TIPOS DE SISTEMAS DE CONTROL DE VERSIONES

* Local: Esta solo en el computador donde estamos, si el PC se daña se puede perder todo.
* Centralizado: Los cambios se guardan en un servidor, el servidor reparte copias a cada usuario, el problema es que si se daña el servidor se pierde todo igual.
* Distribuidos: Cada uno de los usuarios tiene una copia de todo el proyecto, si alguien lo daña puede pedir una copia a los demas, son espejos del proyecto

AREAS DE TRABAJO

git config —global user.email “<email>”

git config —global user.name “<nombre del usuario>”

git config —global color.ui true

git init

git init <a>

a: nombre de la carpeta donde del proyecto que quiero crear e inicializar

git status : Para ver el estado del repositorio

git rm --cached <a>

a: nombre del archivo que quiero borrar del stage

git rm -f <a>

a: nombre del archivo que quiero borrar definitivamente

git add <a>

a: nombre del archivo a pasar al stage

git add ./git add -A : guarda todos los archivos que esten listos para pasar al stage

git add -n <a>

a: mirar si el aechivo existe

DIFERENCIAS

ver las diferencias de la ubicacion actual con el HASH especificado

git diff <SHA1>

comparar especificando los dos puntos, se pueden ivertir el orden de comparacion

DIFERENCIAS

ver las diferencias de la ubicacion actual con el HASH especificado

git diff <SHA1>

comparar especificando los dos puntos, se pueden ivertir el orden de comparacion

git diff <SHA1><SHA1>

RESET SUAVE SOFT

resetear los cambios y dejarlos en el stage

git reset --soft <a>

a: SHA1,a partir de este punto se van a borrar los comits este SAH1 queda como head

pasar del repositorio local al area de trabajo sin pasar por el stage

git reset --mixed <a>

a: SHA1 a partir de este punto se borran los cambios y todo queda en el area de trabajo