

Preguntas teóricas

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

La principal utilidad de git es que permite a los desarrolladores trabajar proyectos en paralelo y no ver afectado su flujo de trabajo por los cambios que pueda estar realizando otro desarrollador. Esto gracias a que facilita la creación de ramas para que los desarrolladores puedan trabajar individualmente en diferentes tareas. También cuenta un sistema de control de versiones, lo que da la flexibilidad de recuperar trabajo previo.

2. ¿Qué es un branch?

Un “branch” intuitivamente se visualiza como una ramificación del repositorio. Se suele iniciar con la rama “main”, aunque usualmente corresponde más que todo al tronco del repositorio. Entonces, para hacer cambios al repositorio sin miedo a arruinar el main actualmente funcional, se crea una ramificación, un branch, y ahí se modifica todo lo que se desee sin miedo, pues si no funciona siempre se puede revertir y el main quedará intacto. Por otro lado, si los cambios son satisfactorios y pasan las pruebas, se realiza un “merge” y se fusionan los cambios realizados en la rama con el main.

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

Se le llama “Pull Request” a la solicitud que realiza el desarrollador que estaba trabajando en la “branch” para que el coordinador o encargado del proyecto revise los cambios y una o haga un “merge” de la “branch” con el main.

4. ¿Qué es un commit?

Un “commit” es la acción de guardar los cambios hechos en una rama en el repositorio local. Tienen su identificador propio, por lo que sirven para rastrear los cambios hechos a la rama correspondiente.

5. Describa lo que sucede al efectuar las siguientes operaciones: “git fetch” “git rebase origin/master”

El “git fetch” es un comando que se utiliza para comprobar si se han realizado cambios en el repositorio “main” o master. Esta comprobación se realiza desde la “branch” en la que se está editando el código.

El “git rebase origin/master” se utiliza para mover o modificar una secuencia de “commits” en el repositorio base a partir de un “branch”, de esta forma se crea un nuevo “commit” en el “branch” con la nueva secuencia. Se puede utilizar para actualizar y agregar “features” al master y además al realizar esto actualizar el “branch” con los últimos cambios que tenga el master.

6. 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.

Los conflictos ocurren cuando dos personas están trabajando en el mismo archivo del mismo branch y no están coordinados. Lo que ocurre es que ambos hacen modificaciones e intentan hacer push a la misma rama. En estos casos una de las partes debe revertir sus cambios y hacerlos de otra forma, ya sea en otra rama o haciéndolos por encima de los cambios de la otra parte.

7. ¿Qué es una Prueba Unitaria o Unittest en el contexto de desarrollo de software?

Un unit test hace referencia a un mecanismo de prueba que se utiliza para comprobar que pequeñas partes del código estén funcionando adecuadamente. Con esto se pretende segmentar la revisión del código en pequeñas partes y así evitar acarrear erros hasta etapas más avanzadas del proyecto.

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

Es una función que permite verificar lo que se espera que una función de Python devuelva (return). Al utilizar esta función uno indica el valor esperado, y si la función retorna otro valor, el programa fallará e indicará el verdadero valor que retorna o un mensaje personalizado que uno escriba.

9. ¿Qué es Flake 8?

Flake8 es una librería de verificación de código de Python, se utiliza para la detección automática de errores en el código. Los errores que puede detectar esta librería son como, por ejemplo: “library imported but unused”, “Undefined name” y código que no está indentado.

10. Explique la funcionalidad de parametrización de pytest.

Sirve para hacer una misma prueba con múltiples sets de entradas.

Referencias

About branches. GitHub Docs. (n.d.). <https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-branches>

About commits. GitHub Docs. (n.d.-b). <https://docs.github.com/en/pull-requests/committing-changes-to-your-project/creating-and-editing-commits/about-commits>

About merge conflicts. GitHub Docs. (n.d.-c). <https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/addressing-merge-conflicts/about-merge-conflicts>

Atlassian. (n.d.-a). *Git Rebase: Atlassian Git Tutorial*. Atlassian. <https://www.atlassian.com/git/tutorials/rewriting-history/git-rebase>

Atlassian. (n.d.). *Pull-requests: Atlassian Git Tutorial*. Atlassian. <https://www.atlassian.com/es/git/tutorials/making-a-pull-request#:~:text=Cuando%20realizas%20una%20pull%20request,de%20tu%20repositorio%20al%20suyo.>

Carrillo, J. (2021, January 23). *Git fetch vs pull: ¿Cuál es la diferencia entre los comandos git fetch y git pull?*. freeCodeCamp.org. <https://www.freecodecamp.org/espanol/news/git-fetch-vs-pull-cual-es-la-diferencia-entre-los-comandos-git-fetch-y-git-pull/>

Diaz, G. A. (2020, June 22). *Escribiendo Código de Alta Calidad en Python (2020)-parte 2: Linters*. Medium. <https://medium.com/@gonzaloandres.diaz/escribiendo-codigo-de-alta-calidad-en-python-parte-2-linters-64ffd8d2df91#:~:text=Flake8%20es%20una%20librer%C3%ADa%20de,para%20verificar%20la%20complejidad%20ciclom%C3%A1tica.>

How to write and report assertions in Tests¶. How to write and report assertions in tests - pytest documentation. (n.d.). <https://docs.pytest.org/en/7.1.x/how-to/assert.html>

Montiel, O. (2022, February 22). *La Guía para principiantes de Git y Github*. freeCodeCamp.org. <https://www.freecodecamp.org/espanol/news/guia-para-principiantes-de-git-y-github/>

Pytest - parameterizing tests. Tutorialspoint. (n.d.). https://www.tutorialspoint.com/pytest/pytest_parameterizing_tests.htm

Team, K. (2023, May 3). *¿Qué son las pruebas unitarias de software?*. KeepCoding Bootcamps. <https://keepcoding.io/blog/que-son-las-pruebas-unitarias-de-software/>

