1 ¿Explique qué es git y su relación con GitHub?

Git es un sistema de control de versiones distribuido que permite llevar un registro del historial de cambios de un proyecto. Cada desarrollador trabaja con una copia local completa del repositorio, lo que facilita la colaboración y recuperación de versiones anteriores. GitHub es una plataforma en línea que utiliza git como base y permite almacenar repositorios de forma remota, además de ofrecer herramientas para la colaboración (Pull Requests, issues, integración continua, etc.).

2 ¿Qué es un branch? ¿Qué es un fork?

Un branch (rama) es una línea independiente del desarrollo que permite trabajar en nuevas funcionalidades sin afectar la rama principal del código. Un fork es una copia completa de un repositorio que se aloja en otra cuenta de GitHub y permite desarrollar una versión independiente, normalmente usada cuando no se tiene acceso directo al repositorio original.

3 En el contexto de GitHub, ¿qué es un Pull Request?

Es una solicitud para integrar cambios realizados en una rama o en un fork dentro de la rama principal del repositorio original. Permite realizar revisiones de código, discutir cambios y aprobarlos antes de fusionarlos.

4 ¿Qué es un commit?

Es un registro de cambios realizados sobre los archivos del repositorio en un punto específico del tiempo. Incluye una descripción que explica lo que se modificó y permite reconstruir el estado del proyecto en ese momento.

5 Explique qué 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 o rebase conflict aparece cuando dos ramas modifican las mismas líneas de un archivo de forma incompatible. Git no puede elegir automáticamente cuál versión conservar y le solicita al usuario que resuelva el conflicto manualmente para poder completar la operación.

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

Una prueba unitaria es un pequeño caso de prueba que verifica el comportamiento de una función o componente específico del código de forma aislada. Sirve para asegurar que cada parte del sistema funciona como se espera y facilita el mantenimiento del software.

7 Bajo el contexto de *pytest*, ¿cuál es la utilidad de un "assert"?

El `assert` es la instrucción que permite verificar una condición dentro de un test. Si la condición no se cumple, pytest marca la prueba como fallida. En otras palabras, es la forma en la que se expresa lo que se espera que el código retorne.

8 ¿Explique qué son GitHub Actions y su utilidad para el desarrollo continuo de código?

GitHub Actions es una herramienta dentro de GitHub que permite automatizar flujos de trabajo. Se utiliza para ejecutar tests automáticamente, generar builds, correr análisis de estilo (como flake8) o desplegar software cada vez que se hace un commit o un Pull Request, favoreciendo el desarrollo continuo y la integración automática.

9 ¿Qué es Flake 8?

Flake8 es una herramienta que analiza el código Python y detecta errores de estilo y posibles errores lógicos. Se basa en las convenciones PEP8 y ayuda a mantener un código limpio y legible.

10 Explique la funcionalidad de parametrización de *pytest*

La parametrización permite ejecutar el mismo test varias veces utilizando diferentes conjuntos de datos. En lugar de escribir varios tests independientes, se define un único test con parámetros y pytest genera automáticamente todas las combinaciones, lo que hace las pruebas más completas y fáciles de mantener.