

## **Respuestas Sección Teórica**

- 1) Git es un software de control de versiones diseñado por Linus Torvalds. El término “control de versiones” se refiere a la gestión de diversos cambios en cuanto a la estructura o configuración que se le pueden realizar a algún software que se encuentra en desarrollo. En otras palabras, Git es un software que nos facilita la administración de diferentes versiones de un software que se encuentra en desarrollo y es susceptible a cambios en su código a lo largo del proceso de diseño y posterior al mismo, además es posible la consulta de los cambios realizados como una especie de registro o historial. Git es una herramienta eficiente y confiable para el mantenimiento de versiones de aplicaciones que poseen un gran número de archivos de código fuente, lo que implica una facilidad al momento de trabajar en equipo sobre el desarrollo de un mismo software.
- 2) Github es un sitio web y servicio en la nube que permite, a los desarrolladores de software, almacenar y administrar algún código desarrollado, así como llevar control de cualquier cambio realizado sobre este código. En otras palabras, Github ofrece un servicio de hosting de repositorios(directorio donde se almacenan los archivos de un proyecto) almacenados en la nube, con el cual se facilita la interacción con otros desarrolladores y la colaboración entre los mismos.
- 3) Una rama o “branch” es una copia del repositorio de un desarrollador. Esta técnica puede ser utilizada cuando se quiere realizar un desarrollo de forma aislada, debido que trabajar en una rama no afectará el repositorio central u otras ramas. Una rama puede ser combinada con otras ramas y con el repositorio central mediante otra técnica llamada pull request, lo cual establece otra ventaja de trabajar con ramas.
- 4) El término “commit” hace referencia a la acción de guardar o subir los archivos de un desarrollo a un repositorio remoto(Github) o donde se encuentren los distintos archivos que forman parte del software en desarrollo.
- 5) Un stage se puede considerar como un paso a realizar previo a la aplicación de un commit sobre un archivo. Cuando se habla de que un archivo está staged, se hace referencia a que git tiene conocimiento sobre el cambio, pero este cambio no será permanente en el repositorio. El siguiente commit que se realice al archivo, incluirá los cambios staged realizados previamente sobre el archivo.
- 6) El comando “git checkout” permite crear ramas o cambiar entre las ya existentes. Por ejemplo el siguiente comando crea una nueva rama y se cambia a ella:

command git checkout -b <branch-name>

Mientras que para cambiar de una rama a otra se puede emplear el siguiente comando:

```
git checkout <branch-name>
```

- 7) El comando “git stash” permite guardar el estado en el que se encuentra el proyecto en un momento determinado, con todos los cambios a los cuales no se les ha aplicado un commit, y los guarda en una pila provisional brindando la posibilidad de poder recuperarlo más adelante.
- 8) El comando “git add” puede ser utilizado para agregar contenido del directorio de trabajo al área de ensayo(staging area o index). Así al ejecutar el comando “git commit”, este sólo apunta a esta área de ensayo. Otra utilización del comando “git add” es el rastreo de un archivo, esto implica que este comando puede recibir tanto una ruta de archivo como de un directorio, en caso de ser un directorio, el comando añade de forma recursiva los archivos que están dentro del mismo.
- 9) El comando “git reset ~HEAD” se utiliza para deshacer cosas o acciones, este comando se mueve alrededor del puntero “HEAD” y opcionalmente cambia el index y el directorio de trabajo si se agregan algunas instrucciones más al comando.
- 10) Pytest es una librería de testing que permite escribir códigos de prueba usando Python. Se puede escribir y probar código orientado a bases de datos e interfaces de usuario (UI), pero principalmente se orienta hacia pruebas en códigos de programación de interfaz de aplicaciones (API).
- 11) Pytest permite la implementación del comando estándar de Python, el cual permite verificar valores esperados al ejecutar una función de prueba. Al comando de assert se le asignará el valor que se espera que la función retorne y mediante la ejecución de dicha función se puede verificar se cumple dicha condición establecida en el assert. En caso de que el assert falle, pytest muestra en pantalla que los valores esperados no coinciden con el assert establecido.
- 12) Flake8 es una de las guías de estilo que se puede seguir cuando se programa en python. Estas guías de estilo permiten mantener la uniformidad del código, para que este sea legible para personas ajenas al mismo, ya que establecen una serie de reglas sobre cómo escribir en código python. En el caso de la flake8, esta guía permite verificar que se cumpla la guía PEP8(actualmente llamada pycodestyle), revisar si existen variables declaradas pero no usadas, imports que no se usan, entre otras funciones. Las guías de estilo como flake8 se emplean con el objetivo de que el código resultante sea lo más limpio posible.

## Referencias

- Chacon, S. & Straub, B. (2014). *Pro Git*. New York: Apress.
- Ziade, T. (2019). Pypi: flake8. Recuperado de: <https://pypi.org/project/flake8/#description>