

1. Git es un sistema de control de versiones de código en forma distribuida, lo que quiere decir que almacena un historial de las versiones que se han modificado en un código fuente que está siendo trabajado por varios programadores. Tiene como característica que la copia de trabajo de cada colaborador puede utilizarse como repositorio, de esta forma generando respaldos de las versiones del código que se está trabajando por todos.
2. GitHub es la plataforma o “red social” por la que los desarrolladores comparten proyectos y se colabora mutuamente. A través de ella se puede planificar y organizar el desarrollo de un proyecto, trabajar en un solo proyecto en conjunto con otros desarrolladores, llevar control de las versiones trabajadas e incluso almacenar versiones anteriores como respaldo en caso de cualquier problema.

Por medio de esta plataforma se puede también conectar con otros desarrolladores que trabajan en temas similares o de interés para el usuario, por lo que también funciona de manera perfecta como herramienta investigativa.

3. Cada vez que se realiza un commit o confirmación en Git, se genera una copia completa del código nuevo, a partir de la versión aceptada más reciente. Esta estructura de guardado, permite que existan varias versiones a partir del mismo archivo, lo cual genera árboles de archivos con una línea principal llamada master, donde se van realizando los commits. Un Branch es un puntero que apunta hacia cada versión del archivo trabajado. (Página oficial Git).
4. 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 qué es lo que se está cambiando y, 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.
5. El comando “cherry pick” tiene como función la de elegir un commit específico de otra rama y aplicarlo a otra. Se denomina Cherry pick por su metodología de elegir solo commits específicos y no fusionar en total ambas versiones del código.
6. Por medio del comando “git checkout” se puede desplazar entre las ramas creadas por el comando “git branch”, con el fin de trabajar en una versión y parte específica del código y modificar este sin temor a dañar alguna otra parte ya funcional del código.
7. 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 está listo para realizar el commit en el repositorio del proyecto, entonces se almacena en un stash y puede ser retomado más adelante.

8. Antes de comparar los comandos “git fetch” y “git pull”, cabe mencionar que estos términos se aplican cuando se está trabajando en una copia o clon del código original. Dicho esto, “git fetch” se utiliza para acceder al código original y verificar si existen cambios respecto a la copia que se tiene de este. Por otra parte, “git pull” cumple la misma función pero con la diferencia de que esté copia o trae dichos cambios a nuestro repositorio.
9. El comando git reset ~HEAD sirve para deshacer cambios en el código, volviendo a una versión previa en el repositorio. HEAD hace referencia al puntero que indica el historial reciente de los cambios, es decir, donde se realizaron las últimas confirmaciones. Este código entonces, devuelve el puntero a un estado previo, restableciendo de esta forma el código a su estado previo.
10. Pytest es un software especializado para detectar errores de código en Python, el cual identifica los errores de manera detallada indicando de manera específica qué está mal y cómo solucionarlo.
11. 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.
12. Flake8 es una herramienta hecha para Python encargada de analizar la manera en la que está escrito el código con el fin de “limpiarlo” y mejorar la sintaxis del mismo. Si bien una herramienta encargada de encontrar errores (como Pytest) es altamente importante, Flake8 es igual de importante pues esta puede reducir la longitud y claridad del código sin reducir su funcionalidad.