

Preguntas teóricas

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

La principal utilidad de Git es administrar y organizar los cambios que se realizan en un proyecto de software a lo largo del tiempo. Se trata de un sistema de control de versiones que permite llevar un registro detallado de cada modificación hecha en el código, identificando quién la realizó y en qué momento. Esto facilita trabajar en equipo, ya que múltiples desarrolladores pueden contribuir al mismo proyecto sin perder información ni sobrescribir el trabajo de otros. Además, Git permite regresar a versiones anteriores del proyecto si ocurre algún error, lo que brinda seguridad y estabilidad durante el desarrollo.

2. ¿Qué es un commit?

Un commit es el acto de guardar oficialmente un conjunto de cambios dentro del historial del proyecto. Cuando se realiza un commit, se registra el estado actual de los archivos modificados junto con un mensaje descriptivo que explica qué se cambió. Este proceso crea un punto de referencia dentro de la línea de desarrollo, permitiendo consultar versiones anteriores y entender cómo ha evolucionado el código con el paso del tiempo.

3. ¿Qué es un branch?

Un branch es una rama o línea alternativa de desarrollo dentro de un repositorio. Permite trabajar en nuevas funcionalidades, mejoras o correcciones sin afectar directamente la versión principal del proyecto. Al crear un branch, se genera una copia del estado actual del código sobre la cual se pueden hacer modificaciones independientes. Posteriormente, si los cambios funcionan correctamente, pueden integrarse nuevamente a la rama principal.

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

En GitHub, un Pull Request es una solicitud formal para integrar los cambios realizados en una rama hacia otra, generalmente hacia la rama principal. Más que un simple mecanismo de unión de código, el Pull Request funciona como un espacio de revisión colaborativa, donde otros miembros del equipo pueden analizar los cambios, sugerir mejoras y aprobar la integración. Este proceso ayuda a mantener estándares de calidad y a reducir errores antes de incorporar nuevas modificaciones al proyecto principal.

5. ¿Si un usuario quiere “Actualizar su repositorio contra el Branch master” que debe de hacer?

Si un usuario desea actualizar su repositorio contra el branch master, debe sincronizar su copia local con la versión más reciente disponible en el repositorio remoto. Esto implica ubicarse en la rama correspondiente y descargar los cambios que hayan sido realizados por otros colaboradores. Al hacer esto, el desarrollador asegura que está trabajando sobre la versión más actualizada del proyecto y evita conflictos futuros.

6. ¿Bajo qué condiciones una herramienta como Git necesita apoyo de un humano para saber cómo integrar cambios locales con cambios remotos?

Git requiere intervención humana cuando detecta conflictos que no puede resolver automáticamente. Esto ocurre cuando dos versiones distintas modifican la misma sección de un archivo o cuando existen cambios incompatibles entre la copia local y la remota. En estos casos, el sistema no puede determinar cuál versión debe mantenerse, por lo que el desarrollador debe revisar manualmente las diferencias y decidir cómo combinar correctamente los cambios.

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

Una prueba unitaria es un procedimiento automatizado diseñado para verificar que una parte específica del código, normalmente una función o método, funcione correctamente de manera aislada. Su objetivo es comprobar que, ante ciertos datos de entrada, la unidad produce el resultado esperado. Las pruebas unitarias ayudan a detectar errores de lógica en etapas tempranas del desarrollo y contribuyen a mantener la confiabilidad del software conforme se realizan nuevas modificaciones.

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

En el contexto de pytest, un assert es una instrucción que permite comprobar si una condición se cumple durante la ejecución de una prueba. Se utiliza para comparar el resultado obtenido por una función con el resultado esperado. Si la condición evaluada es verdadera, la prueba se considera exitosa; si es falsa, el sistema indica un fallo. De esta manera, el assert es el mecanismo fundamental para validar el comportamiento correcto del código durante las pruebas automatizadas.

9. ¿Qué es Flake 8?

Flake8 es una herramienta de análisis estático utilizada en Python para evaluar la calidad del código sin necesidad de ejecutarlo. Su función principal es verificar que el programa cumpla con las normas de estilo establecidas, detectar posibles errores sintácticos y señalar inconsistencias en la estructura del código. Al utilizar Flake8, se mejora la legibilidad, el orden y la mantenibilidad del proyecto, promoviendo buenas prácticas de programación.

10. Comente sobre la utilidad de la aleatorización en pruebas de código.

La aleatorización en pruebas consiste en utilizar datos generados de manera aleatoria para evaluar el comportamiento de un programa bajo múltiples escenarios. Esto permite ampliar la cobertura de prueba y descubrir errores que podrían no aparecer cuando se utilizan únicamente casos predefinidos. Al exponer el código a diferentes combinaciones de entrada, se incrementa su robustez y se garantiza un funcionamiento más confiable frente a situaciones inesperadas.