

Preguntas Teóricas

1) ¿Explique la principal utilidad de git como herramienta de desarrollo de código?

Git es una herramienta de control de versión que permite al programador el seguimiento y administración de los cambios en su código fuente.

2) Explique la diferencia entre git y github.

Git es un software de control de versión de código abierto, que se encuentra de manera local en el ordenador. Mientras que github es un sitio web que permite compartir el control de versión para trabajar de manera colaborativa sobre un código o proyecto.

3) ¿Qué es un branch?

Un “branch” es una versión alterna al código de la “master branch” o “main branch”. Permite probar cambios en el código sin afectar el “código principal”.

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

En caso de que los cambios hechos al código en un branch sean satisfactorios, se puede solicitar la revisión por los integrantes del grupo. Si los cambios son satisfactorios, se realiza un “merge”, para integrar el branch al código principal (main branch).

5) ¿Qué es un commit?

Un commit es la forma que proporciona git de guardar el progreso de los cambios hechos sobre un código.

6) Describa 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. Faltan 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.