



Taller practico GIT y GIT HUB

Nombre del estudiante : _____

1. Introduccion :

Git es un sistema de control de versiones distribuido utilizado para realizar colaboración y seguimiento del código fuente en el desarrollo de proyectos de software. Fue desarrollado por Linus Torvalds en 2005 como una alternativa al sistema de control de versiones llamado SVN. En la actualidad se reconoce como una herramienta básica de trabajo en las tareas de desarrollo de software.

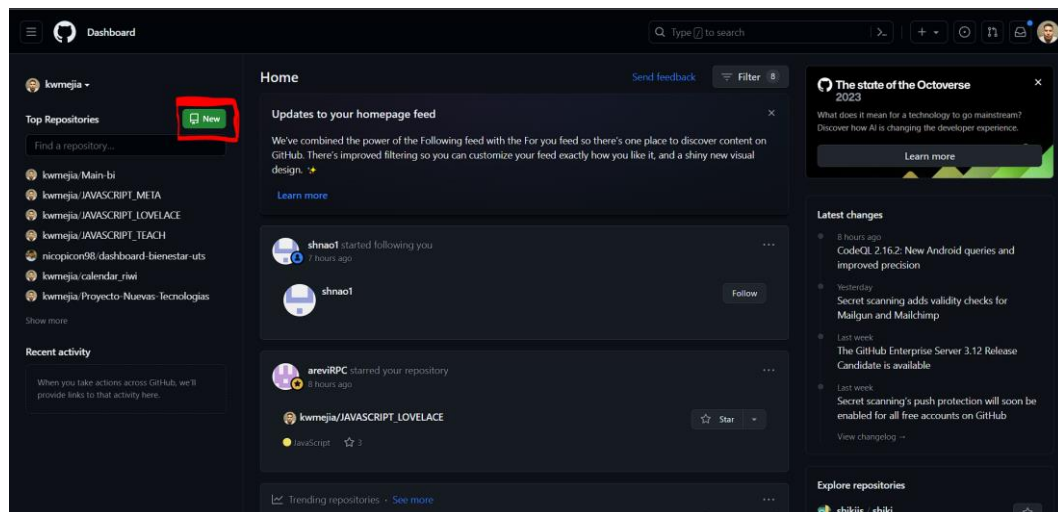
2. Requerimientos de Hardware y Software :

Para el desarrollo de este laboratorio se requiere un PC que tenga instalado el software de Git y conexión a Internet. Con estos elementos se pueden realizar todas las actividades propuestas en este documento.

2.1 Inicio del taller

El primer paso es crear una cuenta en el siguiente enlace: crea una cuenta aquí. Si ya tienes una cuenta, ingresa con tu correo electrónico y contraseña.

Una vez se ha creado la cuenta, ingresa con tus credenciales y podrás ver el siguiente panel. Deberás dar clic en el botón de new para crear un nuevo repositorio.



Una vez de clic en el botón nuevo, se podrá visualizar una vista la cual nos pedirá la respectiva información y configuración para nuestro nuevo repositorio.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Required fields are marked with an asterisk ().*

Owner *

kwmejia

Repository name *

MI_PRIMER_REPOSITORIO

MI_PRIMER_REPOSITORIO2 is available.

Great repository names are short and memorable. Need inspiration? How about [special-goggles](#) ?

Description (optional)

MI_PRIMER_REPOSITORIO

☒ Public

Anyone on the internet can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

Initialize this repository with:

☒ Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

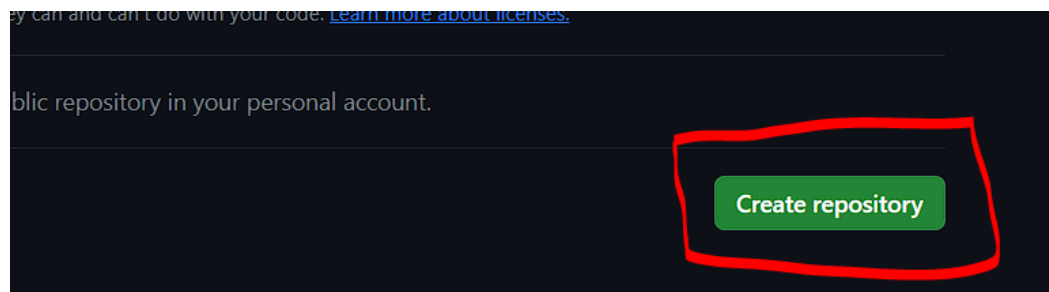
Choose a license

License: None

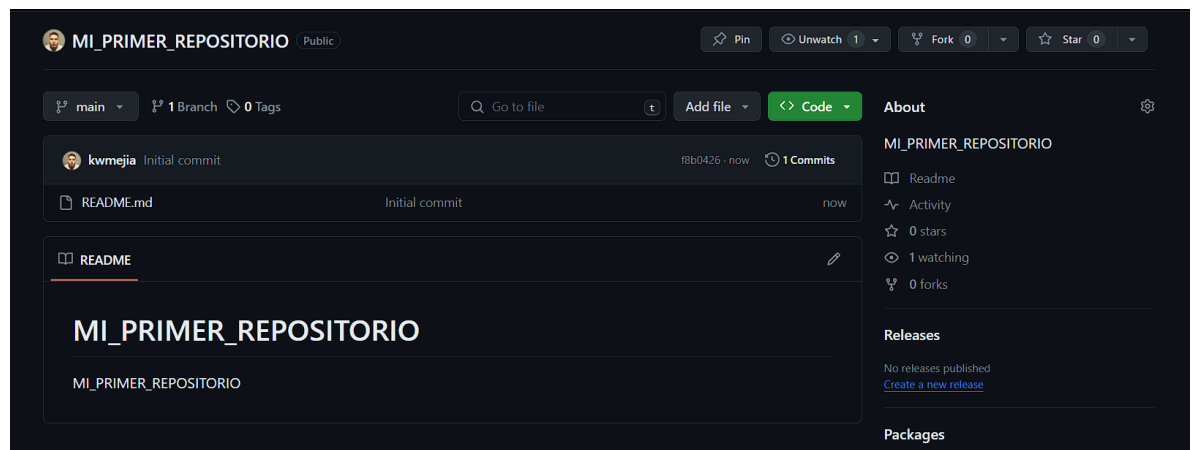


Debes marcar según tu preferencia si deseas que tu repositorio sea público o privado.

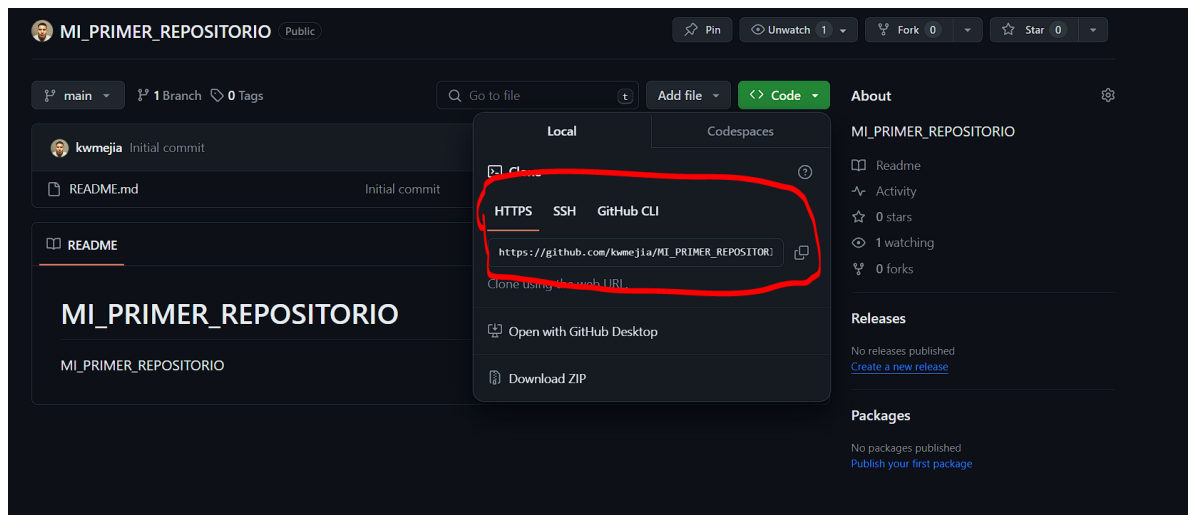
Una vez ingresados todos los datos solicitados, podemos proceder a dar clic en el botón Create Repositorio, el cual se encuentra al final de la página.



Después de dar clic en el botón de crear el repositorio, podrás observar la siguiente vista.



La siguiente imagen permite clonar el directorio y enlazar el repositorio en el pc del usuario con el repositorio en la nube, al utilizar esta opción aparece la pantalla que se puede ver en la siguiente imagen.



En esta pantalla se debe copiar la URL que esta señalada en rojo y esta será utilizada para configurar el repositorio de git en el pc del usuario.

Una vez se ha copiado la URL del repositorio en la nube se selecciona un directorio en el PC del usuario utilizando el manejador de archivo.

Finalmente debes abrir la consola de preferencia.

3. En la consola presentada por el sistema se debe utilizar el siguiente comando

- a. *git clone "url copiada del proyecto en línea en GitHub"*

Pegar la imagen de la consola con el comando

- b. Utilice el comando *git pull* y descargue el repositorio en la nube al directorio de trabajo. Coloque aquí el comando utilizado y su salida

Pegar la imagen de la consola con el comando



4. Configuración inicial de git

- a. El paso inicial en la configuración de git es la configuración de los datos del usuario
- b. Utilice los comandos para colocar el nombre del usuario, la cuenta de correo, coloque aquí los comandos para configurar esos valores.
(Verifique las diapositivas de clase y los apuntes)

Pegar la imagen de la consola con el comando

- c. Configurar git para que ignore las diferencias de final de línea entre Windows y Unix, en tu consola escribe el siguiente comando : `git config --global core.autocrlf false`

Pegar la imagen de la consola con el comando

- d. Utilice el comando `git config --global --edit` para verificar los datos de la configuración de git, coloque aquí la información de la configuración



Pegar la imagen de la consola con el comando

5. Creación de archivos y actualización de repositorio locales y en la nube
 - a. Crea una pagina web con HTML5, con diferentes secciones y paginas, inicio, contacto, nosotros, servicios, ademas de eso crea un archivo llamado credenciales.json.

Pegar la imagen de la consola con el comando

- b. Verificación y registro de archivos que han sido modificados Para determinar cuantos archivos han sido modificados se pueden utilizar el siguiente comando : `git status`

Pegar la imagen de la consola con el comando y su salida

- c. ¿Cuantos archivos hay disponibles para actualizar o para agregar al stach para tener seguimiento de git?: _____
 - d. Agregar todos los archivos al stach de git con el comando : `git add` .



Pegar la imagen de la consola con el comando y su salida

- e. Utilice el comando `git status` y verifique que los archivos están en el área de stage

Pegar la imagen de la consola con el comando y su salida

- f. Utilice el comando `commit` con el mensaje "Primera actualización del repositorio"

Pegar la imagen de la consola con el comando y su salida

- g. Indicarle a git que ignore un grupo de archivos

*Los archivos *.json son datos que serán privados, no es conveniente conservarlos o enviarlos al repositorio ya que ocupan espacio que puede ser necesario para almacenar el código.*

Para indicarle a git que ignore los archivos .class se puede realizar la siguiente actividad.

Para ignorar los archivos .class se debe colocar la información en el directorio .gitignore

Pegar la imagen de la consola con el comando y su salida



www.riwi.io



301 732 53 27



Cl. 16 # 55 - 129



Nota: No se considera necesario incluir el archivo .gitignore en el repositorio

h. Navegar en el listado de confirmaciones

Para navegar en el historial de confirmaciones (commit) se puede utilizar siguiente comando: `git log` o su variación `git log --oneline`

Pegar la imagen de la consola con el comando y su salida

Nota: Si al final del listado aparecen dos puntos y no se puede abandonarlo, coloque la letra "q" sin las comillas y presione enter

6. Copiar la información al repositorio en la nube

Tomando en consideración que se ha configurado un repositorio en la nube (GitHub) y que se han realizado varios commits entonces se puede utilizar un comando para actualizar todo en la nube

Para actualizar el repositorio remoto se utiliza el comando: `git push`

Pegar la imagen de la consola con el comando y su salida

Vaya al proyecto en la nube en gitHub y en la opción "Repository" y luego "Commits" tome una imagen del repositorio y verifique que el hash del commits es igual al del repositorio en el disco.



Pegar la imagen de la consola con el comando y su salida

7. Creación de ramas en git

Una rama en git es una línea de desarrollo independiente que se bifurca de la rama principal o de otra rama existente, cada rama puede contener cambio específicos que no se han fusionado con la rama principal o con otras ramas.

Cuando se crea una rama se realiza una copia del estado actual de la rama principal (main) u otra rama y los cambios posteriores que se hagan no afectan a la rama principal o cualquier otra rama existente.

Cuando se completa el trabajo los cambios pueden fusionarse (merge) en la rama principal u otra rama, lo que combina los cambios en ambas ramas. (Cuando hay cambios en la misma línea se presentan problemas que el usuario debe resolver)

- a. Para crear una copia de la rama principal se pueden usar los siguientes comandos : `git checkout main`
- b. Para crear la rama versión 2 se puede usar el siguiente comando : `git checkout version2`

Pegar la imagen de la consola con el comando y su salida

- c. Para verificar que la rama ha sido creado utilice el comando : `git branch`

Pegar la imagen de la consola con el comando y su salida

- d. Para realizar un cambio en los archivo de la rama recientemente creada se puede realizar adición de información a los archivos HTML crear nuevo código en los archivos, nuevas secciones de la pagina web.



Con los nuevos archivos de datos podemos actualizar el repositorio.

Utilice la secuencia de comandos necesaria para agregar los archivos modificados al repositorio.

- `git status`
- `git add .`
- `git commit -m "Modificacion de pagina web "`

Pegar la imagen de la consola con el comando y su salida

- e. Para enviar todo el repositorio a la nube incluyendo la rama que fue creada recientemente se pueden utilizar los siguientes comandos

- Primero se debe pasar a la rama main : `git checkout main`
- Segundo paso enviar todo el contenido : `git push -all`

Pegar la imagen de la consola con el comando y su salida

- f. Finalmente debes verificar en gitHub la correcta creacion de la rama local asi como el la nube.