

## **Preguntas Teóricas (20 pts, 2pts c/u)**

### **1) ¿Explique la principal utilidad de git como herramienta de desarrollo de código?**

Con Git, los cambios se guardan de manera incremental y permite tener un registro de los cambios, retroceder a versiones anteriores y añadir funcionalidades. Además, permite llevar un seguimiento de los cambios realizados por otros desarrolladores en los archivos. [1]

### **2) ¿Qué es un branch?**

Una rama o branch es una versión del código del proyecto sobre el que estás trabajando. Estas ramas ayudan a mantener el orden en el control de versiones y manipular el código de forma segura. [2]

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

Un Pull Request es una función de GitHub que permite a tu equipo solicitar la revisión y aprobación de sus cambios antes de fusionarlos en la rama principal de desarrollo, denominada “master” o “main”. [3]

### **4) ¿Qué es un commit?**

Un commit en Git es como un punto de guardado en un videojuego. Básicamente, le dices a Git que quieres guardar el estado actual de tus archivos para poder volver a determinado punto si lo necesitas. [4]

### **5) Describa lo que sucede al ejecutar las siguientes operaciones: “git fetch” “git rebase origin/master”**

- git fetch: Descarga cambios del repositorio remoto, pero no los aplica.
- git rebase origin/master: Aplica los cambios remotos antes de los locales, reorganizando el historial.

### **6) Explique que es un “merge conflict” o “rebase conflict” en el contexto de tratar de hacer merge a un Pull Request o de completar una operación git rebase.**

Un merge conflict (conflicto de fusión) ocurre cuando Git no puede combinar automáticamente los cambios de dos ramas porque han modificado la misma parte del código de forma diferente.

Un rebase conflict ocurre cuando se intenta reordenar los commits con git rebase, pero hay diferencias en los mismos archivos que generan conflictos al aplicar los cambios en orden. [5]

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

Una prueba unitaria es una prueba automatizada que verifica que una pequeña unidad de código (como una función o método) funcione correctamente de manera aislada. [6]

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

En Pytest, assert se usa para comparar valores esperados y reales en pruebas unitarias. [7]

### 9) ¿Qué es Flake 8?

Flake8 es una herramienta que analiza código Python para verificar que sigue las normas de estilo del PEP 8 y detecta errores comunes. [8]

### 10) Explique la funcionalidad de parametrización de pytest.

La parametrización en Pytest permite ejecutar una misma prueba varias veces con diferentes datos de entrada sin escribir múltiples funciones de prueba.

La parametrización se puede aplicar a un test mediante el decorador `@pytest.mark.parametrize`. Este decorador permite especificar varios conjuntos de valores para un parámetro, y Pytest ejecutará el test una vez por cada conjunto de valores. [9]

### Referencias

[1] Kodigo, "Introducción a Git: conceptos básicos y su importancia en el desarrollo de software", *Kodigo*, 2023. Disponible en: <https://kodigo.org/introduccion-a-git-conceptos-basicos-y-su-importancia-en-el-desarrollo-de-software/>.

[2] Platzi, "¿Qué es un branch (rama) y cómo funciona un merge en Git?", *Platzi*, 2023. Disponible en: <https://platzi.com/clases/1557-git-github/19947-que-es-un-branch-rama-y-como-funciona-un-merge-en/>.

[3] Platzi, "Utilizando Pull Requests en GitHub", *Platzi*, 2023. Disponible en: <https://platzi.com/clases/1557-git-github/19957-utilizando-pull-requests-en-github/>.

[4] KeepCoding, "¿Qué es un commit en Git?", *KeepCoding*, 2023. Disponible en: <https://keepcoding.io/desarrollo-web/que-es-un-commit-en-git/>.

[5] Stack Overflow, "Git conflicts in rebase vs merge", *Stack Overflow*, 2021. Disponible en: <https://stackoverflow.com/questions/66090607/git-conflicts-in-rebase-vs-merge>.

[6] SmartBear, "What is unit testing?", *SmartBear*, 2023. Disponible en: <https://smartbear.com/learn/automated-testing/what-is-unit-testing/>.

[7] Pytest, "How to use assertions in tests", *pytest documentation*, 2023. Disponible en: <https://docs.pytest.org/en/stable/how-to/assert.html>.

[8] Trunk, "Flake8 Python Linter", *Trunk*, 2023. Disponible en: <https://trunk.io/linters/python/flake8>.

[9] Packet Coders, "Dynamically generating tests with pytest parametrization", *Packet Coders*, 2023. Disponible en: <https://www.packetcoders.io/dynamically-generating-tests-with-pytest-parametrization/>.