

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

La principal herramienta es permitir un control de versiones de forma eficiente y robusta. Esto por medio del rastreo de cambios realizados en el código, la reversión de cambios a versiones anteriores, el trabajo colaborativo sin interferir entre sí, las distintas ramas y fusiones para desarrollar sin afectar la rama principal y cuando esté listo poder integrar, el historial detallado proporcionando un mensaje de commit y finalmente la resiliencia y disponibilidad que ofrece ya que permite trabajar sin conexión.

**2) Explique la diferencia entre git y github**

Git es una herramienta de control de versiones, la cual se instala en la maquina local del desarrollador y permite realizar distintas acciones a la hora de guardar proyectos de software, mientras que GitHub es una herramienta para alojar código basada en web y que utiliza git, además cuenta entorno gráfico.

**3) ¿Qué es un branch?**

Es una línea independiente de desarrollo dentro de un repositorio. Cada rama es una versión separada del proyecto que permite a los desarrolladores trabajar en tareas distintas o correcciones de manera aislada, sin afectar la rama principal o el trabajo de otros colaboradores.

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

Es una solicitud de colaboración que permite que desarrolladores propongan cambios en un repositorio. Su funcionamiento es de la siguiente manera: Un usuario crea un pull request desde su repositorio local hacia una rama principal del repositorio, en dicho pull request se hace una descripción de los cambios, otros desarrollares revisaran este pull request, sugiriendo mejoras, cuando se aprueben dichos cambios, se realiza un merge y los cambios se fusionan con el código principal.

**5) ¿Qué es un commit?**

Es una operación que guarda una instantánea del estado actual de los archivos del repositorio. Cada commit es un punto en el historial del proyecto y representa un conjunto de cambios o actualizaciones al código fuente. Estos permiten a los desarrolladores registrar el progreso y los cambios de una manera detallada y ordenada.

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

Es una operación de reubicación de una rama, en donde se integran los cambios de una rama en otra, en el caso de **git rebase main**, se toman los nuevos cambios del main y se integran a la rama que se desea.

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

Ocurre en Git cuando se intenta fusionar dos ramas que tienen cambios conflictivos en las mismas partes de los mismos archivos, esto sucede cuando dos desarrolladores han

editado las mismas líneas de un archivo en sus respectivas ramas. Git intenta combinar los cambios, pero si hay modificaciones incompatibles en las mismas líneas o áreas del archivo no puede decidir cual camino debe prevalecer. Para resolverlo se deben abrir los archivos en conflicto en un editor de texto, Git marca los conflictos en el archivo con delimitadores especiales. Seguidamente, se edita el contenido entre los delimitadores para resolver el conflicto y luego se eliminan los delimitadores. Luego, se marcan los archivos como resueltos añadiéndolos al área de staging. Se completa la fusión realizando un commit y se verifica que se hayan resuelto los conflictos.

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

Es un tipo de prueba automatizada en el desarrollo de software que se enfoca en verificar el correcto funcionamiento de unidades individuales de código, como funciones, métodos o incluso clases. El objetivo principal es asegurarse de que cada unidad de código funcione como se espera de manera aislada. Esto ayuda a la detección temprana de errores, facilita el refactorizado, es una documentación adicional y mas importante mejora la calidad del código.

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

La función assert se usa para validar el comportamiento de un código a verificar, pytest posee la ventaja de que proporciona mensajes de error detallados cuando un assert es falso.

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

- a) Longitud de Línea Excesiva (E501): PEP 8 recomienda que las líneas de código no excedan los 79 caracteres. Esto mejora la legibilidad del código y facilita la visualización del código en editores de texto con pantallas divididas.
- b) Espacios en blanco innecesarios (E201, E202, E203): El uso de espacios en blanco en varias partes del código puede afectar la legibilidad y consistencia del código.
- c) Importaciones no usadas(F401): Las importaciones que no se utilizan en el código deben ser eliminadas para mantener el código limpio y evitar dependencias innecesarias.