

Tarea 1 - Preguntas Teóricas

1) ¿Qué es Git?

Git es un programa para el control de versiones y trabajo en equipo especializado para la construcción de software. Fue creado pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando tienen un gran número de archivos de código fuente, además proporciona las herramientas para desarrollar un trabajo (software o página que implique código) en equipo de manera eficiente.

2) ¿Qué es Github?

Github es un sistema de control de repositorios de Git. Este, toma un paso más adelante que git y lo vuelve un sistema en línea con más recursos, capacidad de almacenar repositorios y funcionalidad que Git.

3) ¿Qué es un branch?

Una rama o branch es una línea alterna del tiempo en la historia del repositorio. Funciona para crear features, arreglar bugs y experimentar sin afectar la línea principal del proyecto.

La rama principal por defecto es la rama master. Con la primera confirmación de cambios que se realiza, se crea esta rama apuntando a dicha confirmación. En cada confirmación de cambios que se realice, la rama irá avanzando automáticamente.

4) ¿Qué es un commit?

Un commit es una captura de un repositorio de git, la cual se toma en algún momento. Mientras se trabaja en un proyecto, los cambios realizados no se guardan inmediatamente en el repositorio, si no localmente. Cuando se desea almacenar o

agregar los cambios al repositorio se tiene que hacer un “stage”, una vez que se realiza esto se guardan realizando un commit.

5) ¿Qué es la operación cherry-pick?

Con dicha operación se pueden tomar uno o varios commits de una rama para aplicarlo a otra sin tener que hacer un merge completo. Para usar cherry-pick lo único que se necesita saber es el commit específico a aplicar en la rama. Hay que tener en cuenta que este comando crea un nuevo commit, por lo que antes de utilizar cherry-pick no se debe tener ningún archivo modificado que no haya sido incluido en un commit.

6) ¿Qué hace el comando git checkout?

Actualiza el árbol de trabajo de manera que lo iguale a la versión en el índice u otro árbol especificado. Si no se especifica la dirección, se actualiza HEAD para que la rama especificada sea la actual.

(HEAD es la rama actual).

7) ¿Qué hace el comando git stash?

El comando git stash guarda temporalmente los cambios realizados en el código que se está trabajando para poder trabajar en otra cosa y luego regresar para aplicar los cambios más tarde. Guardar los cambios en stashes es muy útil si se tiene que cambiar rápidamente de contexto pero se está en medio de un cambio en el código y no está listo para confirmar los cambios.

8) Compare las operaciones git fetch y git pull

Fetch: recolecta ramas o etiquetas de otros repositorios junto con los datos u objetos necesarios para completar las historias. Como defecto, si una etiqueta apunta a una historia que se está recolectando también es recolectada, esto resulta en la recolección de etiquetas que apuntan a ramas de interés.

Pull: incorpora los cambios de un repositorio remoto a la rama actual. Este comando es igual a llamar a fetch seguido de merge FETCH_HEAD.

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

El git reset Head permite mover el HEAD a la versión que se requiera, además actualiza el índice con los contenidos a esta misma versión indicada. En caso de que se quiera mover el archivo de trabajo, se debe indicar con otro comando.

10) ¿Qué es Pytest?

Pytest es un sistema que facilita escribir pruebas pequeñas mas tiene la capacidad de escalar y funcionar con sistemas complejos para aplicaciones o librerías. Este sistema también reporta los resultados de dichas pruebas ya analizados por lo que facilita una mejor escritura y análisis de códigos.

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

Los “asserts” son comprobaciones que devuelven el estado Verdadero o Falso. En pytest, si una aserción falla en un método de prueba la ejecución del método se detiene ahí. El código restante en ese método de prueba no se ejecuta y pytest continuará con el siguiente.

12) ¿Qué es Flake 8?

Es un sistema que analiza el código que se escribe y busca discrepancias semánticas y recomienda soluciones. Tiene una baja tasa de falsos positivos y revisa pep8, pyflakes y complejidad cíclica.