

## Tarea 1

### Preguntas Teóricas (24 pts, 2pts c/u)

**1) ¿Qué es Git?**

R/ Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.

**2) ¿Qué es Github?**

R/ GitHub es una forja (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador.

**3) ¿Qué es un branch?**

R/ Un branch es una línea de desarrollo distinta de la principal. Generalmente los desarrolladores trabajan sobre el trunk del proyecto, pero en ciertas ocasiones puede ser útil crear una línea de desarrollo paralela, para esto se usa el término branch. En Git, es simplemente el apuntador a la rama local en la que tú estés en ese momento, en este caso la rama master; pues el comando git branch solamente crea una nueva rama, pero no salta a dicha rama.

**4) ¿Qué es un commit?**

R/ Git es un sistema de control de versiones, es decir, desde que activamos el control sobre la carpeta donde está nuestro código el sistema se encargará de controlar los cambios en los archivos.

Cuando queramos marcar estos cambios realizaremos lo que se conoce como commit, que consiste en describir los cambios realizados, apuntándolos en este registro.

**5) ¿Qué es la operación cherry-pick?**

R/ Git cherry-pick es un potente comando que permite que las confirmaciones arbitrarias de Git se elijan por referencia y se añadan al actual HEAD de trabajo. La ejecución de cherry-pick es el acto de elegir una confirmación de una rama y aplicarla a otra. git cherry-pick puede ser útil para deshacer cambios.

**6) ¿Qué hace el comando git checkout?**

R/ En términos de Git, "checkout" es el acto de cambiar entre diferentes versiones de una entidad objetivo. El comando git checkout opera sobre tres entidades distintas: archivos, confirmaciones y ramas. El comando git checkout te permite desplazarte entre las ramas creadas por git branch. Al ejecutar la operación de cambio de rama, se actualizan los archivos en el directorio de trabajo para reflejar la versión almacenada en esa rama y se indica a Git que registre todas las confirmaciones nuevas en esa rama.

**7) ¿Qué hace el comando git stash?**

R/ El comando git stash almacena temporalmente (o guarda en un stash) los cambios que hayas efectuado en el código en el que estás trabajando para que puedas trabajar en otra cosa y, más tarde, regresar y volver a aplicar los cambios más tarde.

**8) Compare las operaciones git fetch y git pull**

R/ Git fetch realmente solo descarga nuevos datos de un repositorio remoto, pero no integra ninguno de estos nuevos datos en sus archivos de trabajo. Git pull, in contrast, is used with a different goal in mind: to update your current HEAD branch with the latest changes from the remote server. This means that pull not only downloads new data; it also directly integrates it into your current working copy files.

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

R/ A nivel superficial, git reset tiene un comportamiento similar a git checkout. Mientras que git checkout solo opera en el puntero de referencia HEAD, git reset moverá el puntero de referencia HEAD y el puntero de referencia de la rama actual.

**10) ¿Qué es Pytest?**

R/ Pytest es una librería de testing para Python. Es un framework maduro y con muchas funcionalidades, desde pruebas pequeñas hasta pruebas de gran escala como pruebas funcionales de aplicaciones y librerías.

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

R/ El comando “assert” existe en casi todos los lenguajes de programación. Ayuda a detectar problemas al principio de su programa, donde la causa es clara, en lugar de más tarde como efecto secundario de alguna otra operación. Con este comando se le está diciendo al programa que pruebe una condición e inmediatamente active un error si la condición es falsa.

**12) ¿Qué es Flake 8?**

R/ Flake8 es una herramienta que ayuda a generar un código mucho más limpio y libre de bugs. Esta en específico verifica pep8, pyflakes y complejidad circular” también ayuda a revisar si existen variables declaradas, pero no usadas, imports que no se usan, etc. Tiene una baja tasa de falsos positivos.