

1. Git como herramienta de desarrollo de código puede rastrear los cambios realizados en el código con el tiempo, permitiendo a múltiples desarrolladores colaborar en un proyecto creando un sistema de control de versiones. Si se tiene una versión funcional, esta se puede dejar como un checkpoint para poder regresar a ella si se quiebra el código a futuro.
2. Git es un sistema de control de versiones que permite a los desarrolladores rastrear los cambios realizados en el código, mientras que GitHub es una plataforma web que proporciona una interfaz gráfica para Git, permitiendo a los desarrolladores almacenar, administrar y compartir los repositorios de código. En otras palabras, Git es la herramienta, y GitHub es un servicio que proporciona una forma conveniente de utilizar Git.
3. Una rama en Git es una línea de desarrollo separada que se desvía del código base principal ("main" o "master"). Las ramas permiten a los desarrolladores trabajar en nuevas características o correcciones de errores de forma independiente del código base principal, sin afectar el código principal.
4. En el contexto de GitHub, un Pull Request es una forma de proponer cambios a un repositorio. Cuando un desarrollador hace cambios en una rama, puede crear una Solicitud de Extracción para pedir a los mantenedores del repositorio que revisen y fusionen sus cambios en el código base principal.
5. Un commit en Git es una instantánea del código en un momento específico, que incluye una descripción de los cambios realizados. Cuando un desarrollador hace cambios en el código, crea un commit para guardar esos cambios, junto con un mensaje que describe qué se cambió. Si a futuro algo ocurre, o se necesitan eliminar ciertos aspectos del código, revertir a un commit viejo es una manera muy eficiente y segura de hacerlo.
6. Cuando se ejecuta el comando "git rebase main", Git toma los commits de la rama actual y los reaplica encima de la rama principal. Esto puede ser útil para mantener la rama actualizada con los últimos cambios en la rama principal, pero también puede reescribir la historia de commits, lo que puede ser problemático si otros ya han basado su trabajo en la rama.
7. Un conflicto de fusión ocurre cuando dos o más desarrolladores hacen cambios en el mismo código, y Git no puede fusionar automáticamente esos cambios. Para resolver un conflicto de fusión, los desarrolladores necesitan editar manualmente el código para reconciliar las diferencias, y luego comprometer los cambios resueltos.
8. Una Prueba Unitaria o Unittest en desarrollo de software es una prueba pequeña y aislada que verifica la corrección de un componente específico del código, usualmente una función o método. Las pruebas unitarias ayudan a asegurar que los componentes individuales del código funcionen como se espera, lo que facilita la detección de errores y bugs temprano en el proceso de desarrollo.
9. En el contexto de pytest, una instrucción "assert" se utiliza para verificar que una condición específica sea verdadera. Si la condición es falsa, la prueba fallará, indicando que hay un problema con el código. Las instrucciones assert son esenciales en las pruebas unitarias, ya que permiten a los desarrolladores especificar el comportamiento esperado del código y detectar cualquier desviación de ese comportamiento.
10. Flake8 es una herramienta que verifica el código Python para detectar errores de formato y violaciones de la guía de estilo. Tres errores comunes detectables por Flake8 son:
 - * E501: línea demasiado larga (más de 79 caracteres)

- * E225: falta de espacio en blanco alrededor del operador
- * F401: módulo o paquete importado no utilizado