

Prueba técnica: Training Calidad Sofka University

Luisa Feranada Restrepo

Septiembre 3 2020

Conceptos Generales

1 Defina en sus palabras qué es la calidad de software:

Un software es de calidad no solo cuando cumple correctamente la funcionalidad requerida por un cliente, sino que además su código sea sencillo, legible, fácil de modificar, coherente y homogéneo. Es decir, el programador debe escribir el código pensando en futuras modificaciones y que cuando lo retome él u otro programador no haya que destrozarse el código para hacer los cambios efectivos. También se puede hablar de calidad de un software cuando este se ejecuta de manera rápida, es decir, que no requiera recursos innecesarios y que optimice el tiempo de ejecución lo más que se pueda sin perder la simplicidad del mismo.

2 Explique la diferencia entre un sistema de control de versiones centralizado y un sistema de control de versiones distribuido.

Los sistemas de control de versiones centralizados, con la intención de que múltiples desarrolladores trabajen en un mismo proyecto, en vez de almacenar los cambios y versiones en el disco duro de los desarrolladores, estos se almacenaban en un servidor. Sin embargo, si dos personas editan el mismo archivo y se presenta un conflicto alguien debe solucionar este problema de manera manual y el desarrollo no puede continuar hasta que todos los conflictos sean resueltos y puestos a disposición del resto del equipo.

Por su parte, **los sistemas de control de versiones distribuidos** ya no trabajan con un repositorio centralizado sino que le dan a cada desarrollador una copia local de todo el proyecto, de esta manera se construye una red distribuida de repositorios, en la que cada desarrollador puede trabajar de manera aislada y tiene un mecanismo de resolución de conflictos mejor que la versión centralizada. Como no existe un repositorio central, cada desarrollador puede trabajar a su propio ritmo, almacenar los cambios a nivel local y mezclar los conflictos que se presenten solo cuando se requiera. Como cada usuario tiene una copia completa del proyecto, el riesgo por una caída del servidor, un repositorio dañado o cualquier otro tipo de pérdida de datos es mucho menor que en un sistema de control de versiones centralizada.

3 ¿Cuál es el comando utilizado en git para clonar un proyecto?:

Para clonar datos de un servidor remoto se usa el comando **git clone *url***. Con la dirección *url* se traen los archivos a dos lugares: una copia del master del proyecto va al directorio de trabajo local y todos los cambios históricos del proyecto van al repositorio local.

4 ¿Qué realizan los siguientes comandos de git?

- **git add -miClase** se añade al staging el archivo llamado 'miClase'
- **git add -A**: Al ejecutarlo añade los archivos nuevos, editados y eliminados al área de staging.
- **git status**: Al ejecutarlo mostrará los diferentes estados de los archivos en el directorio de trabajo y área de staging. Qué archivos están modificados y untracked, cuáles están tracked pero sin commit aún. también muestra algunos consejos básicos sobre cómo mover archivos entre etapas. Muestra la rama en la que se está trabajando
- **git pull**: *git fetch*: se usa para traer actualizaciones del servidor remoto y guardarlas en el repositorio local. *git merge*: se utiliza para combinar los últimos cambios del servidor remoto y nuestro directorio de trabajo. *git pull*: funciona como *git fetch* y *git merge* ejecutados al mismo tiempo.

- **git push** luego de hacer *git add* y *git commit* debemos ejecutar este comando para mandar los cambios al servidor remoto.
- **git commit -m “Hola”**: Una vez tengamos seleccionados todos los archivos que deseamos incluir en el repositorio, hacemos un *commit* con el comando *git commit*. Un *commit* guardará en estado actual de los archivos que hemos marcado, y les asignará el mensaje *Hola* que indica los cambios realizados, así como un autor y un identificador.
- **git log**: Al ejecutar *git log*, podremos ver los cambios que se han efectuado, junto con sus mensajes de *commit*

5 Describa en desarrollo de software qué es la herencia y qué es el polimorfismo

Herencia en programación orientada a objetos, es la habilidad de extender una funcionalidad existente definiendo una nueva clase que hereda una funcionalidad de una clase existente. El objetivo principal es la reutilización del código ya existente.

El polimorfismo es la posibilidad de construir varios métodos, con el mismo nombre pero con relación a la clase que pertenece cada uno y con comportamientos diferentes.