Preguntas Teóricas (20 pts, 2pts c/u)

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

Git es un sistema de control de versiones que realiza un seguimiento de los cambios en los archivos. Git es especialmente útil cuando un grupo de personas está haciendo cambios en los mismos archivos al mismo tiempo [1].

Y su relación reside en que, al cargar archivos en GitHub, los almacena en un "repositorio de Git". Esto significa que al realizar cambios (o "confirmaciones") en los archivos de GitHub, Git se iniciará automáticamente para realizar el seguimiento de los cambios y administrarlos [1].

La mayoría de los usuarios trabajan en sus archivos localmente (en su propio ordenador), luego sincronizan continuamente estos cambios locales y todos los datos de Git relacionados, con el repositorio central "remoto" en GitHub [1].

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

Branch: Es una rama que se crea dentro del mismo repositorio. Su finalidad es aislar el desarrollo de nuevas funcionalidades, correcciones o pruebas, evitando que dichos cambios alteren de inmediato la rama principal. Los *branches* comparten el historial y la configuración del repositorio en el que se crean, y están pensados para facilitar el trabajo colaborativo cuando todos los desarrolladores cuentan con acceso al mismo repositorio.

Fork: Consiste en la creación de una copia completa de un repositorio existente, alojada en una cuenta o una organización distinta a la del repositorio original. Esta copia conserva un vínculo con el repositorio de origen, lo que posibilita la sincronización de cambios y la propuesta de modificaciones mediante pull requests. El fork se utiliza comúnmente en entornos de colaboración abierta o cuando no se cuenta con permisos de escritura en el repositorio original, permitiendo trabajar de forma independiente sin comprometer el código base [2].

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

Un *Pull Request* (PR) es una propuesta para combinar un conjunto de cambios hechos en una rama con otra dentro de un repositorio. Antes de aceptar el PR, los colaboradores pueden revisar, comentar y aprobar o rechazar los cambios propuestos. Además, el PR muestra claramente las diferencias entre la rama de origen y la rama de

destino, facilitando la revisión y el control de calidad del código antes de integrarlo al proyecto principal [3].

4) ¿Qué es un commit?

Un commit es el registro permanente de cambios realizados en uno o más archivos de una rama del repositorio. Cada commit posee un identificador único que indica los cambios efectuados, su fecha y el autor. Incluye un mensaje descriptivo y, en entornos colaborativos, puede tener coautores o realizarse en nombre de una organización [4].

5) 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* ocurre cuando Git no puede combinar automáticamente dos versiones de un archivo porque la misma parte del código ha sido modificada de forma diferente en dos ramas [5].

Un *rebase conflict* es un caso similar, pero ocurre cuando se está intentando reorganizar los commits de una rama para integrarlos sobre otra (rebase) y existen cambios incompatibles en las mismas líneas o secciones de los archivos. En ambos casos, Git detiene la operación y solicita que el desarrollador decida manualmente qué cambios conservar y cómo resolver las diferencias [6].

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

Las pruebas unitarias consisten en evaluar la unidad funcional más pequeña de un programa para garantizar la calidad del código. Estas pruebas se escriben como código y se ejecutan automáticamente cada vez que se realizan cambios, permitiendo identificar rápidamente errores en áreas específicas. Además, fomentan el desarrollo modular, mejoran la cobertura y calidad de las pruebas, y liberan tiempo de los desarrolladores para concentrarse en la programación [7].

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

En pytest, assert se usa para verificar que una condición sea verdadera. Si la condición es falsa, la prueba falla. Esto permite comprobar automáticamente que el código produce los resultados esperados. Ejemplo:

```
assert suma(2, 3) == 5                # Verifica que la función suma retorne 5.
```

8) ¿Explique que son github-actions y su utilidad para el desarrollo continuo de código?

GitHub Actions es una plataforma de integración y despliegue continuos (CI/CD) que permite automatizar la compilación, prueba y despliegue de proyectos. Permite crear flujos de trabajo, los cuales se ejecutan ante eventos en el repositorio, como solicitudes de cambios, y pueden desplegar automáticamente cambios a producción [8].

9) ¿Qué es Flake 8?

Flake8 es un linter esencial para desarrolladores de Python que buscan mantener altos estándares de calidad de código. Es una utilidad de línea de comandos que verifica el código Python según el estilo de codificación (PEP 8), errores de programación y construcciones complejas. Lo que distingue a Flake8 es su naturaleza extensible, que permite la integración con diversos complementos para ampliar sus capacidades [9].

10) Explique la funcionalidad de parametrización de pytest.

El decorador @pytest.mark.parametrize en Pytest es una herramienta que permite ejecutar una misma función de prueba varias veces con diferentes conjuntos de datos de entrada. Para usarlo, se especifica primero una cadena con los nombres de los parámetros que la prueba utilizará, y luego una lista donde cada elemento contiene un conjunto distinto de valores que se pasarán a esos parámetros. De este modo, Pytest ejecuta la prueba de forma independiente para cada conjunto, lo que facilita cubrir múltiples escenarios sin repetir código. Esta técnica no solo mejora la claridad y mantenibilidad de las pruebas, sino que también optimiza el proceso al verificar diferentes comportamientos del código bajo variadas condiciones, contribuyendo a una mayor eficiencia en las pruebas automatizadas [10].

Referencias:

[1] GitHub Docs, *Acerca de GitHub y Git* [en línea]. Disponible: https://docs.github.com/es/get-started/start-your-journey/about-github-and-git. [Accedido: 11-ago-2025].

[2] GitHub Docs, *Fork a repository* [en línea]. Disponible: https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/working-with-forks/fork-a-repo?versionId=free-pro-team%40latest&productId=pull-requests&restPage=committing-changes-to-your-project%2Ccreating-and-editing-commits. [Accedido: 11-ago-2025].

[3] GitHub Docs, *Acerca de las solicitudes de incorporación de cambios* [en línea]. Disponible: https://docs.github.com/es/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-pull-requests. [Accedido: 11-ago-2025].

[4] GitHub Docs, *About commits* [en línea]. Disponible: https://docs.github.com/en/pull-requests/committing-changes-to-your-project/creating-and-editing-commits/about-commits#about-commits. [Accedido: 11-ago-2025].

[5] GitHub Docs, *Resolver un conflicto de fusión con la línea de comando* [en línea]. Disponible: https://docs.github.com/es/pull-requests/collaborating-with-pull-requests/addressing-merge-conflicts/resolving-a-merge-conflict-using-the-command-line. [Accedido: 11-ago-2025].

[6] GitHub Docs, *Resolving merge conflicts after a Git rebase* [en línea]. Disponible: https://docs.github.com/en/get-started/using-git/resolving-merge-conflicts-after-a-git-rebase. [Accedido: 11-ago-2025].

[7] Amazon, *What is Unit Testing?* [en línea]. Disponible: https://aws.amazon.com/what-is/unit-testing/. [Accedido: 11-ago-2025].

[8] GitHub Docs, *Entender las GitHub Actions* [en línea]. Disponible: https://docs.github.com/es/actions/get-started/understand-github-actions. [Accedido: 11-ago-2025].

[9] Trunk, *Flake8: The Python Linter for Code Quality and Style* [en línea]. Disponible: https://trunk.io/linters/python/flake8. [Accedido: 11-ago-2025].

[10] Allure *ReportParametrización en Pytest* [en línea]. Disponible: https://allurereport.org/es/docs/guides/pytest-parameterization/.
[Accedido: 11-ago-2025].