

**PROGRAMACION ORIENTADO A OBJETOS**

**TRABAJO EN CASA**

COMANDOS BASICOS DE GIT



NOMBRE: PABLO ANDRES TIPAN CHANCUSIG

CURSO: SEGUNDO “A”

DOCENTE: Ing. NANCY JACHO

NRC:

BIBLIOGRAFIA: www.hostinger.es

PERIODO ACADEMICO: OCT.2018-FEB.2019

**COMANDOS BASICOS**

1. **git config --list:** Este comando nos muestra toda la información acerca de la computadora como el nombre de usuario y la dirección de correo electrónico. Esto es importante porque las confirmaciones de cambios (commits) en Git usan esta información para guardar cualquier tipo de archivo.
2. **git config --global user.name "Nombre del usuario":** Este comando nos permite crear el nombre del usuario con el que vamos a trabajar.
3. **git config --global user.email “correo electrónico**”: Este comando nos permite crear un correo electronico con el cual se identifica el sitio donde se subirán los archivos que vamos a trabajar.
4. **pwd:** nos visualiza el sitio en donde nos encontramos (ubicación).
5. **cd (nombre de la carpeta):** Este comando nos lleva a la carpeta en la que queramos trabajar.
6. **ls:** Este comando nos muestra una lista de archivos y carpetas que existen en el sitio en donde nos encontramos.
7. **git init**: Este comando se usa para crear un nuevo repertorio GIT.
8. **git add:** Este comando puede ser usado para agregar archivos al index. Para agregar solamente tenemos que escribir el código con el nombre del archivo.
9. **git status:** Este comando muestra la lista de los archivos que se han cambiado junto con los archivos que están por ser añadidos o comprometidos. En resumen, nos muestra el estado de los archivos git.
10. **clear:** este comando limpia la pantalla de códigos.
11. **Git commit –m “Nombre del archivo”:** Este comando nos permite asignar un nombre a nuestro programa o carpeta.
12. **git log:** Ejecutar este comando muestra una lista de commits en una rama junto con todos los detalles.

**COMANDOS ADICIONALES**

1. **git push**: Este es uno de los comandos más básicos. Un simple push envía los cambios que se han hecho en la rama principal de los repertorios remotos que están asociados con el directorio que está trabajando.
2. **git remote:** El comando git se usa para conectar a un repositorio remoto.
3. **git clone**: Este comando se usa con el propósito de revisar repertorios. Si el repertorio está en un servidor remoto se tiene que usar el siguiente comando “git clone alex@93.188.160.58:/path/to/repository”, Pero si estás por crear una copia local funcional del repertorio, usa el comando: git clone /path/to/repository
4. **git push**: Este es uno de los comandos más básicos. Un simple push envía los cambios que se han hecho en la rama principal de los repertorios remotos que están asociados con el directorio que está trabajando.
5. **git branch**: Este comando se usa para listar, crear o borrar ramas.
6. **git pull**: Para poder fusionar todos los cambios que se han hecho en el repositorio local trabajando.
7. **git merge**: Este comando se usa para fusionar una rama con otra rama activa.
8. **git diff:** Este comando se usa para hacer una lista de conflictos.
9. **git reset**: Para resetear el index y el directorio que está trabajando al último estado comprometido.
10. **git grep**: Este comando le permite al usuario buscar en los árboles de contenido cualquier frase o palabra.
11. **Gitk**: Este es la interfaz gráfica para un repositorio local.
12. **git instaweb**: Con este comando un servidor web puede correr interconectado con el repositorio local.
13. **git gc**: Para optimizar el repositorio por medio de una recolección de basura, que limpiara archivos innecesarios y los optimizara.
14. **git prune**: Con este comando los objetos que no tengan ningún puntero entrante serán eliminados.
15. **Git rebase**: Este comando se usa para la reaplicación de los compromisos en otra rama.
16. **git instaweb**: Con este comando un servidor web puede correr interconectado con el repositorio local.
17. **git fetch**: Este comando le permite al usuario buscar todos los objetos de un repositorio remoto que actualmente no reside en el directorio local que está trabajando.
18. **git rm**Este comando se puede usar para remover archivos del index y del directorio que está trabajando