branch?**
 - a. En Git, las ramas actúan como espacios o entornos separados que permiten a los desarrolladores trabajar en un mismo proyecto sin afectar o sobrescribir los archivos originales.
- 4) **4) En el contexto de github. ¿Qué es un Pull Request?**
 - a. Una solicitud de extracción, o pull request (PR) en su forma abreviada, es una función de los sistemas de control de versiones como Git. Se trata de una petición para combinar código de una rama a otra.
- 5) **¿Qué es un commit?**
 - a. Un commit es básicamente crear un snapshot de un momento determinado en la historia del código y crea una nueva versión del código en el repositorio actual
- 6) **Describe lo que sucede al ejecutar la siguiente operación: “git rebase main”.**
 - a. El comando git rebase main se usa para aplicar los cambios de tu rama actual sobre el último commit de la rama main.
- 7) **Explique que es un “merge conflict” y como lo resolvería.**
 - a. Un conflicto de fusión (o merge conflict) en Git ocurre cuando Git no puede automáticamente combinar dos ramas porque hay cambios contradictorios en los mismos archivos o líneas de código. Para resolverlo se editan los archivos marcados por Git para resolver los conflictos manualmente.
- 8) **¿Qué es una Prueba Unitaria o Unittest en el contexto de desarrollo de software?**
 - a. Un unittest se basa en la creación de casos de prueba que ejecutan unidades específicas del código y verifican si el resultado es el esperado.
- 9) **Bajo el contexto de pytest. ¿Cuál es la utilidad de un “assert”?**
 - a. La declaración assert en Python se usa para realizar una comprobación en el código y, si la condición especificada es falsa, se genera una excepción AssertionError. A continuación se muestra un ejemplo del uso de la función assert

```
# Código a probar
def suma(a, b):
    return a + b

# Archivo de prueba (test_mi_codigo.py)
def test_suma():
    assert suma(2, 3) == 5 # Verifica que la suma de 2 y 3 es igual a 5
    assert suma(-1, 1) == 0 # Verifica que la suma de -1 y 1 es igual a 0
    assert suma(0, 0) == 0 # Verifica que la suma de 0 y 0 es igual a 0
```

10) Mencione y explique 3 errores de formato detectables con Flake8

- a. Errores de indentación, Flake8 verifica que el código esté correctamente indentado, siguiendo las convenciones de estilo de PEP 8 (Python Enhancement Proposal).
- b. Líneas Demasiado Largas, PEP 8 recomienda que las líneas de código no superen los 79 caracteres de longitud para mantener la legibilidad. Flake8 detecta líneas que exceden este límite y genera una advertencia.
- c. Falta de Espacios Alrededor de Operadores, Flake8 revisa el uso adecuado de espacios alrededor de operadores binarios y de asignación. Los errores comunes incluyen la falta de espacios alrededor de operadores como +, -, *, = y otros.