

a. ¿Qué es un SVC (Sistema de Control de Versiones)?

Controlar las modificaciones que van teniendo los archivos, documentos, etc.
Sistema el cual lleva a cabo el registro de los cambios sobre nuestros archivos, regresando a versiones específicas , permitiéndonos revertir y comparar cambios, revisar quien hizo modificaciones ,etc.

b. ¿Qué es git?

Git es un sistema de control de versiones de código libre, escrito en C, multiplataforma creado en 2005 por Linus equipo Torvalds, desarrollado por la necesidad de tener un sistema de control de versiones eficiente para el desarrollo del Kernel de Linux. Hoy en día es el sistema de control de versiones más usado y adoptado en el mundo.

Sus 3 características, dentro de la licencia GNU son :

- libre distribución
- libre modificación
- libre uso

Otras: libre plataforma(multiplataforma).

c. ¿Por qué se habla de Sistemas GIT y gratuitos?

Surgen distribuciones(toman el núcleo, kernel del sistema git y se montan un conjunto de características en forma gráfica).

Se van modificando para que sus uso sea más sencillo, los desarrolladores crean interfaces gráficas, y, algunas veces, llegan a tener un costo(donde se cobra el uso del ambiente gráfico). Algunos tienen versiones gratuitas.

¿Qué sistemas Git gratis existen? Github, GitLab, entre otros.

d. ¿Para qué sirve el comando git?

Permite crear los cambios, este comando se combina con alguna instrucción.

e. ¿Para qué sirve el comando git init?

Inicializar un repositorio.

f. ¿Para qué sirve el comando git add?

Agregar archivos.

g. ¿Para qué sirve el comando git commit?

Confirma la instantánea preparada. El comando abrirá un editor de texto que te pedirá un mensaje para la confirmación. Una vez escrito el mensaje, guarda el archivo y cierra el editor para crear la confirmación.

h. ¿Para qué sirve el comando git push?

Sube lo que tenemos en nuestro repositorio local al servidor.

i. ¿Cuáles son los pasos a seguir para subir un proyecto a un repositorio?

```
echo "# MWtaller2" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/ollivap/MWtaller2.git
git push -u origin main
```

j. ¿Por qué git te permite trabajar colaborativamente en un proyecto?

El trabajo colaborativo es la facilidad a la hora de compartir código, ya que, simplemente subiendo código al repositorio principal, cualquier compañero que trabaje en el mismo proyecto podrá actualizar su copia local y tener esos archivos.

k. ¿Qué errores o conflictos pueden surgir al trabajar colaborativamente en un proyecto con git?

Uno de los principales es cuando en un proyecto de varias personas, puede pasar que cada uno tenga su rama de trabajo en su repositorio. Esto puede complicar las cosas si se quiere probar la compatibilidad de cambios de diferentes personas sin pasar por el repositorio principal.