|  |  |
| --- | --- |
| **COMANDOS GENERALES (CONSOLA)** | |
| pwd  ls  ls -l  ls-la  cd *Nombre*  cd ..  clear  mkdir nuevaCarpeta  history  echo “hola”  echo “hola” > notepad.txt  nano notepad.txt | Derectorio exacto donde estamos parados  Contenido del Directorio  Contenido más detallado  Contenidos más archivos ocultos  Moverse entre las carpetas  Ir carpeta atrás.  Limpiar consola  Crea un nuevo directorio(carpeta)  ver el historial de comandos usados  escribe “hola” en consola  escribe “hola” dentro de un .txt  permite modificar contenido del .txt (ctrl+x: salir y: guardar) |
| **AUTENTICARNOS CON GIT-GITHUB (PC PESONAL UNA VEZ)** | |
| $ git config --global user.name “nick”  $ git config --global user.email “a@g”  $ git config –-list  $ git config –-global -l | Datos de nuestro verdadero git  Verificar que se hayan registrado bien.  Ver únicamente el user y email+  Nota esto se puede modificar y reescribir para aclarar quien esta trabajando en esta pc. |
| ssh-keygen -t rsa -b 4096 -C "*your\_email@example.com*" | **Generará clave**  **Nos pide una referencia y una clave(no nos deja ver) luego repetir la clave.**  **Nos crea un la dirrecion unos archivos, abrir el .pub en un blog de notas.**  **Copiar el código SSH en nuestro github y agregarlo desde configuraciones. Listo** |
| **CLONAR UN REPOSITORIO EXISTENTE** | |
| forma1:  -Abrir gitbash  git clone <https://github.com/jonyalanoca/jonyalanoca.git>  forma2:  descargar el zip del repositorio | (no hace falta hacer git init) |
| **CONFIGURAR UN REPOSITORIO NUEVO** | |
| **Crear un repo nuevo vacío:**  -Crear el repo desde github  -copiar en gitbash  **Subir mis archivos a un repo nuevo:**  -extra plus: poner el mismo nombre de la carpeta y el repo(opcional)  -dentro de la carpeta que están los archivos y carpetas:  git init  git add .  git commit -m”primer commit”  -Crear el repo desde github  -copiar el sigiente comando |  |
| **COMANDOS DE GIT** | |
| git init  git status  git status -s  git pull  git log / git log –oneline | inicializar git (para configurar datos o para proyectos locales)  Informa cambios hechos. Si están “libres”, trakeados, comiteados o si esta listo para subir al repo  Igual, pero más comprimido.  Actualiza los datos desde el repositorio de GitHub  muestra los commits realizados en esa rama y sus detalles. |
| git commit –amend -m “nueva descripcion”  git commit –amend –no-edit | se le cambia la descripción al último commit  se agregan los cambios la ultimo commit sin crear uno nuevo (posterior a git add .) |
| git add -A  git add .  git add texto.txt  git add carpeta  git add \*.py | menor usado  mas usado (hacen lo mismo)  agregar un archivo  agregar una carpeta  agregar todos los archivos con terminaciones .py |
| **TAGS** | |
| git tag  git tag v0.1  git tag v0.1 738dsf1  git tag -d v0.1  git push --tags | Para referirnos a un commit (en vez de Em0274g4…)(descripción no es tag)  listar tags  agregar un tag al último commit  agregar un tag a un commit especifico.  Borrar un tag  Mostrarlo en git |
| **POR SI TE OLVIDAS** | |
| git add .  git commit -m “descripción”  git push origin main | agregar al stage  tomar foto (commitear)  subir al github. |
| **VOLVER EN EL TIEMPO (COMMIT ANTERIOR)** | |
| git reset –hard 523jh5  volver sin borrar… | -Volver en el tiempo (código obtenido en git log –oneline)  -Al volver borra los commits posteriores. |
| **RAMAS O BRANCH** | |
| git branch  git branch ramaNueva  git checkout ramaNueva  git checkout -b “ramaNueva”  **Combinar dos ramas**  Git marge rama1 | Lista de las ramas e informa en que rama estamos.  Creamos una Branch llamada ramaNueva  Nos movemos al ranch ramaNueva  Crea una nueva rama y nos posiciona a ella  Desde la rama que estemos ->  se combinan (rama que estemos) con rama1 |
| **CONTRIBUIR (PULL REQUEST)** | |
| 1. Hacer un fork 2. Clonar en remoto 3. Crear una branch y moverse a la branch 4. Modificar y pushear en la nueva Branch del repositorio 5. Revisar que se hayan agregado los cambios. 6. Desde GH cambiar a la rama nueva y ->contribute ->solicitar |  |
| Actualizar ropositorio local y github de un fork | |
| git remote add upstream <url repo original>  git fetch upstream  git checkout main  git rebase upstream/main  git push origin main --force | Enlazamos al repositorio original  Actalizamos el repositorio local  Forzamos un push para actualizar el repositorio de github |