

Preguntas Teóricas

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

Git es una herramienta que lleva a cabo la gestión de versiones de código de manera distribuida. Se trata de una plataforma de código abierto con mantenimiento continuo y ostenta el título de ser la herramienta de este tipo más utilizada a nivel mundial. Numerosos proyectos de software, tanto comerciales como de código abierto, confían en Git para llevar a cabo la gestión de versiones de su código fuente.

2. ¿Qué es un branch?

Una rama o branch es una versión del código del proyecto sobre el que estás trabajando. Estas ramas ayudan a mantener el orden en el control de versiones y manipular el código de forma segura.

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

Un Pull Request es una función de GitHub que permite a tu equipo solicitar la revisión y aprobación de sus cambios antes de fusionarlos en la rama principal de desarrollo, denominada “master” o “main”.

4. ¿Qué es un commit?

El comando commit se usa para confirmar los cambios en el área de preparación y guardarlos en el repositorio local en Git. Sin embargo, antes de guardar algún cambio, tienes que decirle a Git cuáles cambios quieres incluir de los muchísimos que puedes haber hecho.

5. Describa lo que sucede al ejecutar la siguiente operación: “git cherry-pick <SHA>”.

Al ejecutar la operación "git cherry-pick <SHA>", se está solicitando a Git que aplique los cambios introducidos por el commit identificado por el hash SHA especificado en la rama actual. Esto implica seleccionar ese commit específico y aplicar sus cambios en la rama actual, creando un nuevo commit.

El comando "git cherry-pick" se utiliza para traer cambios específicos de una rama a otra, sin tener que fusionar ramas completas. Al ejecutar esta operación, Git intentará aplicar los cambios del commit identificado por <SHA> en la rama actual. Si la aplicación de esos cambios es exitosa, se crea un nuevo commit en la rama actual con esos cambios.

6. Explique que es un “merge conflict” y como lo resolvería.

Es cuando dos personas modifican exactamente la misma línea de código, en este caso Git no sabe cómo resolver el problema y genera un error llamado merge conflict.

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

Las pruebas unitarias son el proceso en el que se prueba la unidad funcional de código más pequeña. Las pruebas de software ayudan a garantizar la calidad del código y son una parte integral del desarrollo de software. Una práctica recomendada en el desarrollo de software es escribir el software como unidades pequeñas y funcionales, y luego escribir una prueba unitaria para cada unidad de código. Puede escribir primero pruebas unitarias como código. Luego, ejecute ese código de prueba de forma automática cada vez que realice cambios en el código del software.

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

Pytest permite utilizar el Python estándar assert para verificar expectativas y valores en las pruebas de Python.

9. ¿Mencione y explique 3 errores de formato detectables con Flake8?

Variables no utilizadas o importaciones redundantes: PyFlakes puede identificar variables que se han definido, pero no se utilizan, así como importaciones que no se están utilizando en el código.

Convenciones de estilo no seguidas: Utiliza pycodestyle (anteriormente conocido como pep8) para verificar que tu código sigue las convenciones de estilo definidas en PEP 8. Esto incluye reglas sobre el formato del código, la indentación, la longitud de línea, entre otros.

Errores de sintaxis: Flake8 utiliza PyFlakes para realizar análisis estático del código, lo que le permite identificar errores de sintaxis en tu código antes de ejecutarlo.

10. Explique la funcionalidad de parametrización de pytest.

En Pytest, la parametrización es una característica que te permite ejecutar una prueba con diferentes conjuntos de datos o argumentos. Permite definir múltiples conjuntos de datos y ejecutar la misma prueba para cada conjunto de datos, ayudando a reducir la redundancia en la escritura de pruebas.

Referencias

HubSpot Product Team (2020). An intro to Git and Github for beginners (Tutorial). GitSpot. Recuperado de <https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners>.

Github Inc (2024). Documentación de Github. <https://docs.github.com/es/get-started/start-your-journey/hello-world>