## Comandos básicos

|  |  |  |
| --- | --- | --- |
| Comandos | Modo de uso | ¿Para qué sirven? |
| **rm** | rm nombreArchivo.extensión | Elimina el archivo que le indiquemos (no debemos olvidarnos de escribir la extensión). |
| **mv** | mv nombreAnterior nombreNuevo | Cambia el nombre de un archivo por el nombre nuevo que le indiquemos |
| **mv** | mv mover.txt /c/users/andrea/desktop/raiz/carpeta1 | Mover el archivo a otra carpeta |
| **clear** | clear ó ctrl+L | Limpia todo lo que hayamos escrito en el terminal. |
| **cd** | cd nombreCarpeta | Nos permite acceder a la carpeta que le indiquemos. |
| **cd..** | cd.. | Nos permite retroceder una carpeta |
| **mkdir** | mkdir nombreCarpeta | Crea una carpeta con el nombre que le indiquemos. |
| **touch** | touch nombreArchivo.extension | Crea un archivo de texto con el nombre que le indiquemos |
| **ls** | ls | muestra los archivos de la carpeta en la que  estamos ubicados |
| **ls -l** | ls -l | Para que me muestre la fecha y más detalles |
| **pwd** | pwd | Para saber dónde estamos ubicados en la terminal |
| **dir** | dir | muestra los archivos de la carpeta en la  que estamos ubicados. |
| **nano** | Nano nombreArchivo.txt | Nos ingresa al archivo y podemos copiar dentro de él, para guardar ctrl+o enter y salir con ctrl+x |
| **git --version** | **git --version** | Para saber la version de git que se tiene. |

**Comandos** **Repositorio**

**User.name =** andreaorozco9020

**User.email**= [andreaorozcoramirez12@gmail.com](mailto:andreaorozcoramirez12@gmail.com)

|  |  |
| --- | --- |
| **Comando** | **Explicación** |

|  |  |
| --- | --- |
| git init | crea el repositorio |
| ls -a | Muestra los archivos ocultos de git |
| git config user.name "nombre de usuario" | Registra nuestra identidad |
| git config user. email "nombreUsuario@email.com" | Registra nuestro email |
| git config –global user.name "nombre de usuario" | Lo registra para todos los archivos del computador, así solo lo tenemos que hacer una ves. |
| git add nombreArvhico.js | Agregamos este archivo al repositorio |
| git status | Para consultar el estado de los archivos dentro del repositorio. |
| git add . | Para agregar todos los repositorios que hay en la carpeta |
| git commit -m "mensaje" | Para agregar un mensaje |
| git log | Registra el historial de los commit con fecha y nombre autor |

**Comandos** **Para Subir Archivos – Bajar Archivos**

|  |  |
| --- | --- |
| **Coma**ndo | **Explicación** |

|  |  |
| --- | --- |
| git remote add origin https://github.com/andreaorozco9020/ensayo | Nos conecta con el remoto GitHub, y el link es el del repositorio donde queremos almacenar |
| git remote -v | Nos confirma que se sincronizo el repositorio local con el remoto (la nube) |
| git push origin master | envía los cambios al repositorio remoto (la nube) |
| git clone https://github.com/andreaorozco9020/ensayo.git | Nos permite clonar los archivos de otros, ósea descargar archivos de otra persona en nuestra computadora, o los nuestros en otra computadora. (para bajar archivos) |
| git pull origin master | Envía los cambios (de la nube) al repositorio remoto (el computador) (bajar los cambios del remoto al local) |

**RAMAS**

|  |  |
| --- | --- |
| **Coma**ndo | **Explicación** |

|  |  |
| --- | --- |
| git branch nombreNuevaRama | crea una nueva branch(rama) |
| git checkout nombreNuevaRama | cambia a una branch existente, se cambia a nombreNuevaRama |
| git push -u origin nombreNuevaRama | Guarda la nueva rama en el repositorio remoto, sincroniza |
| git branch -a | Para verificar las ramas que ya tenemos sincronizadas |
| git merge NombreRama | Fusiona la copia y la original, eso quiere decir que, si estoy en master y quiero traerme los archivos de NombreRama, hago esto. |