



Instituto Tecnológico de Costa Rica

Curso: Microcontroladores y microprocesadores

Profesor: Rodolfo Piedra Camacho

Tarea 1

Estudiantes:

Xochilth Víctor Angulo 2019190056

Marco Rodríguez Morales 2019163031

Grupo: 01

I Semestre

2022

Fecha de Entrega: del 2022

1) ¿Diferencie la herramienta Git de Github?

Según [1], Git es un sistema de control de versiones distribuido (DVCS) de software gratuito y de código abierto diseñado para administrar todo el historial del código fuente. Puede mantener un historial de commits, puede revertir cambios y permite a los desarrolladores compartir código. Cada desarrollador debe tener Git instalado en su dispositivo local para colaborar.

Por otro lado, GitHub es un servicio de alojamiento basado en web para repositorios de Git. Ofrece todo el DVCS SCM de Git y tiene algunas características adicionales. Esto incluye funcionalidad de colaboración como gestión de proyectos, gestión de tickets de soporte y seguimiento de errores. Con GitHub, los desarrolladores pueden compartir sus repositorios, acceder a los repositorios de otros desarrolladores y almacenar copias remotas de los repositorios para que sirvan como copias de seguridad.

Otra diferencia en la comparación de Git vs GitHub es el entorno en el que operan. Git se instala localmente en un sistema, por lo que los desarrolladores pueden administrar su historial de código fuente utilizando sus máquinas locales como repositorios. Esto significa que no se requiere un servidor centralizado para usar Git, y tampoco se necesita acceso a Internet.

Mientras tanto, GitHub vive en la nube, por lo que se requiere acceso a Internet. También tiene un sistema de gestión de usuarios incorporado y una GUI (Interfaz Gráfica de Usuario) fácil de usar. Además de su sitio web principal, GitHub presenta una versión de escritorio que se puede instalar en computadoras locales para ayudar a sincronizar el código. Cabe señalar que Git se puede usar sin GitHub, pero GitHub no se puede usar sin Git.

2) ¿Qué es un branch?

Un branch, o rama, es una copia que se genera del código en el que originalmente se trabaja para realizar, de forma paralela, algún cambio que se necesite en el mismo. De esta forma, ambos códigos, a partir del momento en el que se crea el branch, presentan funcionamientos diferentes según el programador. [2]

3) ¿Cómo se crea un nuevo Branch?

1. Dar clic en la pestaña "Code" del repositorio.
2. Dar clic en el menú desplegable en la parte superior de la lista de archivos que dice "main"
3. Escribir un nombre de rama en el cuadro de texto. Por ejemplo, el nombre "readme-edits"
4. Dar clic en Create branch: readme-edits from main. [3]

4) ¿Qué es un commit?

Este comando se usa para guardar un cambio que se le hizo a la programación en un nuevo proyecto, basado en el proyecto original. [4]

5) ¿Qué es la operación “git stash”?

Este almacena temporalmente (o guarda en un stash) los cambios que se hayan efectuado en el código en el que se está trabajando para que se pueda trabajar en otra cosa y, más tarde, regresar y volver a aplicar los cambios. [5]

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



La rama en la que se estaba trabajando pasa al frente de la rama local.

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

Unittest es un tipo de prueba de software donde se prueban unidades o componentes individuales de un software. El propósito es validar que cada unidad del código de software funcione como se espera. Las pruebas unitarias se realizan durante el desarrollo (fase de codificación) de una aplicación por parte de los desarrolladores. Las pruebas unitarias aíslan una sección de código y verifican su corrección. Una unidad puede ser una función, método, procedimiento, módulo u objeto individual. [6]

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

Un assert es una instrucción utilizada para verificar que la respuesta de “x” función es la esperada. [7]

9) ¿Qué es Flake 8?

Flake8 es una biblioteca de Python que envuelve PyFlakes, pycodestyle y el script McCabe de Ned Batchelder. Es un excelente conjunto de herramientas para verificar su base de código contra el estilo de codificación (PEP8), errores de programación (como "biblioteca importada, pero sin usar" y "Nombre indefinido") y para verificar la complejidad ciclomática. Cabe destacar que la complejidad ciclomática es una métrica de software para medir la cantidad de rutas independientes a través del código fuente. En términos generales, cuanto mayor sea el número de ifs dentro de una función, mayor será el número de caminos que tendrá, por lo tanto, una mayor complejidad ciclomática. Por supuesto, hay otras operaciones de flujo de control que afectan el cálculo de la complejidad ciclomática. [8]

Adicionalmente, Flake8 posee algunas características [9]:

- Los archivos que contienen esta línea: `# flake8: noqa` se omiten.
- Las líneas que contienen un comentario `# noqa` al final no emitirán advertencias.
- Se pueden ignorar errores específicos en una línea con `# noqa: <error>`, por ejemplo, `# noqa: E234`. Se pueden dar múltiples códigos, separados por comas. El token `noqa` no distingue entre mayúsculas y minúsculas, los dos puntos antes de la lista de códigos son obligatorios; de lo contrario, se ignora la parte posterior a `noqa`.

Extensible a través de los puntos de entrada `flake8.extension` y `flake8.formatting`

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

Un valor de error de retorno es un número que se devuelve al momento de que se presente algún error en un programa, este número está relacionado a un tipo de error en específico. Por otro lado, un log de error es un registro donde se encuentran todos los errores registrados en un programa. [10]

Referencias

- [1] Simplilearn. "Git vs GitHub: What are the Major Differences?" Simplilearn.com. <https://www.simplilearn.com/tutorials/git-tutorial/git-vs-github> (último acceso: el 15 de febrero de 2022).
- [2] «git --everything-is-local,» [En línea]. Available: <https://git-scm.com/book/es/v2/Ramificaciones-en-Git-%C2%BFQu%C3%A9-es-una-rama%3F>. [Último acceso: 12 febrero 2022].
- [3] GitHub. "Hello World - GitHub Docs". GitHub Docs. <https://docs.github.com/en/get-started/quickstart/hello-world> (accedido el 18 de febrero de 2022).
- [4] J. Carrillo, «freeCodeCamp,» [En línea]. Available: <https://www.freecodecamp.org/espanol/news/el-comando-git-commit-explicado/>. [Último acceso: 12 febrero 2022].
- [5] Bitbucket. "git stash: Cómo guardar los cambios | Atlassian Git Tutorial". Atlassian. <https://www.atlassian.com/es/git/tutorials/saving-changes/git-stash#:~:text=El%20comando%20git%20stash%20almacena,aplicar%20los%20cambios%20más%20tarde>. (accedido el 15 de febrero de 2022).
- [6] Guru99. "Unit Testing Tutorial: What is, Types, Tools & Test EXAMPLE". Guru99. <https://www.guru99.com/unit-testing-guide.html> (accedido el 15 de febrero de 2022).
- [7] «pytest,» 2015. [En línea]. Available: <https://docs.pytest.org/en/latest/how-to/usage.html>. [Último acceso: 17 febrero 2020].
- [8] V. Freitas. "How to Use Flake8". Simple is Better Than Complex. <https://simpleisbetterthancomplex.com/packages/2016/08/05/flake8.html#:~:text=Flake8%20is%20a%20Python%20library,and%20to%20check%20cyclomatic%20complexity>. (accedido el 15 de febrero de 2022).
- [9] Python Software Foundation. "flake8". PyPI. <https://pypi.org/project/flake8/> (accedido el 15 de febrero de 2022).
- [10] «Hector Docs,» 25 setiembre 2018. [En línea]. Available: <https://docs.hektorprofe.net/python/programacion-de-funciones/retorno-de-valores/>. [Último acceso: 16 febrero 2022].