

## Preguntas Teóricas

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

Git es una herramienta de control de versiones, la cual permite rastrear y gestionar cambios en el código, facilitando que varios desarrolladores pueden colaborar en un mismo proyecto y en diferentes objetivos de manera paralela. Esta herramienta mantiene un historial detallado de todas las modificaciones realizadas, lo que permite revertir o revisar cambios de forma sencilla. Además, los integrantes del equipo pueden trabajar sin conexión y obtener un respaldo, ya que, cada miembro puede tener una copia completa del repositorio en la computadora.

2. Explique la diferencia entre git y github

Git es un sistema de control de versiones la cual permite que los desarrolladores puedan rastrear y gestionar cambios en el código fuente desde su respectiva computadora. Esta herramienta mantiene un historial detallado de todas las modificaciones realizadas, lo que permite revertir o revisar cambios de forma sencilla. Asimismo, los desarrolladores pueden trabajar sin conexión y obtener un respaldo, ya que, cada miembro posee una copia completa del repositorio en su computadora.

Por otro lado, GitHub es una plataforma basada en la web la cual utiliza Git como su sistema de control de versiones. Dicha plataforma facilita la colaboración entre desarrolladores a través de un entorno para alojar repositorios Git en la red, permitiendo copiar, corregir y compartir código con otros desarrolladores.

3. ¿Qué es un branch?

Un branch es una ramificación de un proyecto “madre” que permite a los desarrolladores trabajar en agregar un nuevo recurso, corregir errores o realizar experimentos sin afectar la rama principal. Esto mantiene la estabilidad del proyecto, ya que permite revisar y probar cambios antes de fusionarlos con la rama principal. También facilita que los desarrolladores trabajen de manera paralela sin interrumpir el trabajo de los demás.

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

Un Pull Request es una función que solicita al equipo de trabajo la revisión y aprobación de los cambios realizados en una rama en específica antes de fusionarse con la rama principal.

5. ¿Qué es un commit?

Un commit es una operación que guarda el estado actual de los archivos del proyecto. Captura todos los cambios realizados desde el último commit incluyendo los archivos que se eliminan. Además, cada commit incluye un mensaje que describe los cambios realizados y posee un identificador único que permite rastrear y revertir modificaciones en el historial del proyecto.

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

Al ejecutar el código Git rebase main, lo que sucede es que, en la rama actual, Git toma los cambios de esta rama y los aplica sobre la punta de la rama “main”, creando un nuevo historial de commits en la parte superior del “main”. Esto hace que la cadena de confirmación sea lineal y mucho más fácil de leer.

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

Un merge conflict, es un error que sucede cuando múltiples cambios en una misma línea de un archivo al mismo momento, por lo que Git no puede realizar la combinación automática de los cambios.

Para resolver el “merge conflict”, se debe de resolver manualmente, primero se debe de identificar y abrir los archivos que están en conflicto en un editor de texto. En este se verán secciones de código delimitados donde se indican las diferencias entre las ramas en conflicto. Posteriormente se debe de decidir cuál de los cambios se mantiene o bien combinarlos manualmente o bien crear una nueva solución que integre lo mejor de ambas versiones.

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

Es una técnica de prueba que verifica el funcionamiento correcto de las unidades individuales de código, que suelen ser las funciones o métodos de una aplicación. El objetivo de las pruebas unitarias es asegurarse de que cada componente pequeño del software funcione de manera correcta y conforme a lo esperado, de manera aislada del resto del sistema.

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

El “assert” es una declaración clave utilizada para verificar que una condición específica sea verdadera. Si la condición es falsa, “pytest” considera que la prueba ha fallado. La utilidad principal de “assert” en “pytest” es su simplicidad y la claridad que aporta a la escritura de pruebas, facilitando la verificación del comportamiento esperado del código.

10. Mencione y explique tres errores de formato detectables con Flake8

- **E501: Línea demasiado larga:** Flake8 detecta cuando una línea de código excede la longitud máxima recomendada. Según PEP 8, la longitud máxima de una línea debe ser de 79 caracteres.
- **W292: Falta de línea en blanco al final del código:** Flake8 indica que falta una nueva línea al final de un archivo de código. Según las convenciones de estilo de PEP 8, todos los archivos de código deben terminar con una sola línea en blanco. Esto es importante para algunas herramientas y sistemas que procesan el código, y también contribuye a la consistencia y legibilidad del código.
- **F821: Nombre de variable indefinida:** Flake8 indica el uso de una variable o nombre no definido (undefined name). Este error ocurre cuando se intenta utilizar

una variable, función o clase que no ha sido previamente declarada o importada en el ámbito actual del código