

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

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

Su principal utilidad radica en su capacidad para gestionar y rastrear cambios en el código fuente de manera eficiente y efectiva.

### **Control de Versiones**

Git permite a los desarrolladores llevar un registro detallado de todos los cambios realizados en el código a lo largo del tiempo.

### **Trabajo Colaborativo**

Git facilita la colaboración entre múltiples desarrolladores en un mismo proyecto.

### **Aislamiento de Cambios**

El uso de ramas permite a los desarrolladores trabajar en diferentes características o correcciones de errores simultáneamente sin interferir entre sí.

Proporciona una serie de funcionalidades claves para el desarrollo de proyectos como son: el control de cambios en el código, la reversibilidad de dichos cambios, y la posibilidad de colaborar en el desarrollo del código.

2) Explique la diferencia entre git y github

Git es la herramienta para realizar el seguimiento de los cambios, mientras que GitHub es el lugar para almacenar los proyectos y trabajar en equipo. GitHub usa Git para hacer su magia.

3) ¿Qué es un branch?

Unidad de almacenamiento que contiene los archivos y las carpetas que componen el conjunto de contenido de un proyecto.

Las ramas separan los flujos de trabajo, a los que se suele hacer referencia como versiones. Las colaboraciones siempre se realizan en una rama específica, a la que están limitadas exclusivamente.

4) En el contexto de github. ¿Qué es 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”, es una petición que el propietario de un fork de un repositorio hace al propietario del repositorio original para que este último incorpore los commits que están en el fork. Básicamente es una petición para integrar nuestras propuestas o cambios de código a un proyecto.

5) ¿Qué es un commit?

Es un sistema de control de versiones en el cual se crea lo que se entiende como confirmaciones, **estas** confirmaciones son las unidades básicas más importantes dentro del cronograma del proyecto de Git, las cuales se presentan como instantáneas o hitos dentro del proyecto de Git.

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

Una rebase de Git cambia la base de la rama del desarrollador de una confirmación a otra, por lo que parece que ha creado su rama a partir de una confirmación diferente. Internamente, Git crea una nueva confirmación y la aplica a la base especificada.

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

Los “merge conflicts” ocurren cuando se hacen cambios contrapuestos en la misma línea de un archivo o cuando una persona edita un archivo y otra persona borra el mismo archivo. Se resuelve decidiendo qué cambios incorporar desde las diferentes ramas de una confirmación nueva.

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

Las pruebas unitarias son un recurso viable para estos efectos, al aplicarse con el propósito de identificar errores, vulnerabilidades o funciones defectuosas. Los desarrolladores aíslan una línea del lenguaje codificado para saber si el sistema está operando correctamente en una función, proceso o actividad específica.

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

El “assert” es una forma de verificar si una determinada condición es verdadera en un punto específico de un programa. Si la condición es verdadera, el programa continúa su ejecución, pero si la condición es falsa, se genera una excepción AssertionError, que interrumpe la ejecución del programa.

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

- a) **(F401) module imported but unused:** este error ocurre cuando se realiza una importación innecesaria de un módulo en un código de programación, y esta no se utiliza del todo, lo que provoca un error de formato en el código.
- b) **(F501) invalid % format literal:** este error pasa cuando se quiere utilizar el operador de formato “%” y se utiliza de forma incorrecta en su sintaxis o la secuencia de formato donde no se especifica de manera concreta su utilidad.
- c) **(F621) assertion test is a tuple, which is always True:** este error sucede cuando se utiliza el comando “assert” y sea cual sea su condición, el resultado de una tupla siempre será verdadero, lo que provocaría que las pruebas siempre pasen aunque la lógica del código sea incorrecta.

## Referencias bibliográficas

Astigarraga, J., & Cruz-Alonso, V. (2022). ¿ Se puede entender cómo funcionan Git y GitHub!. *Ecosistemas*, 31(1), 2332-2332.

Lopez-Pellicer, F. J., Béjar, R., Latre, M. A., Nogueras-Iso, J., & Zarazaga-Soria, F. J. (2015, July). GitHub como herramienta docente. In *Actas de las XXI Jornadas de la Enseñanza Universitaria de la Informática* (pp. 66-73). Universitat Oberta La Salle.

Carlyrevier. (2024, 2 febrero). Aspectos básicos sobre Git y GitHub para documentación de Microsoft Learn - Contributor guide. Microsoft Learn. <https://learn.microsoft.com/es-es/contribute/content/git-github-fundamentals>

Terra, J. (s.f). ¿Qué es Git Rebase y cómo se utiliza? <https://www.simplilearn.com/what-is-git-rebase-command-article>

GitHub. (s.f). Resolver un conflicto de fusión con la línea de comando. <https://docs.github.com/es/pull-requests/collaborating-with-pull-requests/addressing-merge-conflicts/resolving-a-merge-conflict-using-the-command-line>

Tamushi. (2022). ¿Qué son las pruebas unitarias de software? <https://www.testingit.com.mx/blog/pruebas-unitarias-de-software>

N.D. (2024). Uso de la declaración Assert en Python. <https://keepcoding.io/blog/declaracion-assert-en>

python/#:~:text=La%20declaración%20assert%20en%20Python%20es%20una%20herramienta%20esencial%20para,y%20maneja%20adecuadamente%20las%20excepciones.

N.D . (2016) .Error / Violation Codes. <https://flake8.pycqa.org/en/latest/user/error-codes.html>