TRABAJO DE RECUPERACION

CLASE DE INTRODUCCION A LA INFORMATICA

ANDRES MAURICIO SANCHEZ TRILLERAS

MINI MANUAL DE GIT

PROFESOR:

CARLOS MAURICIO TOVAR

FUNDACION ESCUELA TECNOLOGICA

FET

2019

**MANUAL DE GIT:**

**COMANDOS BASICOS DE GIT:**

**GIT INIT:** Crea un repositorio nuevo en GIT. Podemos ejecutar ese comando para crear localmente un repositorio con GIT y así utilizar todo el funcionamiento que GIT ofrece.  Basta con estar ubicados dentro de la carpeta donde tenemos nuestro proyecto y ejecutar el comando.  Cuando agreguemos archivos y un commit, se va a crear el branch master por defecto.

**CONOCER LA RUTA DONDE QUIERE GUARDAR EL REPOSITORIO:**con el comando cd y la ruta donde guardo o creo el nuevo repositorio.

**COMO CLONAR UN REPOSITORIO:** Con el comando **git clone** espacio (a continuación se pega el URL del repositorio a clonar), seguido a eso damos ENTER.

**GIT ADD:el comando git add .** agrega a nuestro repositorio los archivos que hemos modificado

**GIT COMMIT: El comando git commit –m “mensaje”** hace un COMMIT (Consolidar, confirmar​ o hacer un commit se refiere, en el contexto de la ciencia de la computación y la gestión de datos, a la idea de confirmar un conjunto de cambios provisionales de forma permanente. Un uso popular es al final de una transacción de base de datos,al hacer esto guarda en el repositorio las modificaciones de los archivos de nuestro repositorio, consiguiente nos pide hacer inicio de sesión en nuestra cuenta de GTHUB.

**GIT STATUS:** El comando git status nos indica el estado del repositorio, por ejemplo: cuales archivos han sido modificados cuales no a detectado git entre otras características.

**COMO INICIAR SESION DESDE LA CONSOLA:** el comando **git config user.email “su correo”** les da la potestad para acceder a us cuenta de github, el comando **git config user.name “su usuario de git”.**

**CREAR UN BRANCH:** el comando **git branch** ( colocan el nombre de la rama a crear)

Al pulsar enter se crea una rama o branch del repositorio ya clonado.

**GIT BRANCH**. El comando **git branch** es para ver el listado de las ramas creadas,ya existentes de aquello repositorio.

**GIT CHECKOUT:** sirve para verificar y pasar de una rama a la otra **git checkout** 8nombre de la rama)le da acceso a la rama o **git checkout master** para volver a la rama principal.

**GIT PUSH:** el **git push** sirve para subir los archivos después de hacer un commit, el comando **git push origin master** es para subir los cambios al repositorio principal, el **comando git push origin (nombre de la rama)** sube los archivos de la rama o branch.

Actualiza & fusiona

Para actualizar tu repositorio local al commit más nuevo, ejecuta **git pull**

en tu directorio de trabajo para bajar y fusionar los cambios remotos.

Para fusionar otra rama a tu rama activa (por ejemplo master), utiliza **git merge <branch>**

en ambos casos git intentará fusionar automáticamente los cambios. Desafortunadamente, no siempre será posible y se podrán producir conflictos. Tú eres responsable de fusionar esos conflictos manualmente al editar los archivos mostrados por git. Después de modificarlos, necesitas marcarlos como fusionados con

**git add <filename>:a**ntes de fusionar los cambios, puedes revisarlos usando **git diff <source\_branch> <target\_branch>**

**git pull origin NombreDeBranch:** hace una actualización en nuestro branch local, desde un branch remoto que indicamos en el comando.

# Git pull request: las solicitudes de extracción o pull request. git pull request le permiten informar a otros sobre los cambios que ha introducido en una rama en un repositorio en GitHub. Una vez que se abre una solicitud de extracción, puede discutir y revisar los posibles cambios con los colaboradores y agregar confirmaciones de seguimiento antes de que sus cambios se fusionen en la rama base.