

**Instituto Tecnológico de Costa Rica**  
**Área Académica de Ingeniería Mecatrónica**  
**MT-7003 Microprocesadores y microcontroladores**

# Solución Tarea 1: GitHub, Pytest y Flake 8

Emily Flores Rojas  
Esteban Vargas Monge

16 de agosto de 2025

Profesor: Ing. Rodolfo Piedra Camacho  
II Semestre 2025

---

## Preguntas Teóricas (Solución)

1. ¿Explique que es git y su relación con github?

**Git** es un sistema de control de versiones distribuido (DVCS) de código abierto. Su función principal es rastrear los cambios en los archivos de un proyecto, permitiendo la colaboración y el mantenimiento de un historial completo de versiones [1].

**GitHub** es una plataforma de alojamiento basada en la web para repositorios de Git. Provee una ubicación remota y centralizada para los repositorios, además añade herramientas colaborativas como gestión de issues, pull requests y automatización [2]. En resumen, Git es la herramienta, y GitHub es un servicio popular que la hospeda y potencia.

2. ¿Qué es un branch? ¿Qué es un fork?

Un **branch (rama)** es una línea de desarrollo independiente dentro de un repositorio de Git. Permite trabajar en nuevas características o corregir errores sin afectar la rama principal [3].

Un **fork (bifurcación)** es una copia personal de un repositorio completo que reside en la cuenta de otro usuario. Se utiliza para proponer cambios a proyectos en los que no se tienen permisos de escritura directa, a través de Pull Requests [4].

3. En el contexto de github. ¿Qué es un Pull Request?

Un **Pull Request (PR)** es un mecanismo para proponer cambios a un repositorio en GitHub. Es una solicitud para que el propietario de un repositorio revise e incorpore los commits de una rama o un fork a la base de código principal [5].

4. ¿Qué es un commit?

Un **commit** es una “instantánea” de los cambios guardados en el repositorio en un momento específico. Es la unidad atómica de trabajo en Git. Cada commit tiene un identificador único (hash SHA-1) y un mensaje descriptivo [1].

5. Explique que es un “merge conflict” o “rebase conflict”.

Un **conflicto de merge o rebase** ocurre cuando Git no puede fusionar automáticamente los cambios de dos ramas porque ambas han modificado las mismas líneas

en el mismo archivo. El desarrollador debe resolver manualmente estas discrepancias para completar la operación [6].

6. **¿Qué es una Prueba Unitaria o Unittest?**

Una **Prueba Unitaria (Unit Test)** es un método de prueba de software que consiste en verificar la funcionalidad de la unidad de código más pequeña y aislable de un programa. Su objetivo es validar que cada componente funcione correctamente de forma independiente.

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

En pytest, la palabra clave **‘assert’** es el mecanismo principal para realizar verificaciones. Un **‘assert’** evalúa una expresión; si es falsa, la prueba falla y pytest genera un informe de error detallado para facilitar la depuración. Esta es una característica estándar del lenguaje Python [7], que pytest utiliza de manera extensiva [8].

8. **¿Explique que son github-actions y su utilidad?**

**GitHub Actions** es una plataforma de automatización integrada en GitHub que permite crear flujos de trabajo (workflows) para la Integración Continua y Despliegue Continuo (CI/CD). Estos flujos automatizan tareas como la ejecución de pruebas, el linting y el despliegue de aplicaciones, activándose por eventos en el repositorio [9].

9. **¿Qué es Flake 8?**

**Flake8** es una herramienta de “linting” para Python que combina PyFlakes, pycodestyle (PEP 8) y McCabe. Su propósito es ayudar a escribir código limpio, legible y libre de errores comunes, mejorando su calidad y mantenibilidad [10].

10. **Explique la funcionalidad de parametrización de pytest.**

La **parametrización** en pytest, implementada con el decorador **‘@pytest.mark.parametrize’**, permite ejecutar una misma función de prueba múltiples veces con diferentes conjuntos de argumentos. Esto reduce la duplicación de código y permite probar una función con diversos casos de entrada de manera eficiente [11].

## Referencias

- [1] The Git Development Community. (2023) Git documentation. [Online]. Available: <https://git-scm.com/doc>
- [2] GitHub, Inc. (2023) Github docs. [Online]. Available: <https://docs.github.com/>
- [3] The Git Development Community. (2023) Git branching - branches in a nutshell. [Online]. Available: <https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell>
- [4] GitHub, Inc. (2023) About forks. [Online]. Available: <https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/working-with-forks/about-forks>
- [5] ——. (2023) About pull requests. [Online]. Available: <https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-pull-requests>
- [6] The Git Development Community. (2023) Basic merge conflicts. [Online]. Available: [https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging#\\_basic\\_merge\\_conflicts](https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging#_basic_merge_conflicts)
- [7] Python Software Foundation. (2023) The assert statement. [Online]. Available: [https://docs.python.org/3/reference/simple\\_stmts.html#the-assert-statement](https://docs.python.org/3/reference/simple_stmts.html#the-assert-statement)
- [8] The pytest development team. (2023) Assertions about expected exceptions. [Online]. Available: <https://docs.pytest.org/en/latest/how-to/assert.html>
- [9] GitHub, Inc. (2023) Understanding github actions. [Online]. Available: <https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions>
- [10] PyCQA. (2023) Flake8: Your tool for style guide enforcement. [Online]. Available: <https://flake8.pycqa.org/en/latest/>
- [11] The pytest development team. (2023) Parametrizing fixtures and test functions. [Online]. Available: <https://docs.pytest.org/en/latest/how-to/parametrize.html>