

Preguntas Teóricas (20 pts, 2pts c/u)

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

Git es un sistema de control de versiones que permite a los desarrolladores seguir los cambios en el código fuente durante el desarrollo del software. Su principal utilidad radica en su capacidad para gestionar diferentes versiones del código de manera eficiente, facilitando el trabajo en equipo, permitiendo la creación de ramas para nuevas funcionalidades, y ayudando en la resolución de conflictos de código.

2) Explique la diferencia entre git y github.

Git es una herramienta de control de versiones que se utiliza localmente en la máquina del desarrollador para gestionar y mantener un historial de cambios en el código. GitHub, por otro lado, es una plataforma basada en la nube que proporciona alojamiento para repositorios Git. Ofrece funcionalidades adicionales como colaboración en equipo, seguimiento de issues, pull requests, y una interfaz web para gestionar y revisar el código.

3) ¿Qué es un branch?

Un branch (rama) en Git es una línea de desarrollo independiente que permite a los desarrolladores trabajar en nuevas funcionalidades, corregir errores o experimentar sin afectar la rama principal o main. Cada branch puede evolucionar independientemente hasta que se decida fusionar los cambios con la rama principal u otras ramas.

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

Un Pull Request (PR) es una solicitud para fusionar cambios de una rama a otra, generalmente de una rama de características o de corrección de errores a la rama principal (main). Permite a los desarrolladores revisar y discutir los cambios propuestos antes de integrarlos al código base.

5) ¿Qué es un commit?

Un commit en Git es una acción de guardar los cambios en el código en un punto específico del tiempo. Cada commit tiene un identificador único y contiene un mensaje descriptivo que explica el propósito de los cambios realizados.

6) Describa lo que sucede al ejecutar la siguiente operación: “git rebase main”.

La operación git rebase main aplica los commits de la rama actual sobre la cima de la rama main, cambiando el punto de inicio de la rama actual. Esencialmente, reescribe el historial de commits para que parezca que se desarrollaron a partir de la punta de main. Esto puede resultar en un historial de commits más lineal y limpio.

7) Explique que es un “merge conflict” y como lo resolvería.

Un merge conflicto ocurre cuando Git no puede resolver automáticamente las diferencias entre dos ramas que se están fusionando debido a cambios concurrentes en las mismas líneas de un archivo o en archivos muy cercanos. Para resolver un conflicto de fusión, el desarrollador debe revisar los conflictos manualmente, editar los archivos conflictivos para combinar los cambios de manera adecuada, y luego marcar los conflictos como resueltos mediante `git add` y realizar un nuevo commit.

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

Una prueba unitaria (Unit Test) es un tipo de prueba automatizada que verifica el comportamiento de una unidad específica de código, generalmente una función o método. Su objetivo es asegurar que la unidad de código funcione correctamente de manera aislada. En Python, unittest es un marco de pruebas integrado que proporciona una manera estructurada de escribir y ejecutar pruebas unitarias.

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

En pytest, assert se utiliza para comprobar que una condición es verdadera durante la ejecución de una prueba. Si la condición es falsa, pytest registra el fallo de la prueba e informa al desarrollador, ayudando a identificar problemas en el código.

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

E501: "Line too long" - Indica que una línea de código excede el límite de longitud permitido, generalmente 79 caracteres.

E302: "Expected 2 blank lines, found 1" - Indica que se esperaban dos líneas en blanco antes de una función o clase, pero se encontró una sola.

F401: "Module imported but unused" - Señala que se ha importado un módulo o paquete que no se utiliza en el código, lo cual es innecesario y puede eliminarse para limpiar el código.