

Preguntas teóricas

1. ¿Diferencie la herramienta Git de GitHub?

Git es una herramienta de código abierto que se instala localmente con la finalidad de que los desarrolladores la utilicen para gestionar el código fuente de un proyecto en desarrollo, mientras que GitHub es un servicio en línea al que los desarrolladores que utilizan Git pueden conectarse y cargar o descargar recursos. En resumen, Git es el sistema de control de versiones y GitHub es un servicio de alojamiento para los repositorios de Git.

2. ¿Qué es un Branch?

Una rama de desarrollo (“Git Branch”) representa una línea de desarrollo independiente, la cual también puede verse como una bifurcación del estado del código que crea un nuevo camino para la evolución de este. Las nuevas confirmaciones se registran en el historial de la rama actual, lo que da como resultado esa bifurcación en el historial del proyecto. Un Git Branch puede pensarse como una forma de solicitar un nuevo directorio de trabajo, área de preparación e historial de un proyecto.

3. ¿Qué es un commit?

Un “*Commit*” en el contexto de Git hace referencia a la acción de confirmar un conjunto de cambios provisionales de forma permanente.

4. ¿Qué es la operación Cherry-pick?

El comando “*git cherry-pick*” se utiliza cuando se desea elegir una confirmación de una rama y aplicarla a otra, por lo que “*git cherry-pick*” puede ser útil para deshacer cambios. Por ejemplo, supongamos que una confirmación se aplica accidentalmente en la rama equivocada. Se puede cambiar a la rama correcta y ejecutar “*git cherry-pick*” en la confirmación para aplicarla a donde pertenece.

5. ¿Qué hace el comando git stash?

El comando “*git stash*” lo que realiza es un tipo de “congelamiento” del estado en el que se encuentra el proyecto en un momento determinado, con todos los cambios a los cuales no

se les ha realizado un “commit”, y los guarda en un espacio provisional para tener la posibilidad de recuperarlos más adelante.

6. Compare las operaciones git fetch y git pull.

El comando “*git fetch*” se utiliza para que el git local recupere todos los ficheros de un repositorio remoto (en este caso GitHub). El comando “*git fetch*” solo recupera la información del repositorio y la ubica en una rama oculta de un repositorio local, pero no la fusiona con el repositorio que se está trabajando localmente. Por otro lado, “*git pull*” realiza una acción similar a “*git fetch*” con la diferencia que este caso el comando “*git pull*” sí integra los datos al repositorio que se está utilizando localmente.

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

Si se está ubicado en el “secundario” y se ejecuta un “git rebase master” lo que se realiza es mover toda la información del Branch “secundario” a la punta del branch principal “master”, incorporando de manera efectiva todas las actualizaciones de “master”. Luego, si se parte del supuesto que el repositorio en el que se está trabajando es un clonado de uno que se encuentra en GitHub, al ejecutar un “git rebase master/origin” lo que se estaría realizando es primero actualizando la información del master con lo que se encuentre en el repositorio remoto y luego coloca el branch secundario en la punta del branch master.

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

En desarrollo de software una prueba unitaria consiste en la comprobación del correcto funcionamiento de una unidad de código, es decir un único proceso, función o clase. Esto permite una revisión modular de un proyecto, y facilitar la detección y corrección de errores.

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

Un “assert” es el enunciado que se desea comprobar en una prueba. En el contexto de pytest un “assert” sería que la aplicación funcione de la manera en la que se esperaba o presente el error esperado. En otras palabras, es un predicado el cual puede ser verdadero o falso, sobre la forma en la que se comporta el programa.

10. ¿Qué es Flake8?

Flake8 es una herramienta de “*Linting*”, es decir, es una herramienta de corrección y detección de errores en un programa o código. Sirve principalmente para estandarizar y mejorar la eficiencia a la hora de programar.