GIT:

1. Es un sistema de control de versiones.
2. Permite almacenar revisiones diferentes de un mismo archivo y compararlas, restaurarlas y a veces mezclarlas.
3. Git no necesita de un repositorio central (REPOSITORIO CENTRAL: lugar donde se alojan archivos y sus versiones)

ESTADOS PRINCIPALES:

Git cuenta con tres estados principales

1. Modificado (modified): modificamos el archivo pero no está confirmado en el repositorio
2. Preparado (staged): marcamos un archivo modificado en su versión actual para que vaya a la próxima confirmación
3. Confirmado (commited): Los datos están almacenados de manera segura en el repositorio local.

*MODIFIED 🡪 STAGED 🡪 COMMITED*

AGREGAR Y COMMITEAR ARCHIVOS Y CAMBIOS:

La creación de un repositorio es sumamente sencilla, solamente se debe abrir el Panel De Comandos y colocar el siguiente “git init”.

Podemos corroborar que se creó verificando que se haya creado en ese directorio una carpeta con el nombre “.git”.

Para verificar el estado de nuestro repositorio ejecutamos el comando “git status”

Para agregar un archivo en un commit debemos usar el comando “git add” seguido de la ruta del archivo.

Por ejemplo: “git add compras.txt”

No se devolverá ningún resultado. Lo que hace el comando “add” es pasar el archivo a “stage”

“git add” acepta otros tipos de parámetros que nos serán útiles para trabajar con varios archivos a la vez.

Por ejemplo:

“git add \*.jpg” agregara todos los archivos cuyo nombre terminen en jpg al área de stage

“git add ruta/a/directorio” agregara todos los archivos del directorio al área de stage.

“git add .” agregara todos los archivos modificados y/o no trackeados al área de stage (UNTRACKED FILES o NO TRACKEADOS: archivos sin versionar)

El comando “git commit” confirma los cambios del área de stage y los guarda en el repositorio.

El parámetro –m es requerido y se utiliza para asociar un mensaje al commit que estemos haciendo.

Si ejecutamos ahora mismo el comando de “git status” veremos que nos indica que no hay cambios y que nuestro espacio de trabajo se encuentra “limpio”.

En caso de haber modificado más de un archivo que ya teníamos en nuestro repositorio, podemos ahorrarnos el comando “git add” ejecutando la siguiente variacion “git commit”

Por ejemplo: git commit –am “mensaje del commit”

Este comando agrega al área de stage y guarda los cambios de todos los archivos ya trackeados que hayamos modificado.

VER EL LOG:

Sirve para ver el historial de nuestro repositorio, lo ejecutamos tras el siguiente comando “git log”

A veces es necesario indicar a git que no realice seguimiento alguno sobre algunos archivos. Para estos casos contamos con el siguiente comando “.gitignore” donde podemo definir las reglas de nustro archivo a ignorar.

Gitignore: archivos que necesito que estén dentro de la carpeta, debido a que forman parte del proyecto, pero no es el armado del sitio (comandos y archivos que deben estar para armar el proyecto pero que no deben figurar en el archivo final).

Pasos para crearlo:

1. Crear un archivo txt que se llame “.gitignore”
2. Abrir un bloc de notas dentro del recién creado gitignore
3. Colocar dentro del bloc de notas el nombre del archivo y luego la extensión

Ejemplo: imagen.jpg

1. Guardar

Estos archivos se mostraran en la carpeta pero se ignoraran en el panel de control

Comandos mas importantes de git:

Git init: sirve para iniciar un proyecto

Git add .: sirve para pasar archivos a staged

Git commit –m “(descripción del cambio o de la nueva adicion)”: sirve para confimar un archivo y pasar de staged a commited

Git log: sirve para ver el historial de nuestro repositorio.

GITHUB, SERVIDOR REMOTO:

Si deseo pasar los datos que edite desde github a los archivos que tengo en la carpeta, vamos al control de comandos y insertamos el siguiente comando “git pull”, automáticamente modificara todos los archivos que tengo en la carpeta de la maquina a lo que modifique en github