**Manera de usar git**

Hay dos maneras de usar git, con una terminal o con un programa grafico (entiéndase gráfico, un programa que tiene interfaz y no es solo texto como en la terminal).

De cualquier manera que decidamos usar git, necesitamos tenerlo instalado, lo puedes descargar de [aquí](https://git-scm.com/), si deseamos usar el instalador gráfico de [GitHub](Manera%20de%20usar%20git) , hay mas opciones, otra que conozco (aunque es de pago ) [Gitkraken](https://www.gitkraken.com/).

**Escogiendo un editor de texto.**

Git necesita de un editor de texto, para hacer algunas cosas, entre los más recomendados son [Atom](https://atom.io/),[visual studio code](https://code.visualstudio.com/),[notepad++](https://notepad-plus-plus.org/downloads/),etc, todos los puedes descargar de manera gratuita, personalmente yo usare visual studio code como editor predeterminado,

**\*\*nota:** todas las instalaciones las veremos con calma, si es que quieres instalar algo por tu cuenta, adelante

**working directory:** el folder/carpeta donde estamos trabajando

repositorio: la carpeta “.git”

**staging Area**: Donde se guardan los cambios, antes de hacer un commit

**HEAD**: es un “apuntador” que nos dice nuestra localización actual en nuestro repositorio,

este usualmente apunta a branches.

**git config user.name:** Revisa si tienes un usuario

**git config --global user.name 'name':** asociara el usuario “name” con tu trabajo en git

**git config user.email:** Revisa si tienes un email

**git config --global user.email 'email'**: asociara el email con tu trabajo que hagas en git

|  |  |
| --- | --- |
| Comando | Descripción |
| **git init** | Crea un repositorio vacío de git o reinicializa uno existente |
| **git status** | nos da información del estado de un repositorio de git y sus documentos |
| **git add** | añade los archivos al staging area |
| **git log** | muestra los commits |
| **git log –oneline** |  |
| **git commit [-m]** | hace commit de todos los cambios en el staging area |
| **git commit --amend [-m]** | rehace el mensaje del commit anterior |
| **git branch [-v]** | muestra las branchs existentes , -v muestra más información |
| **git branch <branch-name>** | crea una nueva branch desde el HEAD actual |
| **git branch --delete [-d]** | Elimina la branch |
| **git branch -D** | shortcut para –delete –force |
| **git branch -M <branch-name>** | cambia el nombre de la current branch |
|  |  |
|  |  |
|  |  |