

Instituto Tecnológico de Costa Rica Ing. Rodolfo Piedra Camacho

Área Académica de Ingeniería Mecatrónica II Semestre 2020

MT-7003 Microprocesadores y microcontroladores Valor: 5%

Emanuel Cerdas Ramirez

Jorge Sanabria Guadamuz

Tarea 1

GitHub, Pytest y Flake 8

A continuación, se presentan 2 asignaciones para entender y aplicar herramientas de desarrollo de proyectos de programación.

Preguntas Teóricas (24 pts, 2pts c/u)

1) Qué es Git?

R/ Un herramienta que funciona como control de versiones de código, pero de una manera distribuida. Esto significa que cada usuario tiene su propio repositorio, donde los repositorios pueden intercambiar y mezclar revisiones entre ellos mismos. Un repositorio es el lugar en el que se almacenan los datos actualizados y la historia de cambios, un servidor.[1]

2) ¿Qué es Github?

R/ Es un servicio para hacer hosting de repositorios de software que se administra con Git. Es algo más particular: un sitio web que usa Git para ofrecer a la comunidad de desarrolladores repositorios de software. GitHub mantiene una copia de repositorios en la nube, que además pueden estar disponibles para otros desarrolladores. [1]

3) ¿Qué es un branch?

R/ Las ramas son espacios o entornos independientes para que un Desarrollador sea Back-end, Front-end, Tester, etc. pueda usar y así trabajar sobre un mismo proyecto sin alterar o borrar el conjunto de archivos originales del proyecto. [2]

4) ¿Qué es un commit?

R/ Un "commit" es la acción de guardar o subir tus archivos a tu repositorio remoto (una actualización de tus cambios) también puede hacerse al local (depende donde hayas creado tu repositorio). [3]

5) ¿Qué es la operación cherry-pick?

R/ Trabajando con git es habitual usar diferentes ramas en un mismo proyecto, por ejemplo, podemos estar usando una rama master con el código que tenemos en producción y una rama development donde trabajamos en la siguiente iteración. De esta forma puede suceder que, en un momento dado, queramos aplicar algunos de los cambios que hemos hecho en otra rama sin tener que hacer un merge completo, es decir sin tener que guardar todos los cambios sino solamente el que queremos. [4]

Para esto se utiliza el comando cherry-pick. Este comando permite coger uno o varios commits de otra rama sin tener que hacer un merge completo.

6) ¿Qué hace el comando git checkout?

R/ El comando git checkout te permite desplazarte entre las ramas creadas por git branch. Al ejecutar la operación de cambio de rama, se actualizan los archivos en el directorio de trabajo para reflejar la versión almacenada en esa rama y se indica a Git que registre todas las confirmaciones nuevas en esa rama.[5]

7) ¿Qué hace el comando git stash?

R/ El comando git stash permite guardar cambios realizados en el código y retomar a la última versión o commit almacenado, estos cambios también pueden recuperarse en cierto momento de ser requerido, o en caso de que se hayan deshecho por error del usuario.[6]

8) Compare las operaciones git fetch y git pull

R/ Un git fetch permite descargar commits, archivos y referencias de un repositorio remoto al repositorio local, sin embargo este no fusiona los contenidos de los repositorios, lo cual sirve para revisar los commits antes de integrarlo al repositorio local [7], en cambio un git pull es la combinación de dos comandos git fetch y git merge, lo cual genera que se extraiga y descargue contenido de un repositorio remoto y esto genera que se actualice el repositorio local al instante.[8]

Por lo que es más seguro ejecutar un git fetch, ya que este no modifica el repositorio local, para revisar el contenido del repositorio remoto antes de realizar la función de contenidos.

9) ¿Qué hace el comando git reset ~HEAD?

R/ Dicho comando nos permite deshacer cambios tanto en la preparación de un archivo, como en un archivo modificado. Para el primer caso sería como por ejemplo cuando se utiliza el comando git add ., en donde se van a preparar para subir los documentos que se tengan a disposición, sin embargo si se requiere que los archivos sean subidos con distintos git commit, entonces con git reset HEAD <file>, se puede sacar un documento de la preparación para ser subido.

Para el segundo caso se debe ejecutar primero el comando `git checkout`, y seguidamente el comando `git reset HEAD <file>`, sin embargo en este caso, este comando es un poco riesgoso usar, ya que elimina cualquier cambio que se haya efectuado en este archivo.[9]

10) ¿Qué es Pytest?

R/ Pytest es un software gratuito y de código abierto, el cual permite realizar pruebas de funcionalidad a aplicaciones y bibliotecas, entre sus principales características posee información detallada sobre las pruebas fallidas, puede ejecutar pruebas unitarias o conjuntos de pruebas además posee una rica arquitectura de complementos.[10]

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

R/ Un `assert`, se utiliza para verificar las expectativas y los valores en las pruebas de Python, con el fin de afirmar que su función devuelve el valor esperado. Si esta aserción falla, verá el valor de retorno de la llamada a la función. Por otra parte se permite especificar un mensaje de aserción, esto se logra colocando una coma al final del comando y el comentario entre comillas dobles.[11]

12) ¿Qué es Flake 8?

R/ Un Flake 8, es una librería de Python, la cual cuenta con una serie de herramientas y envuelve a PyFlakes, pycodestyle y el script McCabe de Ned Batchelder. Una de sus funciones es verificar el código con el estilo de codificación PEP8, errores de programación y además revisa la complejidad ciclomática, este último término se refiere a la cantidad de rutas independientes a través del código fuente, a este término también se le conoce como complejidad de McCabe.[12]

Bibliografía

- [1] I. Alcázar, «desarrolloweb.com,» 3 Junio 2014. [En línea]. Available:
] <https://desarrolloweb.com/articulos/introduccion-git-github.html> . [Último acceso: 14 Setiembre 2020].
- [2] Nube Colectiva, «NC,» 21 Febrero 2019. [En línea]. Available:
] <https://blog.nubecollectiva.com/que-son-las-ramas-branches-en-git-como-utilizarlas-y-otros-detalles/> . [Último acceso: 14 Setiembre 2020].
- [3] D. Pantalones, «Códigofacilito,» 19 Agosto 2015. [En línea]. Available:
] <https://codigofacilito.com/articulos/commits-administrar-tu-repositorio> . [Último acceso: 14 Setiembre 2020].
- [4] Runroom, «Runroom,» 15 Junio 2017. [En línea]. Available:
] <https://www.runroom.com/realworld/seleccionando-commits-cherry-pick#:~:text=Git%20Cherry%20Pick&text=Este%20comando%20permite%20coger%20uno,que%20hacer%20un%20merge%20completo..> [Último acceso: 14 Setiembre 2020].
- [5] ATlassian, «ATlassian Bitbucket,» [En línea]. Available:
] <https://www.atlassian.com/es/git/tutorials/using-branches/git-checkout#:~:text=En%20t%C3%A9rminos%20de%20Git%2C%20%22checkout,%3A%20archivos%2C%20confirmaciones%20y%20ramas.> . [Último acceso: 14 Setiembre 2020].
- [6] J. Ochoa, «Styde,» 5 Mayo 2015. [En línea]. Available:
] <https://styde.net/deshacer-y-recuperar-cambios-con-git/#:~:text=El%20comando%20git%20stash%20permite,deshecho%20por%20error%20del%20usuario..> [Último acceso: 14 Setiembre 2020].
- [7] ATlassian, «ATlassian Bitbucket,» [En línea]. Available:
] <https://www.atlassian.com/es/git/tutorials/syncing/git-fetch>. [Último acceso: 14 Setiembre 2020].
- [8] ATlassian, «ATlassian Bitbucket,» [En línea]. Available:
] <https://www.atlassian.com/es/git/tutorials/syncing/git-pull>. [Último acceso: 14 Setiembre 2020].
- [9] git, «git,» [En línea]. Available: <https://git-scm.com/book/es/v2/Fundamentos-de-Git-Deshacer-Cosas> . [Último acceso: 14 Setiembre 2020].

[1] pytest, «pytest,» [En línea]. Available: <https://docs.pytest.org/en/stable/>. [Último acceso: 14 Setiembre 2020].

[1] pytest, «pytest,» [En línea]. Available: <https://docs.pytest.org/en/3.0.1/assert.html#asserting-with-the-assert-statement>. [Último acceso: 14 Setiembre 2020].

[1] V. Freitas, «Simpleisbetterthancomplex,» 5 Agosto 2016. [En línea]. Available: <https://simpleisbetterthancomplex.com/packages/2016/08/05/flake8.html>. [Último acceso: 14 Setiembre 2020].