

1. calidad de software:

la calidad de software es el concepto que se define como la preocupación por la funcionalidad y eficiencia del software es decir: si el software es funcional (no presenta errores ni fallos), es fácil de manejar (intuitivo para el usuario), tiene alto grado de confiabilidad, en cuanto al manejo de la información y cumple o trata de cumplir con las normas ISO de seguridad de la información, es eficiente en cuanto al consumo de recursos principalmente tiempo y potencia.

Además de todo esto el software debe ser fácil de entender por lo que la forma de programar y las arquitecturas empleadas deben ser lo más claro posible ya que esto facilita un rápido y adecuado mantenimiento.

2. El sistema de control de versiones centralizado, se basa en realizar el control de versiones o guardar la información, en un mismo servidor único y principal, ya que es más fácil de implementar y seguro, ante posibles infiltraciones maliciosas. si el servidor central falla, falla todo el sistema

mientras que el sistema de control de versiones distribuido se basa en realizar el control de versiones o almacenamiento de la información en distintos servidores con diferente información o con la misma información en todos los servidores dependiendo de la aplicación que se tiene, esto se realiza con el fin de garantizar más seguridad y evitar posibles fallas de todo el sistema.

Si uno de los servidores falla, los demás pueden continuar trabajando con la ausencia del servidor que presenta la falla, pero existen más probabilidades de haber un robo de información.

3. git clone <https://github.com/daniloosorio/Room1.git>

donde lo que se encuentra de color rojo es un enlace al repositorio que se desea clonar de forma remota

git clone [/path/to/repository](#)

lo que se encuentra en rojo es la ruta para clonar de forma local

4. a. git add -miClase

añade el archivo miClase del directorio de trabajo al index donde se podrán hacer verificaciones de este antes de ser enviados al repositorio

b. git add -A

agrega archivos creados modificados y eliminados al index

c. git status

muestra el estado de los archivos del proyecto, si han sido modificados o no, si han sido eliminados y cuáles están en seguimiento.

d. git pull

se usa para descargar el repositorio y combinarlo con la rama del proyecto que este asignada.

e. git push

se usa para comparar los datos que existen en el repositorio local con el repositorio remoto, y subir estos archivos además de los archivos modificados.

f. git commit -m "Hola"

actualiza los datos de a subir al repositorio y añade un mensaje "hola" a los archivos modificados. esto solo se realiza al repositorio git local para enviarlos al repositorio remoto se debe usar git push

g. git log

dependiendo la rama donde se encuentre este comando, mostrará el historial de modificaciones realizadas en el proyecto.

5. herencia es un concepto que se manejan en los objetos para reutilizar código y hacerlos más compacto aprovechando las similitudes que existen entre algunos objetos para crear otros objetos similares pero con propiedades propias y también las mismas propiedades del objeto padre tal como sucede con la herencia genética. por ejemplo un planeta con propiedades como núcleo, tamaño, atmósfera. y otros con con estas mismas propiedades y además con agua, tierra, aire, vida.

El polimorfismo en objetos es usar los métodos heredados de los padres, pero en este caso son modificados o sobrescritos para realizar diferentes acciones u operaciones aun teniendo el mismo nombre del método padre, en cada objeto el método heredado puede ser diferente.

"métodos con el mismo nombre pueden hacer diferentes cosas dependiendo del objeto que lo posee"