



INSTITUTO TECNOLÓGICO DE COSTA RICA

Microprocesadores y microcontroladores gr 1

Elaborado por:

Jorge Cerdas Solórzano

Joseph Muñoz Cascante

Carnés:

2019015533

2019064730

II SEMESTRE, 2021

Preguntas Teóricas (16 pts, 2pts c/u)

1) ¿Diferencie la herramienta Git de Github?

Git es una herramienta de software de control de versiones de códigos de forma distribuida. Dicha herramienta es de software libre y con ella se puede mantener un historial completo de versiones. Esta herramienta utiliza ramas o branches para acceder cada versión de manera organizada y rápida. En cuanto a la funcionalidad de las ramas, las mismas están destinadas a provocar proyectos divergentes de un proyecto principal, para hacer experimentos o para probar nuevas funcionalidades.

Las ramas pueden tener una línea de progreso diferente de la rama principal donde está el core de nuestro desarrollo. En algún momento podemos llegar a probar algunas de esas mejoras o cambios en el código y hacer una fusión a nuestro proyecto principal, ya que todo esto lo maneja Git de una forma muy eficiente.

Github es una compañía sin fines de lucro y sitio web de esta que utiliza la herramienta de Git para ayudar a desarrolladores a almacenar y administrar su código, al igual que llevar un registro y control de cualquier cambio sobre este código. Dicha compañía sin fines de lucro que ofrece un servicio de hosting de repositorios almacenados en la nube. Esencialmente, hace que sea más fácil para individuos y equipos usar Git como la versión de control y colaboración.

2) ¿Qué es un branch?

En Git una branch o rama es un entorno independiente para que un programador sea capaz de programar y probar diferentes versiones de un mismo código sin borrar o sobrescribir los archivos originales del proyecto. Esto facilita el desarrollo de un programa.

3) ¿Qué es un commit?

Commit es un comando que guarda todos los cambios en un entorno con una breve descripción para recordar cambios hechos al commit y poder alterar esa versión. Si varios commits editan partes distintas del proyecto no se sobrescribirán.

4) ¿Qué es la operación cherry-pick?

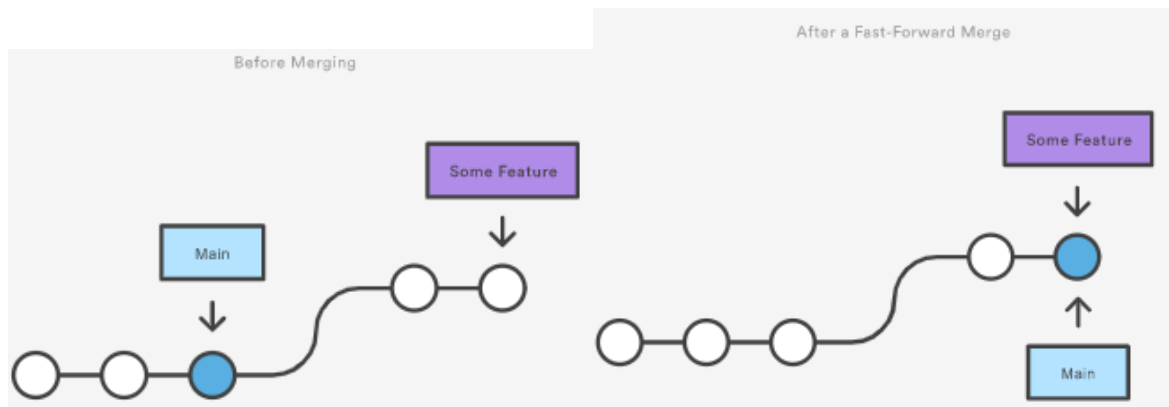
Cherry-pick es un comando en Git que permite a los commits a ser escogidos por referencia y agregados al encabezado funcional actual. Cherry-picking es el acto de escoger un commit de una rama o Branch y aplicarlo a otra. Sirve para deshacer cambios.

5) Explique de forma gráfica como cambia el “master” de un repositorio cuando se hace merge de un Branch.

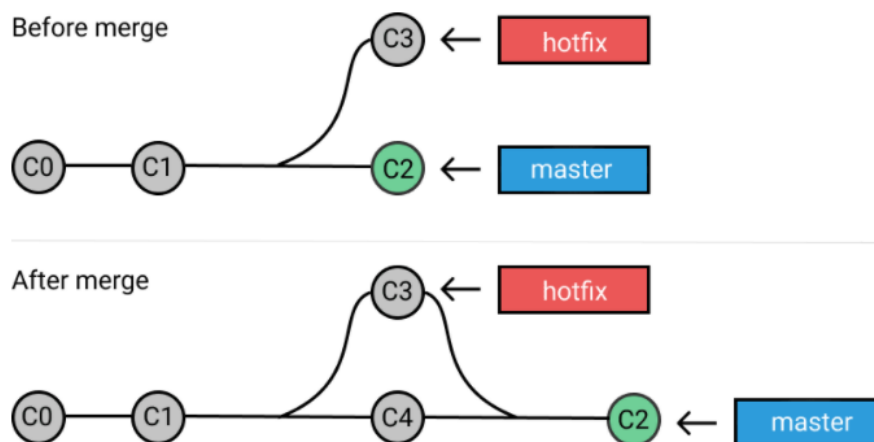
En algún momento se desea fusionar ramas para poder incorporar el trabajo realizado a la rama master, esto se conoce como “merge”. Git para procesar los merge usa un antecesor común y comprueba los cambios que se han introducido al proyecto desde ese momento, combinando el código de ambas ramas. La rama

master, se actualizará para reflejar la fusión, pero la rama de fusión no se verá afectada, por lo que si ya no se necesita se deberá borrar después.

Existe la fusión fast forward o avance rápido, que es una fusión lineal; como se ve en la siguiente imagen, la rama master no ha divergido, hay un proceso lineal de integración de los historiales desde el extremo de la rama actual hasta la rama de destino, ahora el main o master se encuentra en el mismo punto que la rama de fusión y tiene todas las confirmaciones.



Cuando las ramas han divergido, se realiza la fusión de 3 vías o 3-way merge; en la siguiente imagen se observa antes del merge que las dos ramas están en puntos paralelos, no hay un proceso lineal hacia la rama de destino. Las fusiones de 3 vías usan una confirmación específica para unir dos ramas. Esta fusión recibe su nombre del hecho de que Git utiliza tres confirmaciones para generar la confirmación de fusión: los dos extremos de la rama y su predecesor común. Como se observa, después del merge, el master es la unión de la rama de fusión y de la divergencia realizada y posee todas las confirmaciones de ambos.



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

Las pruebas unitarias son un método de pruebas de software, son pequeños tests que validan el comportamiento de un objeto y la lógica. Consisten en aislar una parte del código, se realizan escribiendo fragmentos de código que testeara unidades de código fuente, el objetivo es asegurar que cada unidad funciona como debería de forma independiente. Cuando el código está escrito las pruebas unitarias son las primeras que se realizan y las crea el propio desarrollador.

Algunos desarrolladores dicen que son una pérdida de tiempo y las evitan buscando ahorrar tiempo, sin embargo, estas pruebas detectan errores que no se contemplan en la fase de desarrollo. Además, demuestran que la lógica del código está en buen estado, mejora la calidad del código, aumenta la legibilidad y ayudan a los desarrolladores a entender el código base lo que facilita hacer cambios. Las pruebas Unitarias implementan los 3 pasos: Arrange, establecer los requisitos que debe cumplir el código y el resultado esperados. Act, ejecutar el test. Assert, comprobar si los resultados son los que se esperaban, si es así, se valida, si no, se corrige el error.

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

Pytest permite usar los assert en Python, son herramientas primarias para comunicar fallos al momento de testear el código, especialmente usadas para unittests, su uso permite realizar comprobaciones, si la expresión contenida dentro del mismo es falsa, se lanzará una excepción (AssertionError). En Pytest, se colocan con la prueba, si falla el assert, la prueba se detiene, el código restante en la prueba no se ejecuta y se continua con la siguiente prueba.

8) ¿Qué es Flake 8?

Flake8 es una librería de Python para ayudar a detectar si el código Python esta estandarizado, Flake8 tiene reglas de inspección flexibles, admite la integración de complementos adicionales y tiene una gran escalabilidad. Contiene las herramientas: PyFlakes, para verificar estáticamente errores lógicos en el código, PEP8, para comprobar estáticamente el estilo de codificación PEP8 y el script McCabe de Ned Batchelder, para el análisis estático de la complejidad del código. Además de las herramientas, Flake8 entrega una interfaz de desarrollo extendida.

Bibliografía:

Atlassian. (s. f.-a). *Git Cherry Pick | Atlassian Git Tutorial*. Recuperado 10 de agosto de 2021, de <https://www.atlassian.com/git/tutorials/cherry-pick>

Atlassian. (s. f.-b). *Git merge | Atlassian Git Tutorial*. Recuperado 10 de agosto de 2021, de <https://www.atlassian.com/es/git/tutorials/using-branches/git-merge>

Carrillo, J. (2021, 5 febrero). *El Comando Git Commit Explicado*.

freeCodeCamp.org. <https://www.freecodecamp.org/espanol/news/el-comando-git-commit-explicado/>

Colectiva, N. (2019, 28 mayo). *Que son las Ramas (Branches) en Git, como utilizarlas y otros detalles*. Blog Nube Colectiva.

<https://blog.nubecolectiva.com/que-son-las-ramas-branches-en-git-como-utilizarlas-y-otros-detalles/>

Desarrollo Web. (2021, 21 mayo). *Trabajar con ramas en Git: git branch*.

<https://desarrolloweb.com/articulos/trabajar-ramas-git.html>

Git - Procedimientos Básicos para Ramificar y Fusionar. (s. f.). Git SCM.

Recuperado 10 de agosto de 2021, de <https://git-scm.com/book/es/v2/Ramificaciones-en-Git-Procedimientos-Basicos-para-Ramificar-y-Fusionar>

Git Immersion. (s. f.). Gitimmersion. Recuperado 10 de agosto de 2021, de <https://gitimmersion.com>

Herramienta de inspección de código estático de Python Flake8 - programador clic. (s. f.). Programador Click. Recuperado 10 de agosto de 2021, de <https://programmerclick.com/article/40861407461/>

Patiño, R. E. (2021, 8 agosto). *Que son las pruebas unitarias e importancia de estas*. Anexsoft. <https://anexsoft.com/que-son-las-pruebas-unitarias-e-importancia-de-estas>

¿Qué es GitHub? *Una Guía para Principiantes sobre GitHub*. (2020, 8 octubre). Kinsta. <https://kinsta.com/es/base-de-conocimiento/que-es-github/>

Rubio, J. C. (2020, 30 julio). *Qué es GIT y para qué sirve*. OpenWebinars.net. <https://openwebinars.net/blog/que-es-git-y-para-que-sirve/>

Rungta, K. (2021, 24 junio). *PyTest Tutorial: What is, How to Install, Framework, Assertions*. Guru99. <https://www.guru99.com/pytest-tutorial.html>

Uso del assert(). (s. f.). El Libro De Python. Recuperado 10 de agosto de 2021, de <https://ellibrodepython.com/assert-python>

YeePLY. (2020, 15 diciembre). *¿Qué son las pruebas unitarias y cómo llevar una a cabo?* <https://www.yeeply.com/blog/que-son-pruebas-unitarias/>

Zoric, N. (2018, 30 abril). *How to Use Git Merge [the Correct Way]*. DEV Community. <https://dev.to/neshaz/how-to-use-git-merge-the-correctway-25pd>