

Preguntas Teóricas

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

Git permite que el código fuente se pueda gestionar de manera muy eficiente, sumamente útil para equipos de desarrollo en los que se busca trabajar simultáneamente y cuando se busca implementar práctica de desarrollo. Mediante Git se controlan versiones de código con lo cual se puede gestionar eficientemente al tener distintas versiones según fechas, avances o cualquier otro criterio que se considere importante, como volver a puntos clave o desistir de errores nuevos por ejemplo. Ayuda a rastrear cualquier cambio que se haya hecho. Se pueden fusionar códigos en un repositorio común y tener copias locales según se requiera. En Git se pueden atacar errores de manera aislada mediante ramas. También se tiene un historial completo de cambios que incluye fechas y detalles de lo realizado.

2) Explique la diferencia entre git y github

Git es la herramienta de control de versiones comentada anteriormente. GitHub es una plataforma en la que se facilita el uso de Git en los distintos proyectos de software que se tengan. GitHub se usa para colaboración y como red social en el que se puede interactuar con distintos proyectos. También tiene una interfaz web útil para interactuar con repositorios más fácilmente.

3) ¿Qué es un branch?

Una rama en git permite trabajar con distintas versiones en busca de no alterar el código principal si no se requiere. Permite hacer cambios sin alterar a la rama principal, útil para trabajar en paralelo, que luego se puede fusionar fácilmente si se tiene un resultado correcto. De igual forma en el proceso se tienen un historial de commits para analizar lo realizado. Estas se pueden crear y eliminar de manera sencilla.

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

Es un mecanismo con el que se puede revisar el código y sus cambios de manera sencilla. De igual forma, permite una mejor discusión y en los trabajos de equipo

mejorar la organización y eficiencia en los proyectos. Los cambios se integran de forma controlada en los proyectos.

5) ¿Qué es un commit?

Es la acción en git de guardar cambios hechos en el código en el historial del repositorio. Con esto se tienen puntos de control para saber el estado del código en distintos momentos. Es importante destacar los mensajes en los commit que permiten un contexto de la situación y facilita trabajar en este sea luego de un tiempo y en ocasiones para que otros desarrolladores puedan comprender la situación.

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

El comando de Git “rebase” se enfoca en integrar modificaciones de una rama a otra al cambiar su base. Es decir, hace que parezca que esa rama se ha creado desde un commit diferente. Al ejecutar “git rebase main”, se reescribe la historia de la rama para que parezca que esta se ha creado a partir del último commit en la rama “main”.

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

Un “merge conflict” se refiere a cuando se realizan cambios en las mismas líneas de un archivo y Git por sí solo no puede resolver esas diferencias. Sucede durante la combinación de ramas que contienen cambios en las mismas partes del código. Para poder resolverlo, hay que identificar el conflicto usando “git status”, donde Git indica los archivos en conflicto. Después, se modifica el archivo usando un editor de texto, donde se decide si conservar cierta rama o combinar las opciones. Al finalizar, se eliminan los marcadores que indica Git, dejando solo el código deseado. Para añadir y confirmar los cambios, se utiliza “git add archivo”, para añadir el archivo al staging área, y un “git commit” para confirmar la resolución de los conflictos.

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

Se refiere a una técnica de prueba de software centrada en verificar el funcionamiento de las unidades más pequeñas, cuyo funcionamiento se puede probar, tales como las

funciones, clases o métodos. Asegura que estas unidades funcionen correctamente de forma aislada antes de ser integradas en un sistema más amplio.

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

Un “assert” es una declaración que verifica si una condición es verdadera o falsa dentro de una prueba unitaria. En un pytest, comprueba si el resultado obtenido en funciones o métodos coincide con lo esperado.

10) Mencione y explique 3 errores de formato detectables con Flake8.

El primero es el error de indentación incorrecta. Flake8 verifica que la indentación sea consistente en todo el archivo que revise. Por ejemplo, detecta si en el código se mezcla el uso de tabulaciones y espacios para realizar las indentaciones.

El segundo son errores con respecto a espacios en blanco, donde detecta el uso incorrecto de los mismos. Por ejemplo, tener espacios en blanco al final de una línea, espacios adicionales antes o después de operadores, e, incluso, espacios en blanco innecesarios en la definición de funciones o clases.

El tercero se refiere a errores en nombres de variables y funciones. Flake8 se encarga de detectar que estos sigan las convenciones de nomenclatura de Python, por ejemplo, utilizar letras minúsculas y guiones bajos entre palabras.