***Profundización en GIT/GITHUB***

Cuando uno inicializa GIT (con el comando git init), suceden dos cosas especialmente:

1. Se crea un area en memoria RAM que se llama *“Staging”,* justamente por estar situada en la memoria RAM a esta area también se le conoce como: “area temporal”. Básicamente, es un area completamente desconectada que funciona en memoria RAM y es donde, al principio, vas a ir agregando los cambios que vayas haciendo   
(de tu carpeta), cambios que todavía no han sido guardados en una copia de seguridad (es decir, que todavía no están en repositorio local); importante esto, tan sólo se les está haciendo un seguimiento en un area *de paso*. Todo cambio registrado que quiera hacerle un seguimiento en el area temporal (staging area) debe hacerlo con el comando: “git add”.

2. Ya lo introducimos antes, se logra la creación de un repositorio local de la carpeta actual (local repo) que recibe por nombre: “.git”, una carpeta oculta que emula a la carpeta original y que sirve para guardar, potencialmente, copias de seguridad (commits) de todos los cambios o ediciones que estemos haciendo dentro de nuestra carpeta en cuestión. Para tener en cuenta, antes de recurrir a crear una copia de seguridad de algún cambio o edición que haya hecho... debe primero hacerle un seguimiento; es decir, llevarlo a “area temporal”; y, eventualmente, luego querrá llevar esos cambios a repositorio local (ahora sí, al crearles copias de seguridad).

Es preciso decir que, depende de nosotros qué cambios almacenar de forma definitiva en nuestro disco duro por medio de copias de seguridad llamadas “commits”... en dicha carpeta oculta llamada “.git”: nuestro repositorio local. En repositorio local no habrá nada mientras no se haya decidido qué cosas guardar en copias de seguridad.

Justamente, el comando que se encarga de crear esas copias de seguridad se ejecuta así: “git commit -m ‘’”. La *-m* podría ser considerada como un *elemento* (una extensión del comando “git commit”) que nos permite darle una referencia identificativa y explicita, por medio de un comentario o mensaje, al cambio que estamos almacenando en una copia de seguridad: básicamente nos recuerda qué fue lo que puntualmente modificamos de nuestra anterior versión.

Al ejecutar “git commit” sobre nuestros cambios, éstos pasan de estar en un “area temporal” para llegar a un “repositorio local”. Todo “commit” generado recibe un número de identificación, se llama: identificador (un nombre interno de la base de datos de Git para cada cambio), y básicamente con él es que podemos hacer todo tipo de manipulación sobre los “commits”.

Entonces, tenemos que todas las copias de seguridad que hagamos se almacenarán en repositorio local (y sólo eso). El repositorio local recibe un nombre por defecto de: “Master”. Master realmente es una rama, la rama principal de nuestro repositorio. Eventualmente necesitaría crear varias cuando esté trabajando un proyecto grande o cuando esté trabajando un proyecto en equipo.