GIT

**¿Qué es Git?**

Git es una herramienta que realiza una función del control de versiones de código de forma distribuida, de la que destacamos varias características:

* Es muy potente
* Fue diseñada por Linus Torvalds
* No depende de un repositorio central
* Es software libre
* Con ella podemos mantener un historial completo de versiones
* Podemos movernos, como si tuviéramos un puntero en el tiempo, por todas las revisiones de código y desplazarnos una manera muy ágil.
* Es muy rápida.
* Tiene un sistema de trabajo con ramas que lo hace especialmente potente.
* En cuanto a la funcionalidad de las ramas, las mismas están destinadas a provocar proyectos divergentes de un proyecto principal, para hacer experimentos o para probar nuevas funcionalidades.
* Las ramas pueden tener una línea de progreso diferente de la rama principal donde está el core de nuestro desarrollo. En algún momento podemos llegar a probar algunas de esas mejoras o cambios en el código y hacer una fusión a nuestro proyecto principal, ya que todo esto lo maneja Git de una forma muy eficiente.

COMANDOS BÁSICOS DE GIT

**git help**: Muestra una lista con los comandos más utilizados en GIT.

**git init**: 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.

**git add + path**: Agrega al repositorio los archivos que indiquemos.

**git add -A**: Agrega al repositorio TODOS los archivos y carpetas que estén en nuestro proyecto, los cuales GIT no está siguiendo.

**git commit -m "mensaje" + archivos**: Hace commit a los archivos que indiquemos, de esta manera quedan guardados nuestras modificaciones.

**git commit -am "mensaje"**: Hace commit de los archivos que han sido modificados y GIT los está siguiendo.

**git checkout -b NombreDeBranch**: Crea un nuevo branch y automaticamente GIT se cambia al branch creado, clonando el branch desde donde ejecutamos el comando.

**git Branch**: Nos muestra una lista de los branches que existen en nuestro repositorio.

**git checkout NombreDeBranch**: Sirve para moverse entre branches, en este caso vamos al branch que indicamos en el comando.

**git merge NombreDeBranch:** Hace un merge entre dos branches, en este caso la dirección del merge sería entre el branch que indiquemos en el comando, y el branch donde estémos ubicados.

**git status**: Nos indica el estado del repositorio, por ejemplo cuales están modificados, cuales no están siendo seguidos por GIT, entre otras características.

**git clone URL/name.git NombreProyecto**: Clona un proyecto de git en la carpeta NombreProyecto.

**git push origin NombreDeBranch**: Luego de que hicimos un git commit, si estamos trabajando remotamente, este comando va a subir los archivos al repositorio remoto, específicamente al branch que indiquemos.

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