

Preguntas Teóricas

1) ¿Diferencie la herramienta Git de GitHub?

Git es un Sistema de Control de Versiones Distribuido (DVCS) que permite guardar las diferentes versiones de un archivo para que cualquier versión se pueda recuperar cuando se desee. GitHub, es una plataforma donde los usuarios pueden almacenar los repositorios Git en la web y facilita compartir proyectos con otras personas. La diferencia es entonces que GitHub es una plataforma que emplea del sistema Git en la web y es una interfaz amigable para manejar el sistema [6].

2) ¿Qué es un Branch?

Los Branch o ramas, en español, es análogo a la rama de un árbol. Los Branch son espacios para donde un usuario pueda ir desarrollando y codificando una nueva característica o modificación del software sin borrar los archivos originales del proyecto. La rama principal o por defecto en Git es la rama máster (similar al tronco del árbol) [3].

3) ¿Como se crea un nuevo Branch?

En GitHub para crear un nuevo Branch se debe elegir el repositorio por trabajar, se da clic en Rama Actual y luego se hace clic en New Branch (nueva rama). Además, se puede crear un Branch desde un Commit ubicando el historial, dándole click derecho y luego click en Create Branch From Commit [1].

4) ¿Qué es un commit?

Los commits capturan los cambios a medida que se trabaja. En Git, cada commit es similar a un punto de guardado. Antes de la ejecución del commit, el comando add se utiliza para iniciar los cambios guardados en un commit [2].

5) ¿Qué es la operación “git stash”?

El comando git stash almacena temporalmente (guarda en un stash) los cambios que se hayan efectuado en el código en el que se trabaja. Permite entonces retener algún cambio sin necesariamente aplicarse al código [4].

6) Explique de forma gráfica: ¿Que sucede en mi Branch local cuando hago “git rebase master”?

“rebase” se utiliza para agregar los cambios de una rama a otra [8].

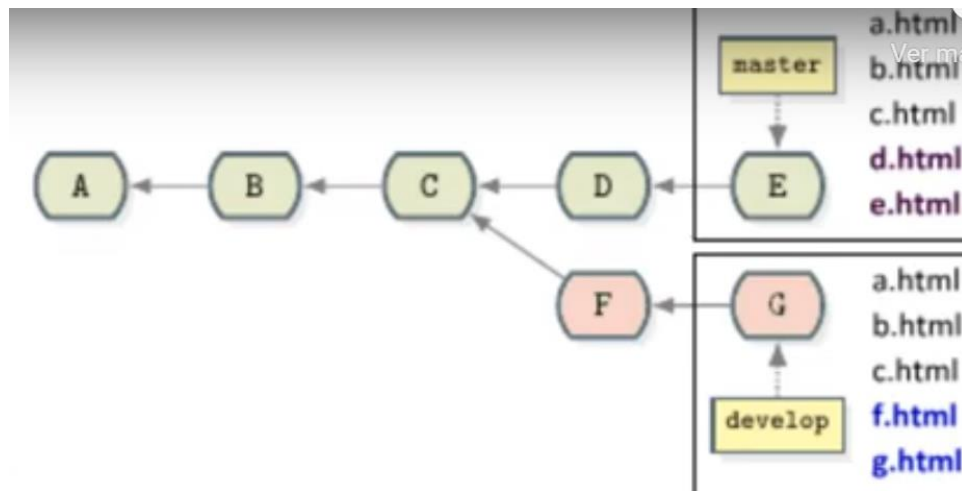


Figura 1. Para este ejemplo, se tienen el branch master y el branch bifurcado de desarrollo. Con la función *rebase* se busca enlazar los commits F y G al branch master [8].

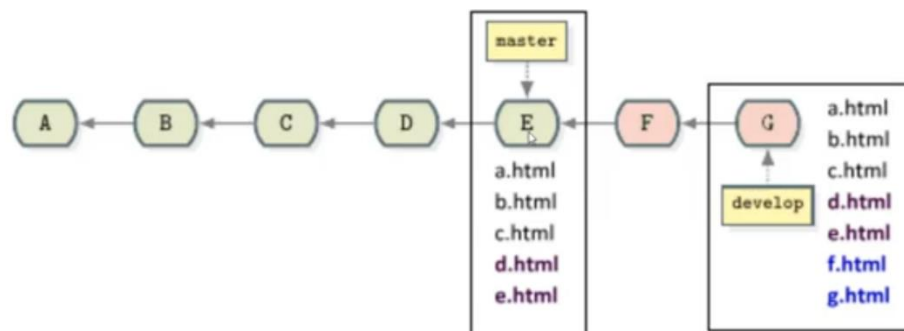


Figura 2. Después de usar el comando “git rebase master develop” los commits F y G quedan enlazados al branch master. Inicialmente develop no podía acceder a los commits D y E, pero luego de ejecutar el *rebase* develop forma parte del branch master por lo que ya puede ver los commits D y E [8].

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

Una prueba unitaria es un método para probar una unidad estructural de código. Estas pruebas tienen las siguientes características:

- Prueba solamente pequeñas cantidades de código.
- No afecta a otras pruebas unitarias.
- No debería requerir intervención manual.

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

Un assert permite expresar una condición que ha de ser cierta siempre, caso contrario se interrumpirá el programa. El comando assert permite encontrar errores según la condición que se le den este. Generalmente es utilizado para confirmar condiciones que siempre van a ser ciertas, de no ser así, el assert lo considera como un error de programación [7].

9) ¿Qué es Flake 8?

Flake8 es un conjunto de herramientas para comprobar la estructura del código en base al estilo de codificación (PEP8), revisar errores de programación (como "biblioteca importada pero no utilizada" y "nombre indefinido") y para comprobar la complejidad ciclomática. Esta última es una métrica del software que proporciona una medición cuantitativa de la complejidad lógica de un programa independientemente del lenguaje.

10) Explique la diferencia entre un “log de error” y un “valor de error de retorno”

Log de error es un método para que tome un registro de los errores ocurridos en el programa; esto a menudo incluye errores atrapados que puede que no necesite o no le interese mostrar al usuario final. Por su parte, valor de error de retorno, al detectar un error en una función esta retorna que hubo un fallo en la ejecución de esta [5].

Referencias

- [1] “Managing branches.” <https://docs.github.com/es/desktop/contributing-and-collaborating-using-github-desktop/making-changes-in-a-branch/managing-branches>.
- [2] “Git commit.” <https://www.atlassian.com/git/tutorials/saving-changes/git-commit>.
- [3] H. Rajora, “Branch in Git,” Nov. 16, 2021. <https://www.toolsqa.com/git/branch-in-git/>.
- [4] “git stash.” <https://www.atlassian.com/es/git/tutorials/saving-changes/git-stash>.
- [5] C. Fenner, “Allow returning a value with an error #2647,” Jun. 09, 2019. <https://github.com/ziglang/zig/issues/2647>.
- [6] E. Castellanos, “Git vs GitHub – ¿Qué es el Control de Versiones y Cómo Funciona?,” 2021. <https://www.freecodecamp.org/espanol/news/git-vs-github-what-is-version-control-and-how-does-it-work/>.
- [7] S. Campbell, “PyTest Tutorial: What is, How to Install, Framework, Assertions,” Jan. 01, 2022. <https://www.guru99.com/pytest-tutorial.html>
- [8] C. McKenzie, Git rebase to master example. YouTube.com, 2020.
- [9] V. Freitas, “How to Use Flake8” Aug. 5, 2016. <https://flake8.pycqa.org/>