**Mochila del viajero**

Introducción a la informática

Clase 3. GIT

# 

# **Resumen de clase:**

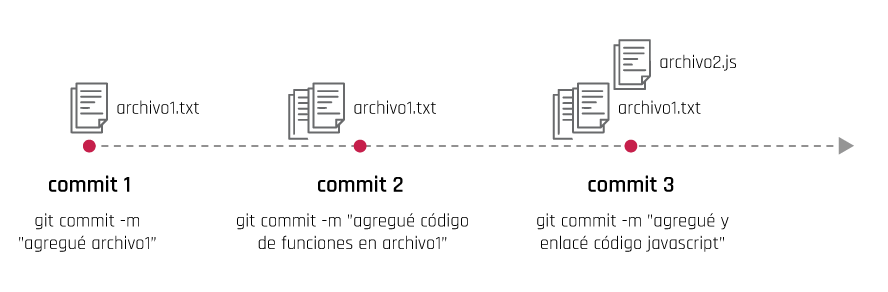
La creación de git nace de la necesidad de almacenar, mantener un backup actualizado y compartir trabajo. Este software de control de versiones nos permite tener un registro seguro de los cambios por fecha/autor de nuestro proyecto, así como también volver a versiones pasadas, compartir desarrollos y tener repositorios en la nube con la plataforma GitHub.

Para iniciar un repositorio local de nuestro proyecto, debemos seguir los siguientes comandos (parados sobre la carpeta deseada del proyecto):

**git init** --------- se crea una carpeta oculta de repositorio local .git.

**git add archivo 1.txt** --------- git ahora sabe que existe un archivo llamado archivo1.txt sobre el cual va a hacer seguimiento.

**git commit – m “agrego/modifico archivo 1.txt”**-------- confirmo que quiero ese archivo o mi modificación en el repositorio local.



Con cada **commit** creamos puntos cronológicos (guardado seguro) que nos permite volver en caso de necesitarlo.Podemos definir **commit** como paquetes que permiten ir haciendo seguimiento de los cambios y así formar el historial de versiones.

Una vez realizado un commit, se encuentra en nuestro repositorio local. Ahora podemos sincronizarlo con un repositorio remoto (GITHUB) primero le indico a cual repositorio apuntar:

**git remote add origin**  htts://github.com/usuario/repo-remoto --------- se sincronizará el local y el remoto (antes debí crearlo en GitHub y tomar la ruta).

**git remote -v** --------- para verificar si ha apuntado bien la ruta.

**Subir archivos:**

**git push origin master** --------- solicito a git que envíe los archivos de mi repositorio local al remoto.

**Bajar archivos:**

**git clone** <://ruta/>--------- opción de clonado para un repositorio remoto indicado, en una carpeta local en nuestro pc (queda sincronizado).

**Actualizar archivos:**

**git pull origin master**--------- actualiza archivos que hayan sido modificados o descarga alguno creado.

**Otros comandos de estado y configuración:**

**git status** --------- me indica el estado de git, los archivos en seguimiento o los cambios sin seguir.

**git show** --------- me muestra los cambios históricos por autor/fecha que se han hecho sobre el archivo.

**git log -p** --------- me muestra las diferencias por commits.

**git config –global user.name “nombre deseado”** --------- le configuro mi nombre de usuario.

**git config –global user.email “email deseado”** --------- le configuro mi email.

**git config --list** --------- me muestra la configuración realizada.