

Preguntas Teóricas (24 pts, 2pts c/u)

1) ¿Qué es Git?

R/ Git, es un software de control de versiones diseñado por Linus Torvalds. Para comprender mejor al respecto, se define como control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo, es decir, se trata de la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración, y para los que aún no les queda claro del todo lo que se tiene hasta el momento.

Básicamente el control de versiones es lo que se hace o aplica a la hora de desarrollar una página web o un software. También, es cuando una persona sube y actualiza su código en la nube, o le añaden alguna parte o simplemente lo editan aspectos que no funcionan como deberían o al menos no como se espera o solicita.

En resumen, cuando se habla de un sistema de control de versiones, en concreto se trata de todas las herramientas que permiten hacer modificaciones sobre el código de la persona y hacen que sea más sencilla la administración de las distintas versiones o variantes de cada producto desarrollado; precisamente a esto se le llama Git.

2) ¿Qué es Github?

R/ Se trata de una derivación del Git, ya que este último es el primer eslabón de herramientas de estructura a plataforma como lo es Github. Este se trata de una plataforma de uso masivo, que se usa para albergar proyectos de software libre y otro tipo de proyectos que pueda tener una organización y básicamente su función principal radica en ser un servicio de gestión de control de versiones distribuidas.

3) ¿Qué es un branch?

R/ Se trata de ramificaciones de un proyecto principal, que no son más que una copia temporal con la que se trabaja en paralelo al proyecto a desarrollar para implementar o probar estos cambios. Si las pruebas creadas en un “Branch” resultan ser satisfactorias, se puede de nuevo integrar esta rama al hilo principal, no

obstante, en caso contrario, se pueden descartar los cambios realizados sin tener que asentar ninguna nueva revisión al hilo principal. Con un Branch se listan ramas de trabajo.

4) ¿Qué es un commit?

R/ Consiste en un grupo de cambios sobre uno o varios ficheros, los cuales llevan una descripción, una fecha, un autor y demás. Los commits se categorizan como “locales” y “remotos”. Un commit se conoce como local hasta que no se efectúa la subida de los cambios realizados al servidor. Los commits locales pueden ser modificados sin ningún problema. Asimismo, se considera que un commit se vuelve remoto cuando se suben los cambios al servidor, pues se considera que dichos cambios ahora formarán parte del histórico compartido entre los desarrolladores del proyecto y, por ello, no es una buena práctica modificarlos del mismo modo en que se hace cuando los commits son los locales y esto último se debe a que la solución puede llevar a más tiempo invertido en caso de errores.

5) ¿Qué es la operación cherry-pick?

R/ La acción de cherry picking se basa en tomar un commit de una branch y aplicarlo a otra. Por ello, si se tiene uno o más commits, con el comando git cherry-pick, se puede aplicar cambios que cada uno de estos commits introduce, registrando cada cambio en un nuevo commit.

6) ¿Qué hace el comando git checkout?

R/ Este comando permite crear una nueva rama en un repositorio y además moverse de inmediato a ella. Es muy importante saber que la nueva rama que se crea estará basada en la rama en la que el desarrollador se encuentre en ese momento. También se utiliza para deshacer los cambios realizados sobre ficheros del workspace, ya sea sobre todos los ficheros o sobre unos ficheros en concreto.

7) ¿Qué hace el comando git stash?

R/ Guardar los cambios del *workspace* y del *stage* para realizar alguna operación (actualizar cambios desde el servidor, cambiar de rama, etc.). Para ello usaremos el *stash*. Puede pensarse en el *stash* como una lista que contiene grupos de cambios temporales, los cuales se pueden consultar, aplicar en el *workspace* actual, etc.

8) Compare las operaciones git fetch y git pull

R/ La operación git fetch, descarga commits, archivos y refs de un repositorio remoto al repositorio local. Este comando recupera la información y los datos desde otro repositorio, y los ubica en una branch oculta. Por ello, al recuperar esta información, no la fusionará o integrará en el repositorio local.

La operación git pull es empleada para realizar el fetching y descarga de información de un repositorio remoto y, de una vez, actualizarlo e integrar este contenido en el repositorio local. Se puede decir que este comando es la combinación de otros dos: un git fetch seguido de un git merge.

9) ¿Qué hace el comando git reset ~HEAD?

R/ A través de este comando, se obtiene una manera rápida de eliminar el último commit sin tener que buscar el identificador del commit anterior.

10) ¿Qué es Pytest?

R/ El pytest es una herramienta de prueba de Python con todas las funciones que te ayuda a escribir mejores programas. Asimismo, pytest facilita la escritura de pruebas pequeñas, pero se escala para admitir pruebas funcionales complejas para aplicaciones y bibliotecas.

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

R/ Un assert básicamente le está diciendo al programa que pruebe esa condición e inmediatamente active un error si la condición es falsa. El assert es una instrucción de Python que permite definir condiciones que deban cumplirse siempre. En caso de que la expresión booleana sea True assert no hace nada y en caso de False dispara una excepción.

Se pueden usar como pre y post condiciones en métodos, funciones, bloques de código, pero sobre todo para especificar invariantes. Es más corto que escribir una sentencia if, mucho más claro para el que lo lee y como bonus la sentencia no se ejecuta en caso de que el intérprete se invoque. Por ello cuando se coloca un “Assert False” salta una excepción *AssertionError* que es posible capturar en un *try/except*.

12) ¿Qué es Flake 8?

R/ Flake8 es una herramienta de Python que pega complementos de pep8, pyflakes, mccabe y de terceros para verificar el estilo y la calidad de algunos códigos de python.

Es un envoltorio alrededor del script McCabe de PyFlakes, pep8 y Ned Batchelder y muestra las advertencias en un archivo, salida combinada. Estos anteriores son usados para depuración del código que se está desarrollando.

Flake8 también agrega la capacidad de suprimir advertencias, enlaces de control de fuente adicionales y es extensible.