Aprendiendo a trabajar con git.

**Herramientas que necesitaremos**

Un pc cualquiera

Programa git

Un editor de código el que desees

Y la consola de comandos sea la de Windows o la que trae por defecto git

**Temas**

Instalación

Configuración

Comandos básicos

Integración con github

**Inicio**

Primero descargamos git desde la página

La buscamos utilizando google solo colocamos git y listo

Ahora que estamos en la página descargamos la versión que corresponda a la del momento en que estés viendo este video.

Descargamos he instalamos como cualquier programa siguiente, siguiente hasta terminar nada del otro mundo.

Para trabajar con los comando de git puedes hacerlo desde la consola de Windows o puedes utilizar git bash que es la terminal que se instaló con git

Lo primero que vamos hacer es crear una carpeta en el escritorio colócale el nombre que tu desees ese será el nombre de tu proyecto

Vamos abrir esta carpeta con el editor de tu preferencia

Vamos a crear un archivo índex.html por poner el ejemplo creamos una estructura básica HTML

Si queres crear más archivos lo podes hacer pero realmente no es importante lo que nos interesa son los comando de git por eso llama curso de git no de desarrollo web tal vez más adelante lo hagamos.

Ahora vamos a la consola como te había mencionado antes utiliza la que desees yo me en mi caso utilizare la de git bash

Si colocamos **pwd** nos mostrara en donde estamos ubicados en el momento

Ahora vamos a entrar donde creamos nuestra carpeta con el comando cd en mi caso la cree en el escritorio entonces mi comando será

**Cd desktop**

Y damos enter

Luego cd y el nombre de nuestro proyecto cd proyecto y ya estamos dentro de nuestro proyecto

Ahora colocamos el comando **ls** para ver los archivos que tiene la carpeta

Ahora para inicializar la rama principal de git se hace solo cuando estamos iniciando un proyecto por primera vez colocamos el comando

**Git init** y damos enter para iniciar git o nuestro repositorio

**Git status** para conocer el estado de nuestro proyecto

Como puedes ver este comando nos muestra los archivos que tenemos pero están en rojo porque no están en seguimiento osea que git aun nos esta revisando los cambios que se ejecuten en estos archivos por eso debemos agregarlo para esto utilizamos el comando

**Git add** y el nombre del archivo que agregaremos para agregar nuestro cambio a la área de trabajo

Colocamos nuevamente git status y vemos que nuestro archivo a cambiado a color verde

Ahora vamos hacer git commit para empezar nuestro primer commit

Pero vemos que nos rechaza el commit porque pues porque primero debemos configurar git para ver quién es el que ha hecho los cambios git está hecho para trabajar en equipo por eso es im portante esta configurar para saber quién ha realizado los cambios. Para esta configuración utilizamos el comando

**Git config --global user.email ‘**[**loaizae9@gmail.com**](mailto:loaizae9@gmail.com)**’**

Y damos enter

Luego

**Git config –global user.nombre ‘edwin loaiza’**

Y damos enter

Ahora si estamos listos para iniciar en forma

Ahora si podemos crear nuestro primer commit con el comando

**Git commit**

Ahora nos cambiara la consola al modo editor si no nos deja escribir precionamos la tecla **i**

Y le damos un nombre luego precionamos escape dos punto y precionamos **wq** y enter ahora si nos ha creado el primer commit y vemos que no ha marcado los archivos con una numeración que identifica los archivos

Si presionamos el comando

**Git log**

para ver los commits que hemos creado y nos muestra un identificador de este commit nois muestra el autor y cuando fue creado

ahora si queremos deshacer cambios colocamos el comando

**git checkout – y el nombre del archivo para cambiar de rama**

hacemos

**git status** nuevamente y vemos que todo esta limpio sin nada que guardar

si modificamos nuevamente un archivo y colocamos

**git status** nuevamente veremos que hay cambios en le archivo si queremos ver esos cambios colocamos el comando

**git diff** y el nombre del archivo

y nos mostrara lo que hay de diferente en los archivos

ahora agregamos esos nuevos cambios con el comando

**git add** y el nombre del archivo

ahora nuestro archivo está en el espacio de trabajo si queremos guárdalo debemos crear un nuevo commit

**git commit**

**i**

**colocamos los cambios o un nombre**

**:**

**Wq**

Y enter

**Git status** y todo limpio

**Git log** y vemos el nuevo commit

Ahora resulta que en mi proyecto tengo archivos y carpetas con ejemplos o cualquier cosa que no quiero agregar

Lo que are será crea un archivo en mi proyecto con el nombre **.gitignore** y aquí colocare el nombre de las carpetas o archivos que no quiero incluir en mis commits pero oviamente este archivo si lo debo agregar a mis commits

**Git commit**

**I**

**Escape**

**:**

**Wq**

**Git status** nuestro espacio de trabajo limpio

**Git log** vemos nuestros commits

Ahora para evitar entrar al editor y realizar los commits directamente colocamos el comando

**Git commit –m “mensaje de identificación o cambio”** y listo

Ahora resulta que hasta el momento venimos trabajando sobre la rama principal y la verda no es lo adecuado entonces colocamos el comando

**Git branch para ver las ramas que tenemos** y me muestra la línea principal o master ahora colocamos el comando para crear un espejo de la rama master o una especie de copia

**Git branch y el nombre que desee me creara una nueva rama**

Ahora colocamos el comando

**Git checkout** y nombre de nuestro rama que hicimos anterior mente y nos muestra que se ha cambiado al nueva rama que creamos

Ahora colocamos

**Git branch nuevamente**

Y vemos que estamos en la línea que habíamos creado porque aparece en verde

Ahora podemos empezar a modificar esta nueva rama que hemos creado

Y agregar los cambios como antes lo hacimos pero ha esta nueva rama

Ahora una cosa como has visto veníamos agregando los archivos de 1 en uno y la verdad un proyecto de compone de muchos archivos y si siempre estamos de 1en 1 pues nos quitaría mucho tiempo y el tiempo es oro para evitar eso y agegar a nuestro espacio de trabajo todos los archivos que hallamos modificado colocamois el comando

**Git add .** y listo

Ahora comitiamos estos nuevos cambios

Git commit –m “agragado login”

Crear una nueva rama

**git checkout –b** y un nombre esto me creara una nueva rama con todos los archivos de el original es como clonar

para eliminar una rama creada

**git branch –d** y el nombre de la rama

para unir una rama con la rama principal o master primero nos cambiamos a la rama master

**git checkout master**

luego

**git merge**

crear cuenta y clave en para enlazar nuestro pc con github

Creamos una cuenta en github con los datos que la plataforma te pida luego para poder conectar nuestro pc con la plataforma debemos crear una clave ssh para esto nos vamos a seguir los siguientes pasos

**Click en nuestro perfil**

**Click en configuraciones**

**Click en claves ssh y gpg del lado izquierdo de la pantalla**

**Click en generar nueva clave y le ponemos un nombre**

**Click en la guía para generar nuevas claves**

**Click en comprobar claves existentes**

**Y seguimos la gia que nos ofrecen ahí si no tenemos claves creadas**

**Click en generar nueva clave**

**Seguimos los pasos**

Luego de haberla creado abrimos en el editor y copiamos la clave y la pegamos en el recuadro de githut donde nos la pide y listo la creamos ahora estamos listos para subir nuestros archivos.

Subir archivos a githut

Vamos en nuestro perfil de githut a repositorios y creamos uno le ponemos el nombre que deseemos y lo creamos

Luego seguimos las instrucciones que aquí nos piden

**Git remote add origin https://github.com/loaiza7030/curso-git.git**

**git push -u origin master**

tener mucho cuidado no cambiar el idioma de la pagina por que todo debe ser en ingles para que funcione

actualizar nuestro repositorio local con el de githut

**git pull origin master**

para actualizar nuestro archivo en githut

**git push**