

## Preguntas Teóricas

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

Git es un sistema de control de versiones que guarda un historial de todos los cambios realizados en los archivos de un proyecto. Permite volver y trabajar en versiones anteriores, crear ramas separadas para probar ideas y combinar cambios de varios colaboradores.

GitHub es, por así decirlo, el lugar donde se guardan esos cambios en internet. Es una plataforma en línea que utiliza Git como su base, pero le pone un entorno para compartir proyectos, colaborar con otros, reportar errores y facilitar tareas.

En resumen: Git es la herramienta que maneja las versiones del código. GitHub es el servicio en la nube que guarda esos trabajos y facilita la colaboración.

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

En Git, un Branch es por así decirlo una línea de tiempo paralela dentro del mismo proyecto y funciona para trabajar en cambios o probar nuevas cosas sin afectar la versión principal del código. Sirve para hacer pruebas o cometer errores sin tener que afectar el código principal.

Un fork es una copia de un repositorio entero, pero hecha en tu propia cuenta de GitHub. Se usa para trabajar en un proyecto que no es de uno o que está en el repositorio de otra persona. Con un fork se puede modificar un proyecto sin afectar el original.

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

En GitHub, un Pull Request es una solicitud para que los cambios realizados en una rama o en un fork se revisen y se agreguen en otra rama del proyecto. Se puede usar cuando se quiere que el propietario de un repositorio o un compañero de equipo revise el trabajo hecho por uno antes de unirlo a la versión principal.

### 4) ¿Qué es un commit?

Un commit es, por así decirlo, un checkpoint en el trabajo. Se realiza cuando el código está funcionando bien y se quiere guardar ese estado para que, en el futuro, si algo sale mal, se pueda volver a ese punto exacto. Cada commit incluye un mensaje descriptivo que explica los cambios realizados, lo que facilita llevar un historial ordenado del proyecto y permite colaborar con otros de manera más eficiente.

### 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 o rebase conflict ocurre cuando, al intentar unir cambios de dos ramas distintas (mediante un merge o un rebase), Git detecta que la misma parte de un archivo ha sido modificada de forma diferente en cada rama. En este caso, Git no puede decidir automáticamente cuál versión conservar y requiere que el usuario resuelva el conflicto manualmente, eligiendo o combinando los cambios antes de completar la operación.

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

Una prueba unitaria es un tipo de prueba de software que se enfoca en verificar el funcionamiento correcto de una unidad pequeña del código, como una función, un método o un módulo específico. Su objetivo es asegurar que cada parte del programa funcione correctamente de forma individual antes de integrarla con otras. Estas pruebas usualmente son automáticas y pueden encontrar errores de forma temprana, facilitando el mantenimiento y la calidad del software.

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

En el contexto de pytest, la instrucción `assert` se usa para verificar que una condición específica sea verdadera mientras se ejecuta una prueba. Si la condición es verdadera, la prueba continúa normalmente, si es falsa, la prueba falla y pytest muestra un mensaje indicando qué valor se esperaba y cuál se obtuvo. Esto nos deja comprobar de forma fácil que el código funciona como se espera y encuentra errores de forma rápida.

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

GitHub Actions es una herramienta de GitHub que permite automatizar tareas en un proyecto. Por ejemplo, se puede configurar para que cada vez que se sube código, el sistema revise que todo funcione bien, compile el programa o lo publique automáticamente. Esto ayuda a ahorrar tiempo y a asegurarse de que el código siempre esté listo y sin errores.

9) ¿Qué es Flake 8?

Flake8 es una herramienta que analiza el código escrito en Python para verificar que siga las reglas de estilo y buenas prácticas definidas en la guía PEP 8 (manual de estilo oficial para escribir código en Python). Además, detecta errores comunes y problemas en el código, lo que ayuda a mantenerlo limpio, ordenado y fácil de entender.

10) Explique la funcionalidad de parametrización de pytest.

La parametrización en pytest permite ejecutar una misma prueba varias veces con diferentes conjuntos de datos de entrada. Esto evita tener que escribir múltiples funciones de prueba para cada caso y facilita comprobar que el código funciona correctamente en distintas situaciones. Con esta funcionalidad, se mejora la cobertura de pruebas y se ahorra tiempo al automatizar la validación de múltiples escenarios.