1. Para crear un repositorio local es necesario inicializar en la terminal integrada. Para eso colocamos: **git init proyecto1**

Con esto permitimos un mejor control de nuestro proyecto para trabajar colectivamente y tener un seguimiento de los cambiamos que hagamos.

1. La clonación de un repositorio es importante porque nos permite acceder, colaborar y trabajar con una copia exacta del proyecto incluyendo su historial de versiones completa. Para eso se usa: **git clone** [**https://github.com/usuario/proyecto1.git**](https://github.com/usuario/proyecto1.git)
2. Algunos comandos útiles son:

* git status: Muestra los archivos modificados o en seguimiento.
* git log: Muestra el historial de commits.
* git add: Agrega archivos.
* git commit: Guarda los cambios en el historial de Git.

1. El “commit” en Git es un punto de guardado en el historial del proyecto. Registra y permite volver a versiones anteriores de este. Para eso usamos el llamado: **git commit -m "Mensaje"**

Cada “commit” debe estar con un mensaje descriptivo indicando el cambio que se realizó, teniendo así un mejor orden.

1. Al usar **“git add”** básicamente le estamos diciendo qué cambios deben guardarse en el historial. Estos siempre se tienen que guardar manualmente:

Ejemplo de llamado:

**git add archivo.html**

**git commit -m “Ultima versión del archivo .html”**

Primero llamamos que archivo deseamos guardar, y después confirmamos.

1. Para ver el historial de cambios se usa:

**git log**

Por cada “commit” permite ver qué cambios se han realizado, quién los hizo y en qué fecha.