Los comandos básicos de Git son

**Para usar por primera vez git ay que agregar nombre y email.**

git config - -global user.email "you@example.com"- -> Comando para asignar un email.

git config - -global user.name "Your Name -> Comando para asignar un nombre de usuario.

**Para Usar git dentro de un proyecto se hace uso de los siguientes comandos.**

Git Init -> Comienza ha hacer un seguimiento del proyecto o inicia el repositorio.

Git add . -> Git hace seguimiento de todos los archivos

Git add index.html ->git hace seguimiento solo al archivo especificado.

*Nota. No hay que olvidar que antes de hacer un commit hay que agregar el archivo al área de ensayo por medio del comando git add.*

Git commit -m “mensaje” –> Git guarda una versión de los archivos.

Git staus -s -> muestra los archivos que ya tienen un seguimiento por parte de git.

Git log - -oneline -> Muestra el listado de los commit realizados, su código y quien realizo la modificación.

Git reset - -hard código del commit al que queremos regresar. Ej git reset - -hard 1cb2bec -> Sirve para regresar a un lugar en el tiempo.

Git commit –am “Mensaje” -> Sirve para agregar al área temporal los archivos cuando tiene una modificación y agregar el commit es decir en una sola instrucción ejecutamos el comando commit y add.

Modificar la descripción de un commit.

Git commit - -amend

Y se abre el editor bin si se tiene visual estudio code solo se edita el archivo.

*Nota: Si realice cambios en el buscador pero quiero que se vean reflejados en mi proyecto local se usa el comando.*

git pull -> Trae las actualizaciones echas desde github a nuestro trabajo local.

Tags: *permite especificar las versiones del proyecto.*

Git tag v# -m “Mensaje”

*Verificar que se creó la tag*

*Git log - -oneline*

*Si existe hay que subir el tag*

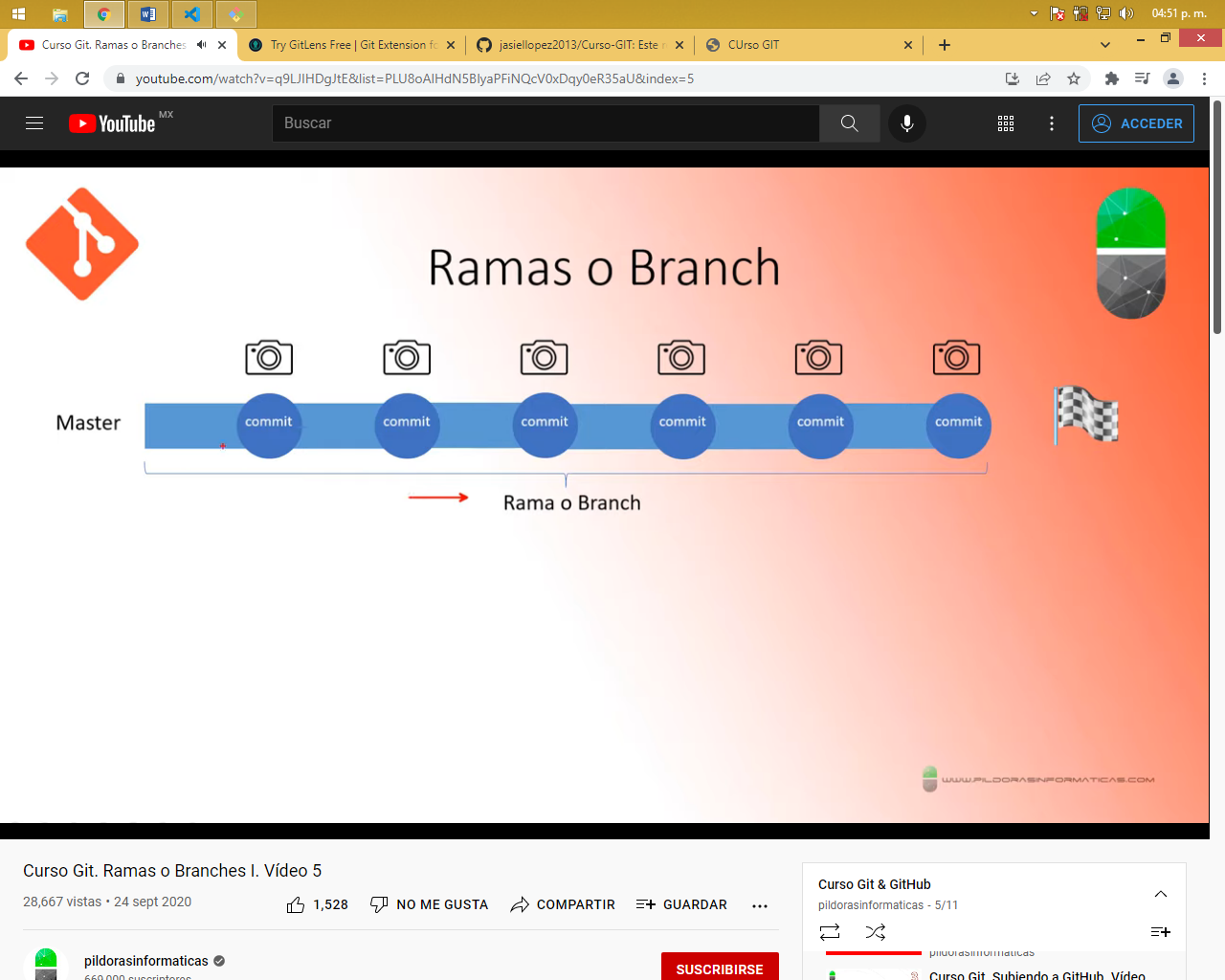
*Git push - -tags ->* Permite subir a GitHub las versiones del proyecto.

*Clonar un proyecto*

Git clone urol -> permite clonar un repositorio de manera local.

Ramas

Rama Master.



Crear rama

Git branch nombre de la rama -> permite crear una rama

Git branch -> permite visualizar el nombre de las ramas creadas y en cual estamos trabajando.

git checkout nombre de la rama -> Permite cambiar de rama para trabajar sobre ella.

Pára nir las ramas siemprese hace desde la rama master

Git merge -nombre de la rama ->que se quiere fusionar con la rama masteer.

U -> unfollow : No hay seguimientos.

A -> Aregados : Los archivos han sido agregadas al espacio de trabajo.

Eliminar una rama eso se hace con :

Git branch –d nombre de la rama -> Permite borrar la rama

FORK

Clona el código del proyecto de un programador pero este se crea en github

Permite colaborar con otros programadores para mejorar el proyecto.

(Contribuciones)

Pull & reques-> permite mandar una petición al programador original para que verifique si las novedades que ofrece el nuevo programador mejoran el proyecto y este puede o no aceptarlo.