

Parte 1: Teoría

1. ¿Diferencie la herramienta Git de Github?

Git: es un sistema de control de versiones de código en forma distribuida, esto quiere decir que se almacena un historial de las versiones que se han modificado en un código fuente que esta siendo trabajado por varios programadores. Las copias de los trabajos de cada colaborador se pueden utilizar como repositorios, así se general respaldos de las versiones de código de todos los que están trabajando.

GitHub: es una plataforma por la cual varios colaboradores pueden compartir proyectos y desarrollarlos mutuamente. Por medio de esta se pueden planificar y organizar el desarrollo de un proyecto, trabajar en un solo proyecto en conjunto con otros desarrolladores, llevar control de las versiones trabajadas y también almacenar versiones anteriores como respaldo en caso de ser necesario accesarlas de nuevo. Este medio también permite conectarse con otros desarrolladores que trabajan en temas similares, por lo que es una buena herramienta para fines investigativos.

2. ¿Qué es un branch?

Cada vez que se realiza un commit o confirmación en Git, se genera una copia completa del código nuevo, a partir de la versión mas reciente. Esta estructura de guardado permite que existan varias versiones a partir del mismo archivo, lo cual genera arboles de archivos con una línea principal llamada master, donde se van realizando los commits. De todo esto nacen los branches, estos son punteros hacia cada versión del archivo trabajado.

3. ¿Qué es un commit?

Un commit es cuando se modifica un programa o código dentro de GitHub y se sube el nuevo archivo (o la nueva versión), cambiando el código anterior. Estos commits o cambios generalmente vienen acompañados de una breve descripción de que es lo que se esta cambiando, una vez que este commit se sube y actualiza el código original, se genera una versión nueva y se almacena la versión anterior como respaldo.

4. ¿Qué es la operación cherry-pick?

Tiene como función de elegir un commit específico de otra rama y aplicarlo a otra. Se le da ese nombre por su metodología de elegir solo commits específicos y no fusionar en total ambar versiones del código.

5. ¿Qué hace el comando git stash?

Tiene la función de guardar los cambios efectuados al código por el usuario sin aplicar cambios al repositorio de modo que no se deben realizar commits. Puede ser utilizado cuando se deben

guardar los cambios pero el programador no esta listo para realizar el commit en el repositorio del proyecto, entonces se almacena en un stash y puede ser retomado mas adelante.

6. ¿Compare las operaciones git fetch y git pull

Git fetch: es para acceder al codigo original y verificar si existen cambios respecto a la copia que se tiene de este.

Git pull: es la misma función que la anterior con la diferencia de que esta copia los cambios a nuestro repositorio.

7. Asumiendo que usted está en un Branch llamado “secundario” y su Branch principal se llama “master” ¿Qué resultado espera de hacer git rebase master? ¿Qué resultado espera de hacer git rebase origin/master?

Con esta acción se pretende ir al archivo “ancestro” en común de ambas ramas, de aquí se obtienen las deferencias entre el “ancestro” y el secundario y esas diferencias se guardan en archivos temporales. Así se hace un reset de “secundario” al mismo commit de la rama “master”, para finalmente aplicar las diferencias guardadas en los archivos temporales. Se espera unificar dos modificaciones diferentes del mismo archivo, es por esto que se usa una rama “secundaria” y la “master” para aplicar cambios al archivo anterior a estas dos ramas.

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

Es un método para probar que un módulo o un conjunto de módulos de un código funcionen correctamente. Estas se implementan a la par junto con el desarrollo de un proyecto y se ejecutan cuando se le han hecho modificaciones al proyecto para garantizar su funcionamiento. Es posible que el código de la prueba unitaria tenga errores, por eso es importante que el código de la prueba con la de su respectivo código estén separadas para que puedan correr independientemente.

9. Bajo el contexto de pytest. ¿Qué es un “assert”?

Los asserts en pytest son pruebas que permiten comparar los resultados de una función con los resultados esperados y devuelven un valor booleano de verdadero o falso. El assert detiene el código del método de prueba si el valor es falso, siguiendo con el siguiente método de prueba.

10. ¿Qué es Flake 8?

Es una herramienta hecha para Python encargada de analizar la manera en la que esta escrito el código con el fin de corregir y mejorar la sintaxis. Gracias a esta es posible reducir la longitud y claridad del código sin reducir su funcionalidad.