GA7-220501096-AA1-EV01 herramientas de versionamiento (GIT) instalada y configurada.

Sergio Andres Perdomo Ortiz

Instructor técnico:

**Willington Arbey Álvarez García**

Servicio Nacional de Aprendizaje (SENA)

Centro de Servicios de Salud- Regional Antioquia

Análisis y desarrollo de software

Ficha (2758327)

**Contenido**

[1 Introducción 3](#_Toc163645191)

[2 Objetivo General 5](#_Toc163645192)

[3 Desarrollo 6](#_Toc163645193)

[3.1 Creando la Base de Datos 6](#_Toc163645194)

[3.2 Use 6](#_Toc163645195)

[3.3 Create Table 6](#_Toc163645196)

[3.4 Drop Table 7](#_Toc163645197)

[3.5 Show Tables 7](#_Toc163645198)

[3.6 Describe 7](#_Toc163645199)

[3.7 Insert Into 7](#_Toc163645200)

[3.8 Select \* from 8](#_Toc163645201)

[3.9 Update 8](#_Toc163645202)

[3.10 Insert Into 8](#_Toc163645203)

[3.11 Count(\*) 9](#_Toc163645204)

[4 Conclusión 10](#_Toc163645205)

# Introducción

En el campo de la gestión de bases de datos, el manejo eficaz de las declaraciones DDL (lenguaje de definición de datos) y DML (lenguaje de manipulación de datos) es crucial para garantizar la integridad, coherencia y eficacia de la gestión de datos. Las declaraciones DDL se utilizan para definir la estructura de la base de datos, mientras que las declaraciones DML se utilizan para manipular los datos almacenados en la base de datos.

Dominar estas habilidades y conocimientos de SQL es fundamental no sólo para los administradores de bases de datos, sino también para los desarrolladores de aplicaciones y analistas de datos que trabajan con sistemas de gestión de bases de datos relacionales. Saber y comprender cómo utilizar declaraciones DDL para crear, modificar y eliminar tablas, índices, restricciones y otros objetos de bases de datos es fundamental para diseñar una base de datos eficiente y escalable.

Por otro lado, dominar las declaraciones DML le permite realizar operaciones clave como insertar, actualizar, eliminar y consultar datos en la base de datos. Estas operaciones son fundamentales para mantener la integridad de los datos y garantizar que la base de datos refleje con precisión la información requerida por los usuarios y las aplicaciones.

En esta introducción, exploraremos las habilidades y conocimientos clave necesarios para dominar las declaraciones SQL DDL y DML, y su importancia en el desarrollo y la gestión de bases de datos relacionales. Desde la creación de tablas hasta la manipulación de datos, exploraremos las mejores prácticas, técnicas avanzadas y casos de uso comunes para aprovechar al máximo estas poderosas herramientas de administración de datos.

# Objetivo General

El objetivo general es obtener una comprensión profunda de las declaraciones SQL DDL y DML para poder diseñar, administrar y operar bases de datos relacionales de manera efectiva. Este objetivo implica desarrollar las habilidades y conocimientos necesarios para utilizar declaraciones DDL para crear y modificar estructuras de bases de datos y utilizar declaraciones DML para manipular datos almacenados en la base de datos.

Al lograr este objetivo, se espera que las personas puedan:

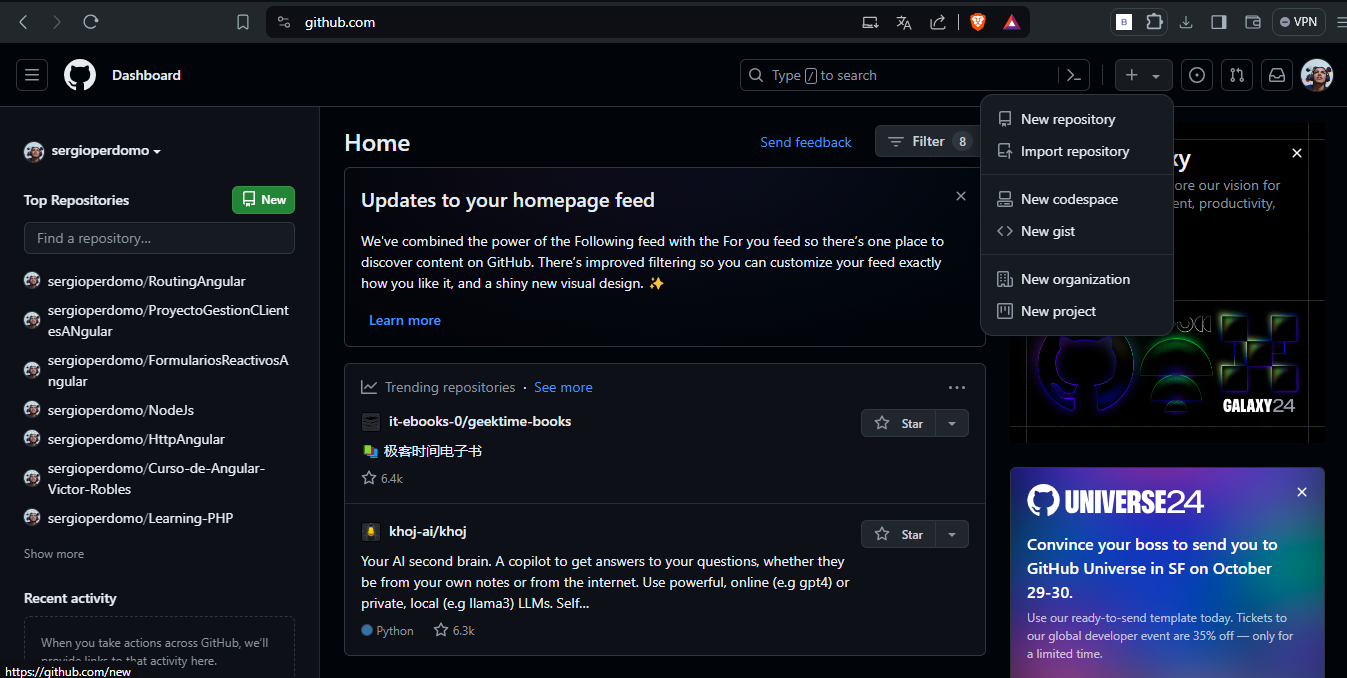
1. Comprender y aplicar correctamente declaraciones DDL para crear, modificar y eliminar tablas, índices, restricciones y otros objetos de bases de datos, asegurando un diseño de base de datos eficiente y escalable.
2. Utilice eficazmente declaraciones DML para realizar operaciones de inserción, actualización, eliminación y consulta de datos y mantener la integridad y coherencia de los datos en la base de datos.
3. Aplicar las mejores prácticas y técnicas avanzadas en el manejo de sentencias DDL y DML para optimizar el rendimiento de la base de datos, minimizar el riesgo de errores y mejorar la eficacia en la gestión de datos.
4. Aplicar los conocimientos adquiridos en situaciones prácticas, como el desarrollo de aplicaciones, la administración de sistemas de gestión de bases de datos y el análisis de datos, para satisfacer las necesidades de los usuarios y las demandas del negocio.

# Desarrollo

## Crear un nuevo repositorio público en GitHub, gitLab o herramienta de su selección con el nombre Programa-git.

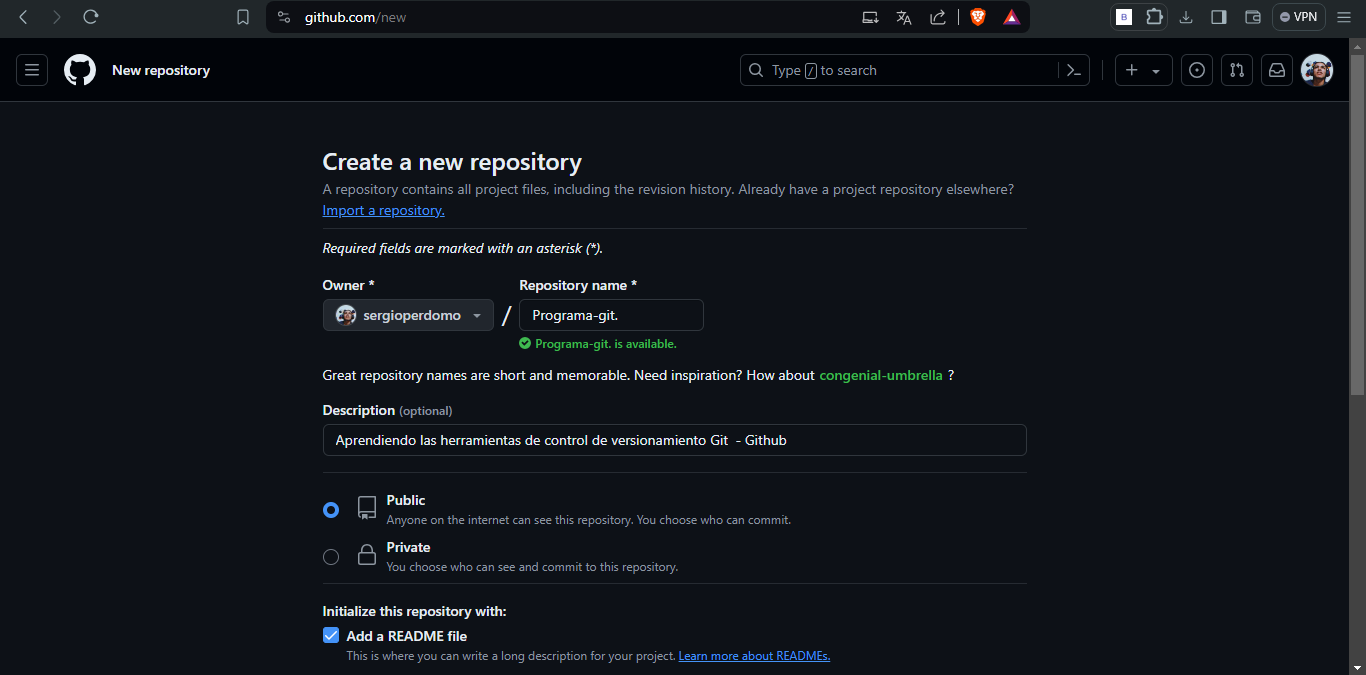
Antes de crear un repositorio en Github, tenemos que ingresar a nuestra cuenta, una vez estando en nuestra cuenta ya procederemos a crear un repositorio y para crear un repositorio existen dos maneras sencillas:

1. En la barra superior, haz clic en el botón "+" y selecciona **New repository** (Nuevo repositorio)
2. En la parte superior izquierda encontraremos un botón “**New**” das clic hay y automáticamente te redirige para que crees tu repositorio en remoto.

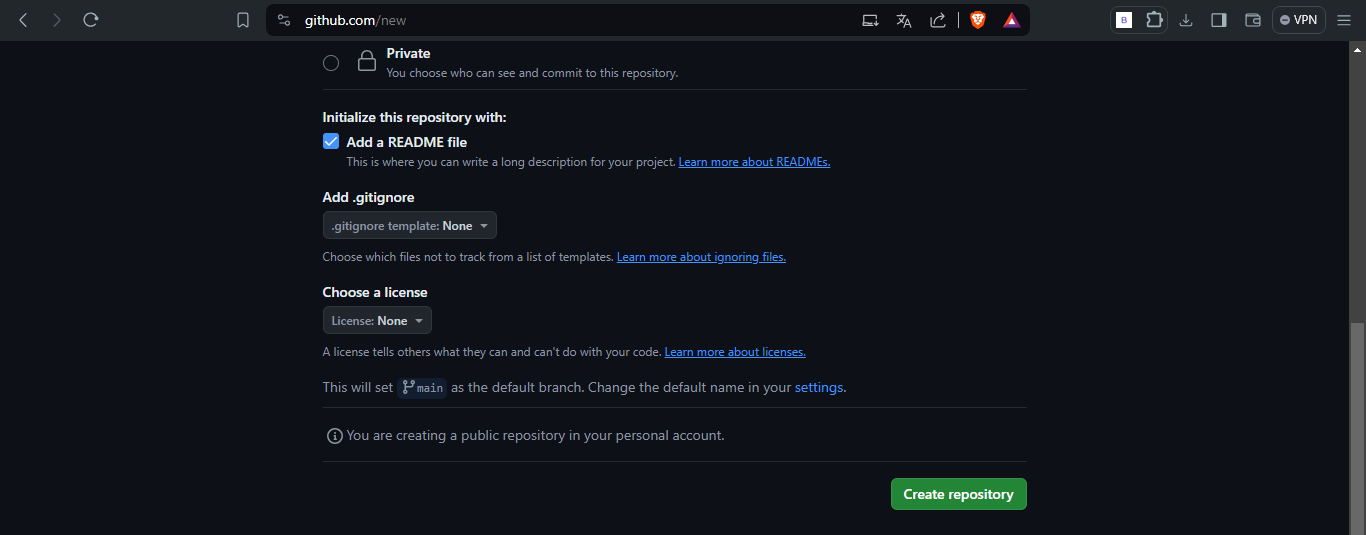


00000000000000000

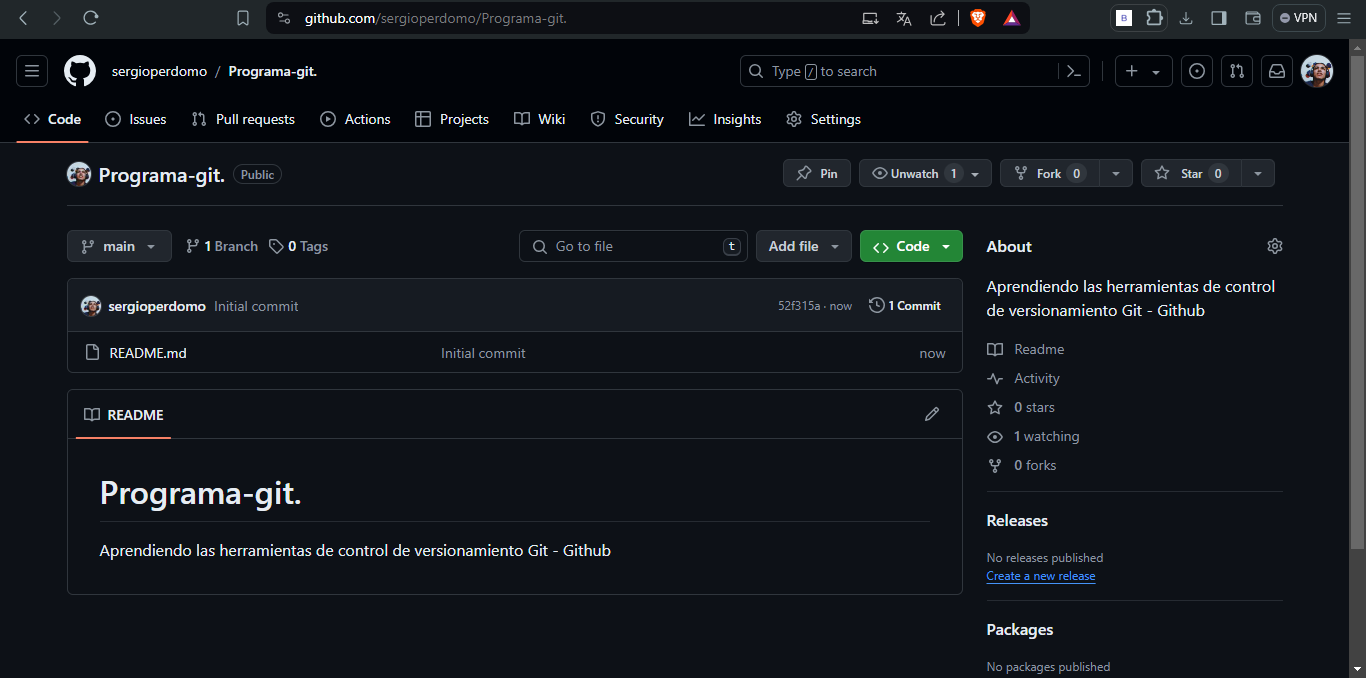
1. Llenamos los campos que se nos solicita para poder crear el repositorio,



Ya para finalizar con el proceso damos clic en **“Create repository”** (Crear repositorio)



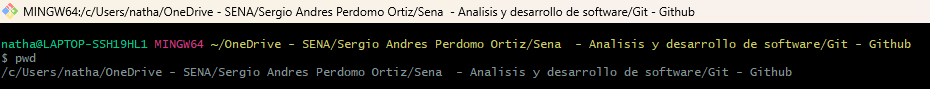
Una vez se crea el repositorio se vera de la siguiente forma:



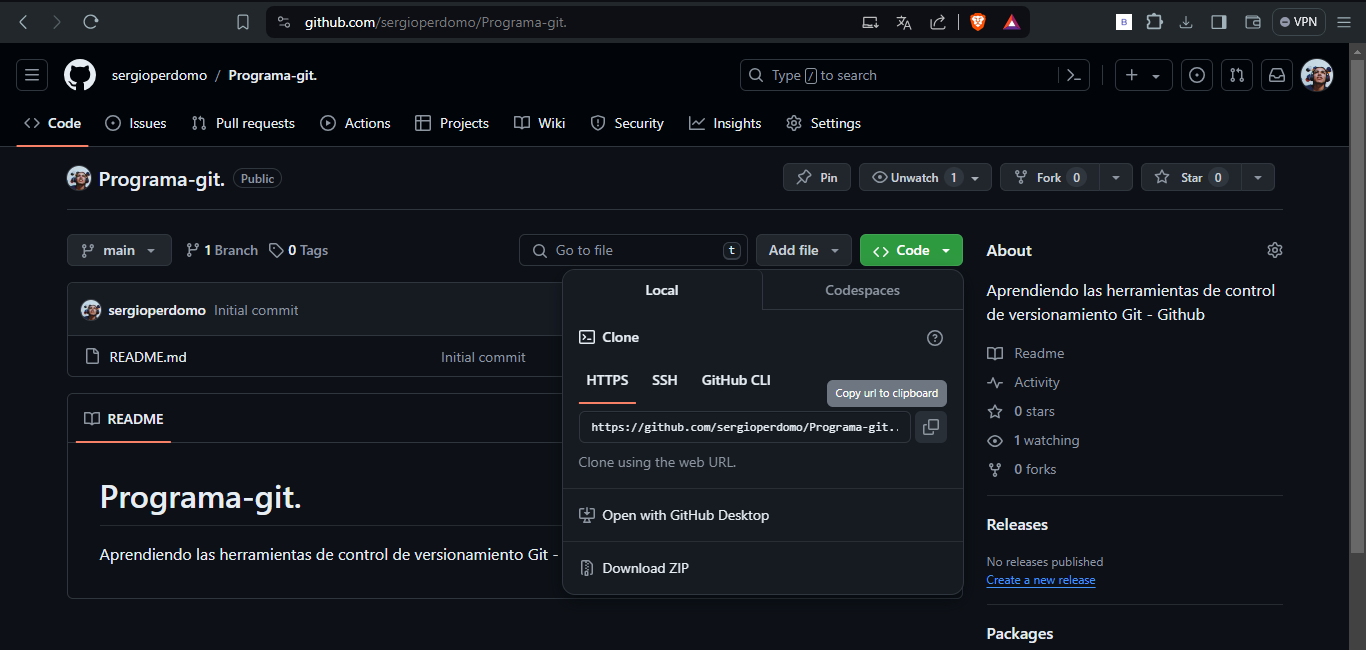
## Añadirlo al repositorio local del Programa – Uso del git clone.

En este punto básicamente lo que hice por temas de agilizar el proceso, utilice el comando **git clone** y le pase la url del repositorio el remoto.

Cabe mencionar que antes de clonarlo me dirigo a la carpeta en donde lo voy a alojar por medio de comando en **git bash**. En mi caso lo guarde en la siguiente ruta de mi ordenador:



Nos dirigimos al boton de **Code,** buscamos el apartado de HTTPS y copiamos el link que es la ruta del repositorio.

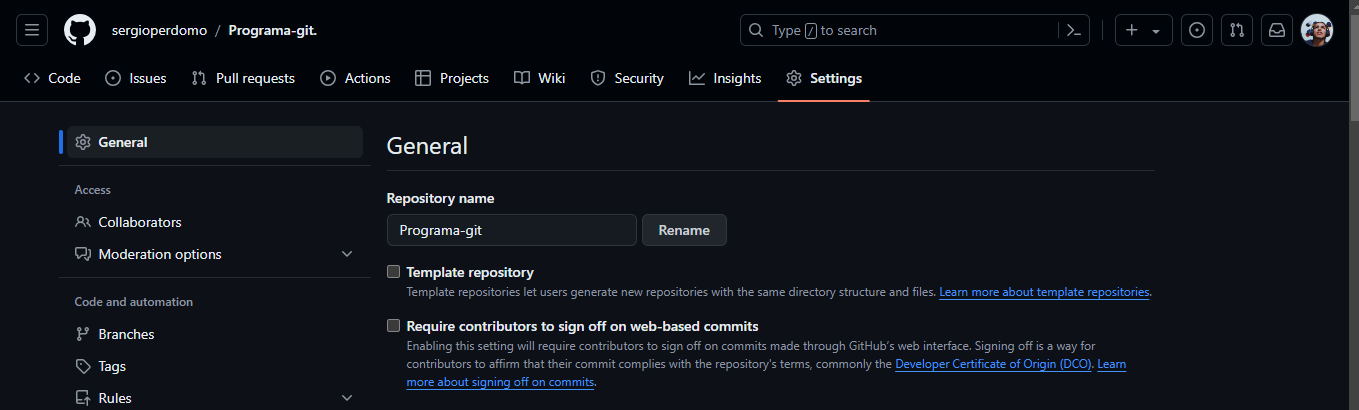


Clonando mi repositorio remoto:

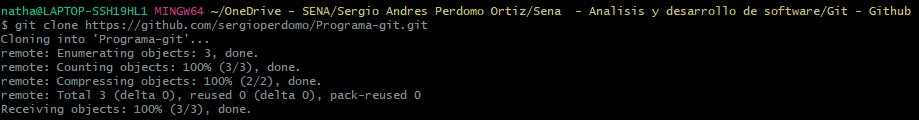
Mientras clonaba mi repositorio me apareció el siguiente error:



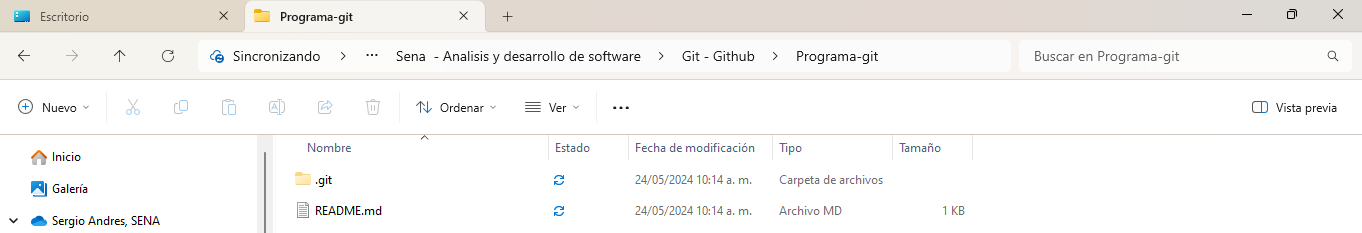
Esto se da debido a que cuando le coloqué el nombre al repositorio le puse **Programa-git.** el punto no tenia que ponerlo, por tanto, me dirigí al siguiente apartado para renombrar mi proyecto:



Una vez copiando el enlace lo clonamos.



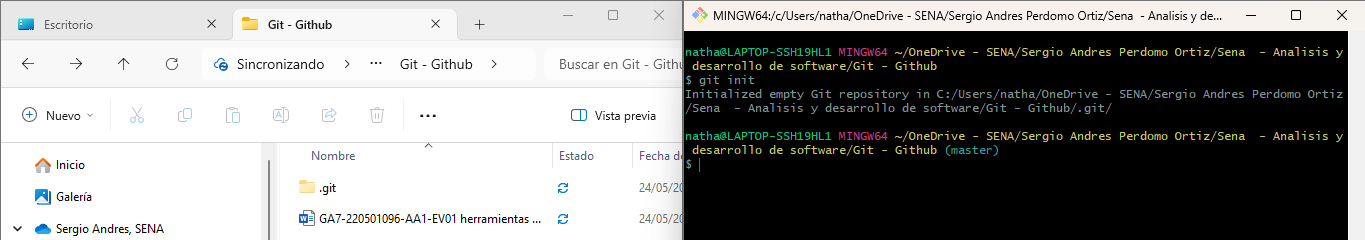
Aquí se podría observar que se clono correctamente:



## Otra forma que existe es creando el repositorio en local.

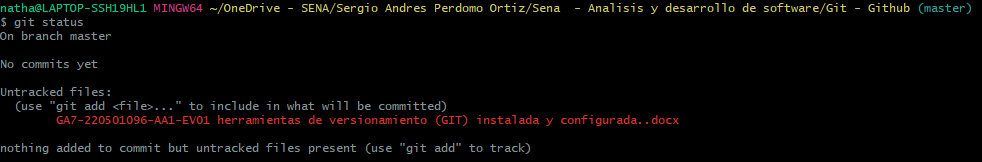
Cabe mencionar que cuando clonamos el repositorio nos omite los pasos que haremos a continuación de forma “manual”.

* Inicializamos el repositorio con el comando **git init**



## Uso de comandos en Git

* Hago uso del comando **git status** que me permite saber en que estado se encuentra mis archivos dentro del repositorio.

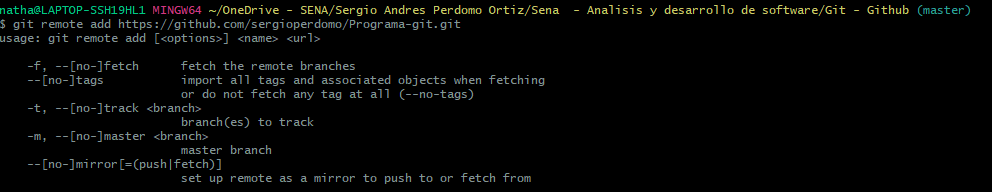


En este caso nuestro archivo GA7… Se encuentra en “Untracked Files”, en otras palabras, es un archivo que no estamos incluyendo en el control de versionamineto Git, es decir, Git no los reconoce, para que Git los reconozca debemos hacer uso del comando **git add**

* Añado los archivos al repositorio con **git add + nombre del archivo**, en caso de tener bastantes archivos utilizamos **git add .** Que básicamente lo que hace es guardarnos todos los archivos.



Me arrojaba las siguientes opciones por que le hacia falta código a ese comando, las opciones eran las siguientes:



Explicación de cada una de las opciones:

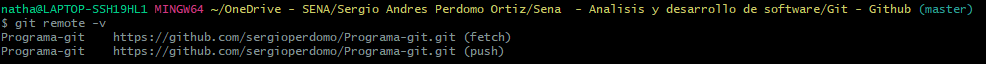
* + la opción -f fuerza la adición del repositorio remoto incluso si ya existe un repositorio remoto con el mismo nombre.
  + La opción -t especifica la rama del repositorio remoto que se debe rastrear en el repositorio local.
  + La opción -m especifica la rama maestra del repositorio remoto.

El código completo es el que aparece en el cuadro de color azul, en donde especifica que se debe poner.

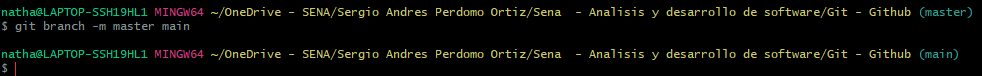
A continuación, el código:



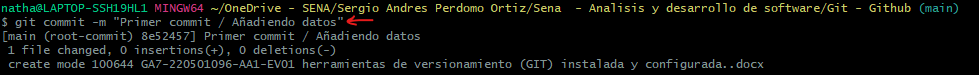
* Uso del comando **git remote -v** este comando te permite ver qué repositorios remotos están conectados a tu repositorio local y dónde se encuentran.



* Antes de hacer un commit y su debido push para poder subir el archivo al repositorio remoto, debemos cambiar el nombre de la rama (**master**), por buenas practicas se recomienda cambiarla a **main**. Para poder hacer este cambio usamos el comado **git branch -m master main** que lo que nos permite es cambiar el nombre de la rama, en donde **-m** significa mover.



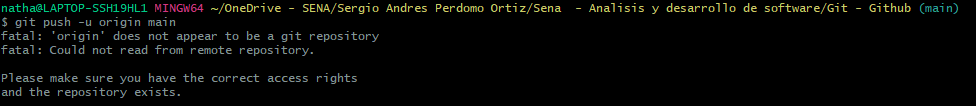
* Haciendo nuestro primer **commit –** la sintaxis es la siguiente **git commit -m “mensaje del commit”,** donde **-m** significa mensaje



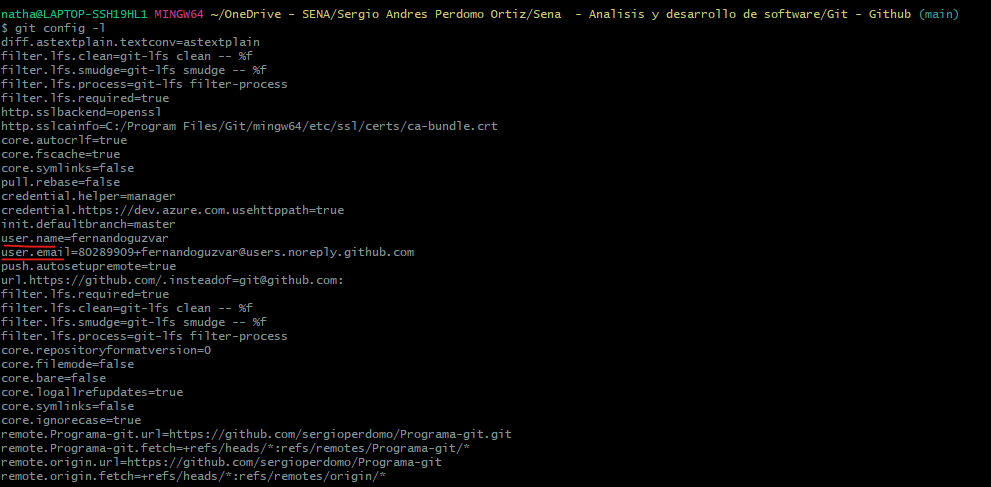
Para saber mirar cuantos commits hemos realizado podemos utilizar el comando **git log –oneline**



* Subiendo commits al repositorio remoto (Github) con el comando **git push -u origin main** en caso de que no se alla cambiado el nombre de la rama (master) pues usan este mismo comando solo que en vez de main colocan master.

Me salío el siguiente error:  
****

Este error se debe a que tenia otra configuración en mi **Git**,es decir, nombre y gmail de otra persona. Por tanto, me toco volver a colocar mi nombre y mi gmail. Como me di cuenta de que tenía otro usuario y otro gmail, fue por medio de este comando: **git config -l** y de este **git config --global --list**



Para poder cambiar el name y email use los siguientes comandos:

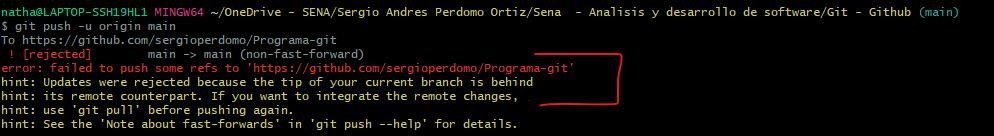
git config --global user.name "name de la persona"

git config --global user.email [gmaildelapersona@gmail.com](mailto:gmaildelapersona@gmail.com)

Use el siguiente comando para cambiar de manera global la rama que nos aparece como master a main

git config --global init.defaultBranch main

Ahora me surge otro error:



# Conclusión

El dominio de las sentencias DDL y DML de SQL es esencial en el ámbito de la gestión de bases de datos, ya que proporciona las herramientas necesarias para diseñar, crear, gestionar y manipular eficientemente datos en entornos relacionales. A lo largo de este proceso de aprendizaje, hemos explorado en detalle las capacidades y aplicaciones de estas sentencias, así como su importancia en el desarrollo de sistemas de información robustos y eficaces.

Al adquirir un dominio integral de las sentencias DDL y DML, se abre la puerta a una serie de oportunidades y posibilidades en el campo de la tecnología de la información. Desde el diseño y la implementación de bases de datos hasta el desarrollo de aplicaciones y el análisis de datos, estas destrezas y conocimientos son altamente valorados y demandados en la industria.

Es importante destacar que el aprendizaje en este campo es un proceso continuo y en constante evolución. A medida que nuevas tecnologías y tendencias emergen, es fundamental mantenerse actualizado y seguir explorando nuevas técnicas y prácticas para mejorar y optimizar los sistemas de bases de datos.

En conclusión, el dominio de las sentencias DDL y DML de SQL es un componente clave en el camino hacia la excelencia en la gestión de bases de datos. Al invertir tiempo y esfuerzo en desarrollar estas destrezas y conocimientos, se está preparando el terreno para una carrera exitosa y gratificante en el emocionante mundo de la gestión de datos.