



**Curso de Microprocesadores y Microcontroladores – MT7003**

**GitHub, Pytest y Flake 8**

**Estudiantes:** Marco Cambronero Villalobos

Cristhofer Loría Calderón

**Link del repositorio:** <https://github.com/Koldeneve/TareasCambroneroLoria.git>

**Preguntas Teóricas (20 pts, 2pts c/u)****1) ¿Explique la principal utilidad de git como herramienta de desarrollo de código?**

Git es un sistema de control de versiones locales, por lo que la principal utilidad como herramienta de desarrollo de código es que permite rastrear los cambios en el código fuente a lo largo del tiempo a la vez que facilita el trabajo en equipo al permitir que múltiples desarrolladores trabajen en diferentes partes del código de forma paralela. También es posible revertir a versiones anteriores del código si se introduce un error, por lo tanto facilita la integración de cambios en el proyecto principal de manera controlada.

**2) Explique la diferencia entre git y github**

Git: Es un software de control de versiones locales.

GitHub: Es una plataforma de alojamiento de repositorios Git en la nube. Permite compartir código, colaborar con otros desarrolladores y utilizar diversas herramientas de gestión de proyectos.

**3) ¿Qué es un branch?**

Un branch (rama) en Git es una línea independiente de desarrollo. Al crear un branch, se genera una bifurcación del historial del proyecto, permitiendo trabajar en nuevas características, corregir errores o realizar testing sin afectar la línea principal del proyecto.

**4) En el contexto de github. ¿Qué es un Pull Request?**

Un Pull Request es una propuesta para combinar un conjunto de cambios de una rama con otra. En un Pull Request, los colaboradores pueden revisar y analizar el conjunto propuesto de cambios antes de integrar los cambios en el código base principal. Los Pull Request muestran las diferencias o la diferencias entre el contenido de la rama de origen y el contenido de la rama de destino.

**5) ¿Qué es un commit?**

Un commit es una instantánea de los cambios realizados en un repositorio en un momento dado el cual tiene un mensaje asociado que describe los cambios realizados.

**6) Describa lo que sucede al ejecutar la siguiente operación: “git rebase main”.**

Al ejecutar git rebase main, se reescriben los commits de la rama actual sobre los commits de la rama main.

**7) Explique que es un “merge conflict” y como lo resolvería.**

Un merge conflict ocurre cuando se intentan fusionar dos ramas que han cambiado los mismos archivos de manera diferente. Para resolverlo, se debe editar manualmente el archivo conflictivo para elegir las partes correctas de cada versión.

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

Una prueba unitaria es una pequeña pieza de código que verifica el comportamiento de una unidad individual de código por ejemplo, una función o una clase. Ayudan a garantizar la calidad del código y a detectar errores temprano en el desarrollo.

**9) Bajo el contexto de pytest. ¿Cuál es la utilidad de un “assert”?**

En pytest, assert se utiliza para verificar que una condición sea verdadera. Si la condición es falsa, se lanza una excepción y la prueba falla. Por ejemplo, si en determinado fragmento de código se debe obtener siempre un valor “True”, se define un assert que lo corrobore, de no cumplirse, el pytest indica que este assert falló, siendo que se esperaba un True pero en realidad hay un False.

**10) Mencione y explique tres errores de formato detectables con Flake8**

E902: Indentation is not a multiple of four: La indentación no es un múltiplo de cuatro espacios, lo que afecta la legibilidad del código.

W291: Blank line at end of file: Falta una línea en blanco al final del archivo, lo que es una convención de estilo común.

E701: Multiple statements on a single line (colon): Hay múltiples instrucciones en una sola línea, lo que dificulta la lectura y el mantenimiento del código.

**Referencias**

[1] Git. <https://git-scm.com/>

[2] GitHub Docs. "Acerca de Git". <https://docs.github.com/es/get-started/using-git/about-git>

[3] Nube Colectiva. "Que son las ramas (branches) en Git, como utilizarlas y otros detalles". [https://blog.nubecolectiva.com/que-son-las-ramas-branches-en-git-como-utilizarlas-y-otros-detalles/#google\\_vignette](https://blog.nubecolectiva.com/que-son-las-ramas-branches-en-git-como-utilizarlas-y-otros-detalles/#google_vignette).

[4] GitHub Docs. "Acerca de las solicitudes de incorporación de cambios – Documentación de GitHub". <https://docs.github.com/es/pull-requests/collaborating-with-pull-requests/proposing-changesto-your-work-with-pull-requests/about-pull-requests>.

[5] KeepCoding Bootcamps. "¿Qué es un Commit en Git?". <https://keepcoding.io/blog/que-es-un-commit-en-git/>

[6] Atlassian. "Git rebase". <https://www.atlassian.com/git/tutorials/rewriting-history/git-rebase>

[7] Atlassian. "Git merge conflicts". <https://www.atlassian.com/git/tutorials/using-branches/merge-conflicts>

[8] A. Feregrino. "Probando nuestro código con pytest". <https://old.tacosdedatos.com/pruebas-unitarias-pytest>.

[9] R. Ramkumar. "PyTest : Assertions and Test Discovery Python". <https://medium.com/@ramanish1992/pytest-assertions-and-test-discovery-python-24b4bcb468eb>

[10] Flake8 Rules. "The Big Ol' List of Rules". <https://www.flake8rules.com/>