

Microprocesadores y Microcontroladores

Tarea 1

José Gutiérrez – Daniel Mena

- (1) La principal utilidad de git como herramienta de desarrollo de código es que permite un flujo de desarrollo dinámico basado en ramas donde cada característica puede ser desarrollada, probada y compartida sin necesidad de afectar el código base/de referencia por cada desarrollador. Además, la habilidad de fusionar estas ramas de manera segura en la rama principal/base.
- (2) Git es la herramienta que realiza el control de versiones, por defecto se utiliza desde una terminal. Github es un servicio web que aloja los repositorios de Git y provee una interfaz gráfica para interactuar.
- (3) Un “Branch” o una rama es una versión copia del repositorio principal/base.
- (4) Para Github un “pull request” es una petición para fusionar dos ramas, al realizarla permite revisar y discutir los cambios que le realizarían a la rama objetivo según los contenidos de la rama de origen.
- (5) Un “commit” o compromiso se refiere a crear una imagen o “snapshot” del repositorio el cual incluye “metadata”. (Guardar)
- (6) “git rebase main” ejecuta los siguientes pasos:
 - a. Primero encuentra el ancestro común entre la rama principal y la rama secundaria en la que se está trabajando.
 - b. Luego guarda temporalmente los cambios realizados en la rama secundaria desde ese ancestro común.
 - c. Cambia el repositorio en trabajo al actual principal.
 - d. Aplica esos cambios previamente guardados.

Esto resulta en que la rama secundaria se actualice con la rama principal. De haber conflictos el proceso se detiene para poder resolverlos.

- (7) “Merge conflict” o conflictos de fusión se refiere que en el proceso de combinar dos ramas existen cambios aplicados en la misma línea de archivo en ambas ramas o incluso que se hacen cambios a un archivo en una rama mientras que en la otra el archivo se elimina.

Para resolver los conflictos de fusión se deben editar manualmente y seleccionar cuales cambios se desean mantener.

- (8) Las pruebas unitarias o “Unittest” son códigos que interactúan con unidades pequeñas de código mediante sus entradas y salidas para comprobar su funcionamiento. Efectivamente son códigos que llaman a funciones para verificar que sus salidas sean las correctas.
- (9) “assert” permite indicar en el software de prueba cual es la respuesta correcta o esperada, en caso de que la función no devuelva el valor esperado se genera un reporte de donde se encontró aserción falsa.
- (10) Errores:
 - a. F401: un módulo se importa, pero no se está usando.
 - b. E101: el sangrado tiene “espacios” y “tabs” combinados.
 - c. E303: existen muchas (3) líneas vacías seguidas.