

I Parte: Preguntas teóricas (Respuestas)

1. Antes de diferenciar Git de Github, es adecuado explicar qué es cada una de estas herramientas. “Git es un software de control de versiones (VCS) para desarrolladores” [1]. Entiéndase control de versiones como el proceso de guardar diferentes archivos (versiones de un archivo) a lo largo de la elaboración de un proyecto. Esto tiene la enorme ventaja de que permite devolverse a versiones anteriores del proyecto y poder revertir o modificar código según sea necesario. El software también tiene otra ventaja, y es que varios colaboradores pueden trabajar en un mismo proyecto; sin embargo, tiene el inconveniente de que, al estar instalado en cada computadora personal, otros colaboradores no pueden ver en tiempo real cuando se estén realizando cambios en el proyecto.

“Github es una plataforma que puede mantener repositorios de código en almacenamiento en la nube para que varios desarrolladores puedan trabajar en un solo proyecto y ver las ediciones de cada uno en tiempo real” [1]. Básicamente es una herramienta que facilita la colaboración con Git. Esta versión contiene muchas funciones entre ellas herramientas de moderación para asignarle a cada individuo o grupo de trabajo una asignación específica. Además, hay tres acciones principales que se pueden ejecutar en lo que respecta a la interacción con el código de otros usuarios de la plataforma y son conocidas como: Bifurcación, Pull y Fusión.

- Bifurcación: Es la simple operación de copiar código de otra persona para modificarlo.
- Pull: Es una acción que se ejecuta luego de haber modificado el código de alguien más para compartir con esa persona los cambios que uno le ha realizado, esto se hace a través de una “solicitud pull”
- Fusión: Cada propietario puede añadir cambios a sus proyectos por medio de una fusión, y su vez otorgar el crédito a las personas que los han sugerido.

La diferencia principal entre Git y Github radica en que Git está más orientado al trabajo individual de los diferentes desarrolladores, y por otro lado Github tiene una estructura que se orienta más al trabajo en equipo de forma simultánea gracias a que se almacena toda la información en la nube, y al estar allí, muchos otros desarrolladores tienen acceso y por ello existe la colaboración entre los mismos.

2. “Un branch o rama en Git es un apuntador móvil que se irá desplazando o apuntado a cada nueva confirmación de cambios que realicemos” [2]. Cada vez que se haga una nueva confirmación de cambios, se crea una nueva rama y el nuevo apuntador se puede mover libremente.
3. Un commit es un comando que se utiliza para realizar una confirmación de los cambios. “Cuando se crea una confirmación con el comando git commit, Git realiza sumas de control de cada subdirectorío, y las guarda como objetos árbol en el repositorio Git” [2].
4. “Cherry-pick es un comando que permite seleccionar confirmaciones arbitrarias por medio de una referencia y agregarlas a la rama HEAD de trabajo actual” [3]. Este

comando es útil para deshacer cambios, al permitir deshacer confirmaciones hechas en ramas incorrectas y pasar la confirmación a la rama donde debería estar.

5. El comando git stash se utiliza cuando se necesita registrar el estado actual del directorio de trabajo y el índice; pero se quiere volver a un directorio de trabajo que está limpio. El comando lo que hace es guardar las modificaciones locales y revierte el directorio de trabajo para que coincida con la confirmación HEAD.
6. “El comando git pull se encarga de fusionar todos los cambios que se han hecho en el repositorio local con el directorio de trabajo local. Por otro lado, el comando git fetch permite a un usuario buscar objetos de un repositorio remoto que actualmente no se encuentran en el repositorio de trabajo local” [5].
7. Todos los cambios hechos en el secundario se incorporan al master, lo que genera un branch nuevo por cada cambio que se tenga que reescribir del branch original. El git rebase origin/master tiene la finalidad de evitar conflictos con la fusión de confirmaciones en el master remoto, cuando dos o más compañeros de proyecto están realizando cambios en el mismo momento. Entonces si yo estoy trabajando en el secundario (mi rama) y otro compañero realiza confirmaciones en el master que me gustaría tener a la vista, el comando git rebase/master va a permitir fusionar mi rama (secundario) con la del compañero para que se apliquen las confirmaciones que él ha realizado localmente, y de esta manera no se afecte el historial de confirmaciones del master remoto.
8. “Las pruebas unitarias o unittest son una manera de comprobar que un trozo de código funcione correctamente” [7]. Estas pruebas tienen que poseer ciertas características para asegurar que sean de calidad. Las mismas tienen que ser: Repetibles o reutilizables, independientes, automatizables, profesionales y completas. Al menos deben poseer estas características antes mencionadas para así lograr identificar los posibles errores de código presentes y hacer más eficiente el proceso de programación.
9. “Assert” en pytest es una declaración utilizada para verificar las expectativas y los valores correctos de una función en una prueba. Esta declaración se encarga de retornar un valor correcto de la función a prueba.
10. Flake8 es una herramienta de python que permite tener un código mucho más limpio, ya que se encarga de revisar si existen variables que están declaradas pero que no se están usando; y además permite revisar que se cumpla con la guía PEP8, esta última siendo una serie de reglas sobre la correcta escritura de código en Python.

Referencias bibliográficas

1. KINSTA, "Git vs Github: ¿Cuál es la diferencia y cómo empezar?", *kinsta.com*. Dic 29, 2020. Consultado en: <https://kinsta.com/es/base-de-conocimiento/git-vs-github/>. [Se accedió en Feb 22, 2021].
2. Git, "Ramificaciones en Git - ¿Qué es una rama?", *git-scm.com*. Consultado en: <https://git-scm.com/book/es/v2/Ramificaciones-en-Git-%C2%BFQu%C3%A9-es-una-rama%3F>. [Se accedió en Feb 22, 2021].
3. ATlassian, "Git Cherry Peak", *atlassian.com*. Consultado en: <https://www.atlassian.com/git/tutorials/cherry-pick>. [Se accedió en Feb 22, 2021].
4. Git, "Git - Stash", *git-scm.com*. Consultado en: <https://git-scm.com/docs/git-stash>. [Se accedió en Feb 22, 2021].
5. Guztavo B., "Comandos de GIT Básicos – Guía Completa", *hostinger.es*. Ene 21, 2021. Consultado en: <https://www.hostinger.es/tutoriales/comandos-de-git>. [Se accedió en Feb 22, 2021].
6. ATlassian, "Merging vs Rebasing", *atlassian.com*. Consultado en: <https://www.atlassian.com/git/tutorials/merging-vs-rebasing>. [Se accedió en Feb 24, 2021].
7. YeePLY, "¿Qué son las pruebas unitarias y cómo llevar una a cabo?", *yeePLY.com*. Consultado en: <https://www.yeePLY.com/blog/que-son-pruebas-unitarias/>. [Se accedió en Feb 24, 2021].
8. Pytest, "Installation and Getting Started", *docs.pytest.org*. Consultado en: <https://docs.pytest.org/en/latest/getting-started.html>. [Se accedió en Feb 24, 2021].
9. G. Navarro, "Tips para desarrollo en python", *codeandoando.com*. Agos 9, 2016. Consultado en: <https://codeandoando.com/tips-para-desarrollo-en-python/>. [Se accedió en Feb 24, 2021].