

I. PREGUNTAS TEÓRICAS

A. *¿Diferencie la herramienta Git de Github?*

Git es un sistema de gestión de archivos de proyectos, con la facilidad de poder tener acceso a versiones anteriores de los mismos proyectos, además de ser una herramienta de código abierto la cual se controla de manera local y no centralizada. Github es una plataforma en línea basada en la nube, que ofrece proyectos de otros desarrolladores, así como observar e identificar problemas de su código, además de cargar y descargar recursos usando la herramienta Git.

B. *¿Qué es un branch?*

Un branch (o rama como su traducción al español), es un espacio independiente que pertenece a un proyecto principal, es decir, que, al crear una rama en nuestro proyecto, podemos trabajar en esta sin la consecuencia de dañar o borrar datos del archivo original. Es de gran utilidad crear branches cuando el proyecto está siendo manipulado por varios usuarios, donde cada una de estas modificaciones son independientes.

C. *¿Qué es un commit?*

Commit es un comando que captura los comandos utilizados en el proyecto en un momento específico. Estas capturas son consideradas como las “versiones” del proyecto, las cuales son subidas al repositorio remoto. Este comando es comúnmente usado después del comando “git add” que prepara los cambios del proyecto.

D. *¿Qué es la operación cherry-pick?*

La operación cherry-pick es una función que permite copiar cambios realizados en una rama y colocarlos en una rama diferente. Este comando es sumamente útil cuando se necesita realizar un cambio muy específico en una de las ramas del repositorio.

E. *¿Qué hace el comando git stash?*

El comando “git stash” almacena de manera temporal los cambios realizados en el código, para dejar el programa en “stand by” si se necesita trabajar en algo más y retomarlo luego.

F. *Compare las operaciones git fetch y git pull*

Git fetch es un comando que nos permite confirmar si hay metadatos dentro del repositorio original, mostrando los cambios realizados, mientras que Git pull revisa si hay metadatos en el repositorio original y los transfiere al repositorio remoto.

G. *Asumiendo que usted está en un Branch llamado “secundario” y su Branch principal se llama “master” ¿Qué resultado espera de hacer git rebase master? ¿Qué resultado espera de hacer git rebase origin/master?*

Se conoce como “origin” a un repositorio común, o rama secundaria donde los miembros de un equipo realizan los cambios del proyecto. Por otro lado, el master es la rama

principal donde se inicia el proyecto, y los cambios realizados en el master pueden fusionarse a cualquier rama secundaria. El comando “rebase” integra cambios desde una rama a otra. Si nos encontramos en un branch llamado “secundario” y aplicamos un rebase master se hace el rebase a la rama secundaria que se encuentra conectada al origen, pero no es el origen específicamente. Al hacer git rebase origin/master, se hace un “rebase” de la rama master del origen produciendo una reorganización en nuestra rama “master”, incorporando todos los nuevos commits de la rama “secundaria”, reescribiendo de esta manera el historial del proyecto y dejándolo más organizado.

Al hacer git rebase origin/master, se hace un “rebase” de la rama actual contra la rama “master”.

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

La prueba unitaria comprueba el debido funcionamiento de los componentes de software. Su propósito es de validar cada una de las partes de este, como su desempeño y diseño. Es utilizado para descubrir la mayor cantidad de errores posibles en el código cuando aún se encuentra en una fase de desarrollo temprano.

I. Bajo el contexto de pytest. ¿Qué es un “assert”?

Las declaraciones de “assert” son, básicamente, un test o una prueba. La idea general es configurar los datos de entrada requeridos, llamar a una función o método desde el código fuente y luego confirmar, mediante un “assert”, que el resultado se ajusta a las expectativas.

J. ¿Qué es Flake 8?

Flake8 es una librería de Python que permite revisar que se cumpla el estilo PEP8 y detecta errores de programación como variables declaradas, pero no usadas, imports que no se usan, problemas de bucles, etc. Lo que permite que el código final sea más limpio.