INSTRUCCIONES PARA HACER UN REPOSITORIO CORRECTAMENTE

TERMINAL

- Dentro de nuestra terminal ejecutamos mkdir +nombre de carpeta (generalmente nombre del proyecto)

- Ingresamos un comando ls para ver que se haya creado correctamente nuestra carpeta

- ingresamos a la carpeta con el comando cd nombre de la carpeta seguida de / , le damos enter

- ingresamos el comando git init +enter

- Ingresamos el comando code . +enter (en nuestro caso usamos vs code sino seria el nombre del editor espacio mas el punto y enter)

- Una vez en nuestro vs code como buena practica se recomienda hacer un README.md por ende hacemos nuevo archivo en el nombre ponemos README.md +enter

Con un # delante de lo que deseemos comentar o explicación del proyecto o participación. Guardamos control + S en caso de no tener activado el autoguardado en vs code.

- Volvemos a la terminal Ingresamos el comando git status para ver que este todo bien y se vea en rojo los cambios que tenemos que agregar.

- Ingresamos el comando git add . ( hacemos git add espacio punto si queremos ingresar todos los archivos que tenemos en rojo, en caso contrario ingresamos por archivos ejem. Git add archivo.txt , y solo agregara ese archivo)

- Comprometemos los cambios , hacemos un comimit con el comando git commit –m “ nombre que especifique y se entienda para otros desarrolladores que vean el commit y entiendan que hicimos” (importante dejar espacio entre –m y la primer comilla doble para evitar conflictos. A veces da conflictos otras no pero tratemos de evitarlos

En la pagina web de GITHUB (no usar github desktop , entrar a [www.github.com](http://www.github.com) y loguearnos con nuestras credenciales)

- En donde esta nuestra foto. Margen superior derecho , en la flechita hacemos clic y bajamos hasta nuestros repositorios ( your repositories si esta en ingles) y hacemos clic nuevamente

- En el margen superior derecho veremos el botón verde que dice nuevo o New damos clic

- Ingresamos un nombre acorde al proyecto , una descripción que es opcional

- Hacemos clic en publico , al menos por ahora hasta saber como usar bien los archivos privados porque hay que luego otorgar permisos de escritura y demás cosas que aun yo no lo se bien por ende me limito a lo que se.

- Al hacer clic en crear repositorio se nos abrirá automáticamente una ventana con muchos comandos para ingresar y demas , de todo eso solo usaremos dos que son

git remote add origin https://github.com/xxxx/xxx.git  
git push -u origin main

* Volvemos a la terminal simpre dentro de la carpeta que creamos para el proyecto y pegamos el primer comando
* git remote add origin <https://github.com/xxxx/xxx.git>
* Git remote nos sirve para que nosotros podamos indicar si es que vamos a tener un servidor remoto en el cual nosotros podemos subir nuestros cambios y en este caso cuando le decimos add origin es que le estamos indicando de donde nosotros tenemos que ir a obtener nuestro código y también a donde tenemos que subir los cambios que realicemos, seguido de eso le damos la url donde esta almacenado nuestro código
* Damos enter e ingresamos el segundo comando
* git push -u origin main y damos enter
* Git push es el comando para que nosotros podamos subir nuestros cambios, los cuales seran con respecto a la rama a la cual nosotros estemos trabajando. Sin emabrgo nuestra rama de main no se encuentra creada en nuestro repositorio en github por lo que nosotros tenemos que indicarle que queremos crearla, para hacer eso lo hacemos con la opcion –u , seguido le indicamos en origen que la rama se llama main.
* Si tenemos configurado correctamente no nos pedirá ninguna credencial en el caso de solicitar usuario , es el usuario de la cuenta de github, si luego nos solicita la contraseña , NO ES la contraseña de usuario sino lo que debemos ingresar es un token , para generar ese token haremos lo siguiente. ( esto es en caso que nos pida, pero se supone que si esta todo bien configurado no debería pedirnos nada)
* Volvemos a github nuevamente a la foto, clic , bajamos hasta settings u ajustes según el idioma , hacemos clic y nos lleva a un apartado que dice public profile o perfil público.
* En el margen izquierdo de nuestra pantalla veremos un menú lateral y bajamos hastya la opción de developers settings u opciones de desarrollador o algo asi, clic ahí.
* Nuevamente nos despliega una pantalla con un menú lateral en el margen izquierdo , bajamos hasta personal access token o token de acceso personal o como salga en español, hacemos clic y nos desplegara una ventana , en el margen derecho arriba nos dará la opción de generate new token o generar nuevo token nos aparecerá una ventana donde tenemos que completar alguna info adicional como un nombre del token , el nombre de la pc generalemente se usa eso, luego en la expiración automática de ese repo por defecto es 30 no se toca salvo que deseen poner mas tiempo o menos, tildamos luego la casilla o check que dice repo y eso es todo, lo demas lo dejamos como esta, bajamos hasta el final y le damos clic en generar token.
* Ya ahí podemos ver nuestro token generado, luego lo copiamos y volvemos a la terminal y donde nos pedía el password o contraseña pegamos eso y enter y si salió bien podremos ver como se subió todo nuestro código a github.
* Para revisar que todo salio bien volvemos a github y actualizamos la pagina y deberíamos tener nuestro repo alli alojado.
* Cuando querramos compartir ese repo hacemos clic en el repositorio y copiamos la url de la pagina y la compartimos.
* Solo el paso de generar nuestro toj¿ken es por si no tenemos configurado lo que les dije el otro dia del CLI y demas. Si todo está configurado no nos debería pedir nada al pegar los comandos git remote add origin https://github.com/xxxx/xxx.git  
  git push -u origin main . ( donde están las x se entiende que van los datos que te genere a vos tu github)
* Cada vez que querramos modificar algo vamos a vs code hacemos lo que tengamos que hacer y luego vamos a la terminal y ponemos los comandos git add + los archivos , git commit –m “xxx” y por ultimo git push .
* En el caso que aun estemos creando una nueva rama y aun no la queremos hacer un merge( que es unir la rama secundaria ala principal , es decir ramax a rama main)
* Pero si queremos que esten en el repositorio lo que hariamos seria, cambiar de rama por ejem si estamos en rama main , creamos una rama con git checkout –b ramac entonces escribimos git push origin ramac y esto lo que hará es crear esta rama en el repositorio sin modifical la rama master o main.
* Espero esta parte haya quedado clara, todo esto es cuando trabajamos solos , con nuestra pc . El próximo paso a paso será ya para trabajar en modo colaborador.