

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

Git es un sistema de control de versiones, este permite a los desarrolladores gestionar el código fuente de una aplicación o página de manera eficiente. Su principal utilidad es permitir el seguimiento de cambios en el código a lo largo del tiempo, facilitando la colaboración entre múltiples desarrolladores y garantizando que los desarrollos se puedan realizar de forma paralela sin interferir con el trabajo de otros. Por otro lado, Git proporciona herramientas para crear ramas o "branches" en inglés, realizar cambios, fusionar (hacer "merge" en inglés) y revertir estos cambios de manera segura. Esto es especialmente útil para proyectos en equipo, ya que se pueden gestionar múltiples versiones de un proyecto sin riesgo de sobrescribir el trabajo de otros.

2. ¿Qué es un commit?

Un commit en Git es una operación que guarda los cambios realizados en el repositorio. Es decir, cada commit captura una parte del código en un momento específico, junto con un mensaje descriptivo que explica qué cambios se han realizado. Es fundamental para el control de versiones, ya que permite rastrear el historial del proyecto, revisar qué cambios fueron realizados y quién los hizo. Además, cada commit tiene un identificador único (hash) que facilita la recuperación o reversión a ese punto específico del proyecto.

3. ¿Qué es un branch?

Un branch en Git es una versión paralela de un proyecto, permite que los desarrolladores trabajen en nuevas funcionalidades, corregir errores, o experimentar sin afectar el código principal (main o master). Los branch permiten que varias personas trabajen de forma independiente en el mismo proyecto, con el objetivo o fin de fusionar ("merge") todo al "main"; ofreciendo flexibilidad para desarrollar diferentes características, features o corregir errores de forma aislada, sin afectar el main o master.

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

Un Pull Request o "PR" por sus siglas en inglés, es una solicitud para que los cambios realizados en un branch de un repositorio sean revidados e incorporados en un main. En GitHub, cuando un desarrollador termina de trabajar en un branch, crea un PR para que otros colaboradores del proyecto puedan revisar los cambios antes de fusionarlos en el main. Portanto, los PR son clave para el control de calidad en el desarrollo colaborativo, puesto que permiten que el código sea revisado, probado y aprobado antes de ser integrado oficialmente.

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

Para actualizar un repositorio local con los últimos cambios de branch master, el usuario debe ejecutar el comando "git pull origin master". Este comando hace 2 cosas: primero, trae los últimos

cambios del repositorio remoto y en segundo lugar, fusiona los cambios del branch actual con el repositorio local. Si hay conflictos entre branches, el sistema pedirá al usuario que los resuelva antes de completar la fusión.

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

Git necesita de apoyo humano cuando se quiere realizar un merge y existen conflictos por diferencias en el código ya que este no puede decidir cuál de las versiones conservar, debido a esto la persona debe solucionar el conflicto manualmente eliminando la parte de código que no interesa conservar.

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

Una prueba unitaria o Unittest es una prueba que se le realiza de forma automatizada a una parte muy pequeña del código, generalmente una función o método con el objetivo de verificar que funcione como se espera, estas pruebas se suelen realizar cada vez que se le realiza algún cambio al código con el objetivo de encontrar rápidamente los fallos en caso de que los haya.

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

Un assert permite comprobar si la condición dentro del mismo es verdadera, en caso de no ser así envía la excepción "AssertionError", es muy utilizada en pruebas unitarias para verificar el funcionamiento de las funciones.

9. ¿Qué es Flake 8?

Es una herramienta de revisión de código Python que sirve para detectar errores de sintaxis, violaciones de PEP8 y problemas de complejidad, dicha revisión se realiza sin ejecutar el código.

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

La aleatorización en pruebas de código permite encontrar errores inesperados y por ende mejorar la robustez del código, además simula condiciones reales a las que puede ser sometido el código, aunque las pruebas sean aleatorias es importante tener un control sobre las mismas, para en caso de encontrar errores poder reproducirlas y verificar que los errores hayan sido arreglados.