Guía de GIT.

# Configuración de nuestro GIT.

Todo lo que hagamos en GIT debe estar asociado a alguien

* Así podremos saber quien ha cambiado las cosas y que además cada uno tendrá una copia.

Lo mínimo que se necesita para identificarnos es tener:

* Un usuario.
* Un email

## git config –global …

Sirve para establecer la configuración global de nuestro git. Así toda persona que entre en mi usuario y en mi equipo podrá acceder a mi configuración y a mis proyectos.

Al utilizar estos comandos se creara un fichero de texto llamado *.git config*

### git config –global user.name “nombre”

Para determinar nuestro nombre de usuario

### git config –global user.mail

## Crear un alias

Si nosotros quisiéramos configurar un comando nuevo para no tener que estar repitiendo comandos largos deberemos usar el alias

Este comando lo que hace es escribir en nuestra configuración(ya sea global o local) un comando con un alias definido por nosotros, por ejemplo:

# Guía básica de Git.

El primer paso para crear una carpeta de GIT es colocar nuestro terminal en la carpeta donde se guardarán los archivos.

Para ello utilizaremos comandos básicos de terminal como:

* ls/dir(list para Mac y directories para Windows): Para ver la lista d ellos directorios y archivos de la carpeta actual
* cd+ “nombre de la carpeta”(change directory): Para entrar en la carpeta deseada
* cd ..: para subir un directorio.
* cd ../.. para ir a nuestro home
* Mkdir(make directory): Para crear una carpeta

Si queremos volver a acceder a un repositorio tendremos que usar los comando básicos para llegar a la carpeta y ya alli podremos usar nuestros comandos. Para asegurarnos podemos usar -git log y -git status

## git init

Estando ya en la carpeta utilizamos este comando para crear un repositorio vacío de git

## git status

Usaremos este comando para saber el estado de nuestro repositorio

## git log

Nos dará información sobre quien ha creado el repositorio y otras cosas. Nos muestra a donde apunta el head de nuestro proyecto, o donde esta el ultimo commit.

## git add “nombre del archivo”

Como podremos ver, nuestro repositorio esta vacio, deberemos introducir nuevos documentos utilizando el comando anterior, y ya al usar git status podremos ver los elementos añadidos.

Al usar git add . añadiremos todos los ficheros de la carpeta que nos encontremos.

## git commit “mensaje”

Habiendo añadido el documento, si queremos “hacer una fotografia” de lo que llevamos hasta ahora podremos usar este comando

## git checkout

## git diff

Es un comando que nos dira los cambios realizados desde el ultimo commit hasta ahora

## git reset

Es una forma para descartar(descarta pero no elimina) los cambios. Le decimos al programa que queremos volver hasta un estado anterior

## git reset –hard “ID del punto al que queremos volver”

Este realiza un reseteo mas bestia. Devolviendonos al punto que queremos ir y prácticamente eliminando los archivos eliminados.

## git reflog

Nos da el historial completo de interacciones en nuestro git

## git tag

Sirven para etiquetar los commit