

Universidad Mariano Gálvez de Guatemala

Análisis de Sistemas II

Ingeniería en sistemas

Esduardo Del Águila



## Manual Técnico.

Jorge Mario Marroquín Roca 0901-19-676

Yordi Daniel Hernández Pérez 0901-19-3281

Pablo Arturo López Paiz 0901-19-2375

Kevin Alejandro González Rivas 0901-19-76

Armando Paolo Sequen Salazar 0901-19-1801

Herbert Leonel Domínguez Chávez 9959-19-5644

Luis Carlos Lee Mejía 9959-19-2356

Miguel Ángel Luna Quevec 0901-10-509

## INDICE

Presentación. ....	3
Objetivo.....	4
Herramientas Utilizadas Para El Desarrollo .....	5
Conexión con Azure. ....	6
Ejecución del software área de seguridad. ....	11

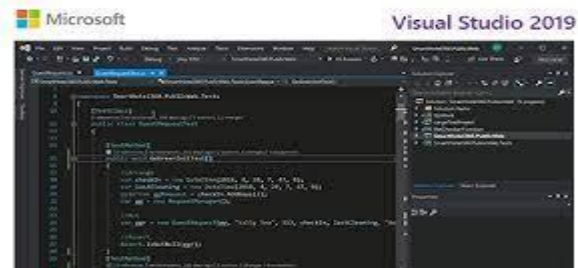



## Presentación.

El siguiente manual guiara a los usuarios que harán soporte al sistema, el cual les dará a conocer los requerimientos y la estructura para la construcción del sistema, en el desarrollo de programa de escritorio conectados mediante una base de datos en la nube, el cual muestra las herramientas necesarias para la construcción y la funcionalidad del sistema.

### Objetivo.

Dar a conocer el uso adecuado del software en aspectos técnicos de manera descriptiva e ilustrada sobre los componentes y funcionalidades que conforman el buen funcionamiento del sistema de “seguridad”.

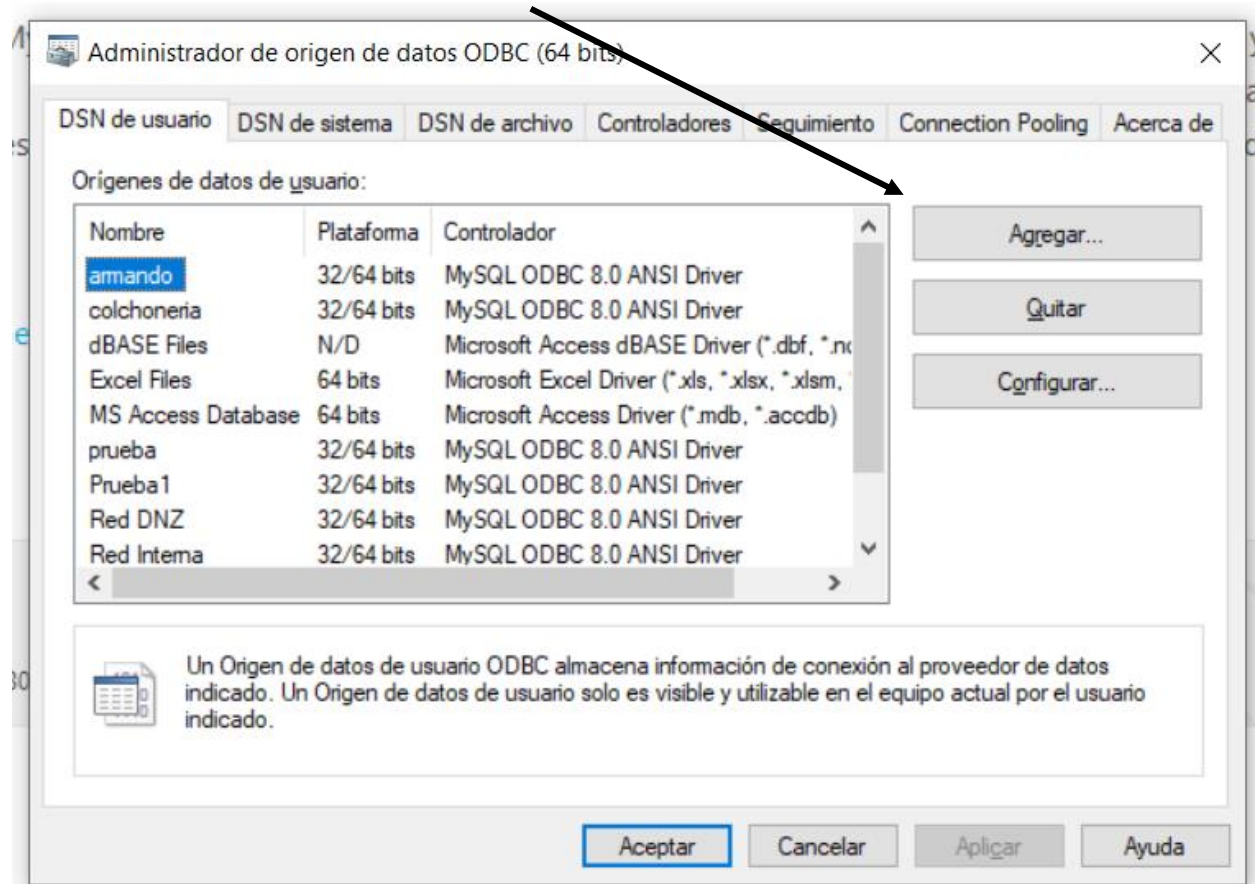
## Herramientas Utilizadas Para El Desarrollo

Herramienta	Descripción.	Imagen.
Visual Studio 2019 community.	es un entorno de desarrollo integrado, lo cual también es conocido como IDE (Integrated Development Environment).	
Workbench.	es una herramienta de diseño y modelado de base de datos visual capaz de acceder a MySQL bases de datos en servidores remotos	
Git	gestiona los diversos cambios que se realizan sobre los elementos de algún producto o una configuración.	
GitHub	permite que los desarrolladores alojen proyectos creando repositorios de forma gratuita. Pero hay que tener una cosa en mente, y es que para poder subir gratis los proyectos deberán ser de código abierto. Tal como es nuestro software en general, "colchonería".	

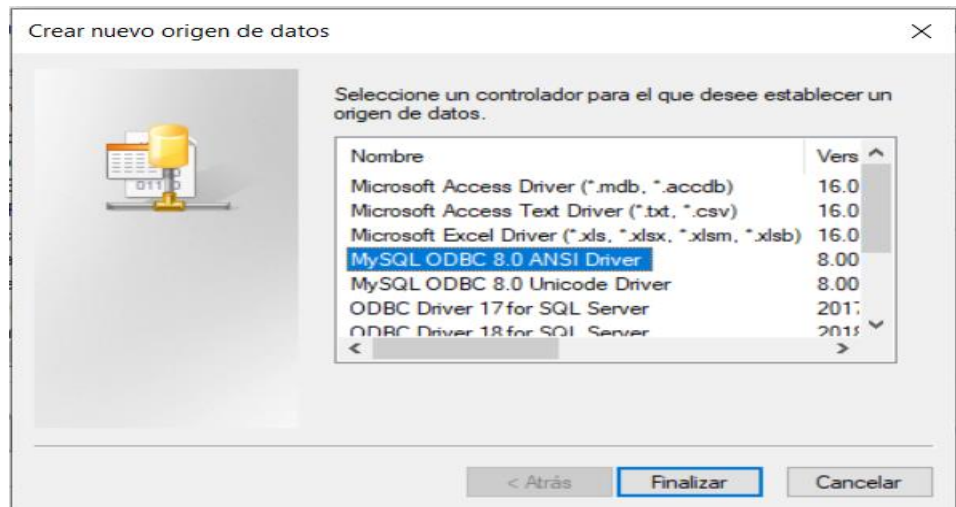
## Conexión con Azure.

Abrimos nuestro ODBC “64/32” bits.

Seleccionamos el botón de agregar.

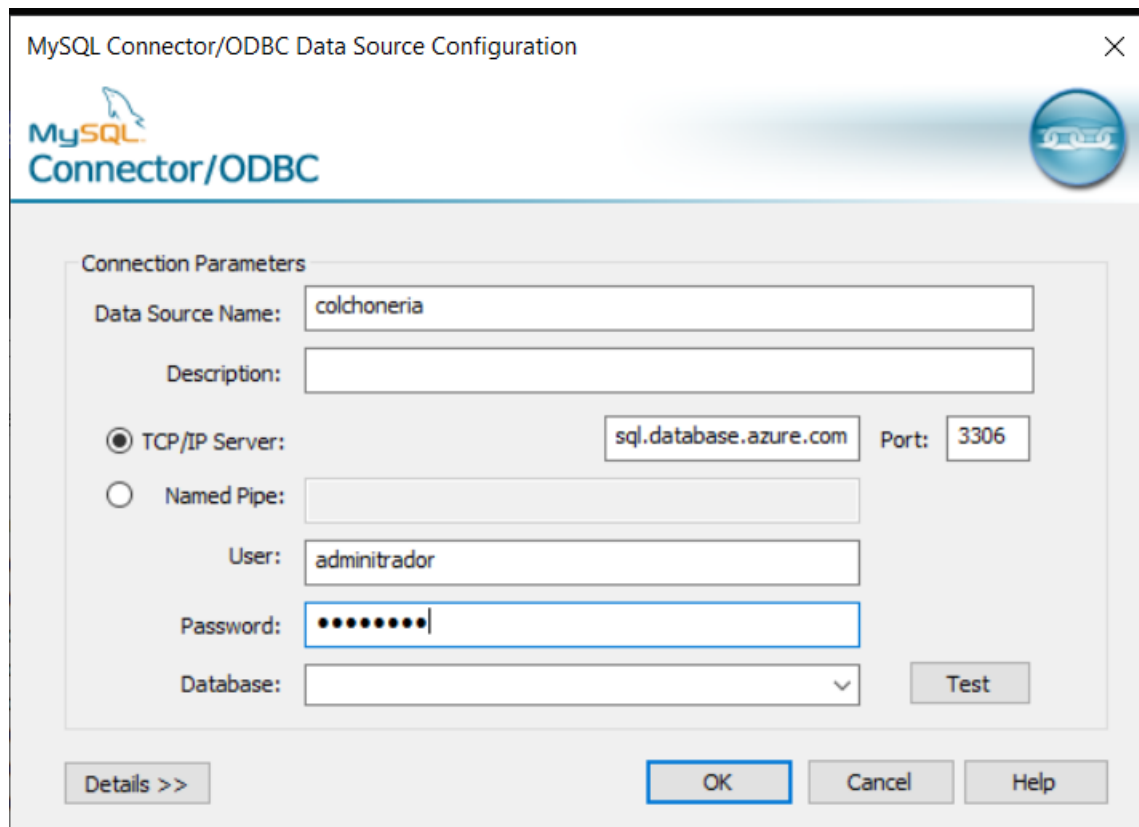


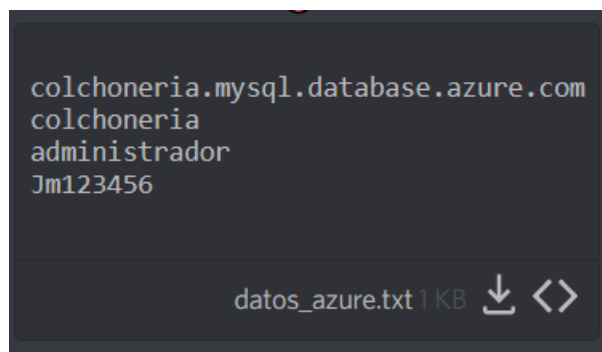
Seleccionamos la opción señalada a continuación. “ANSI DRIVER”.



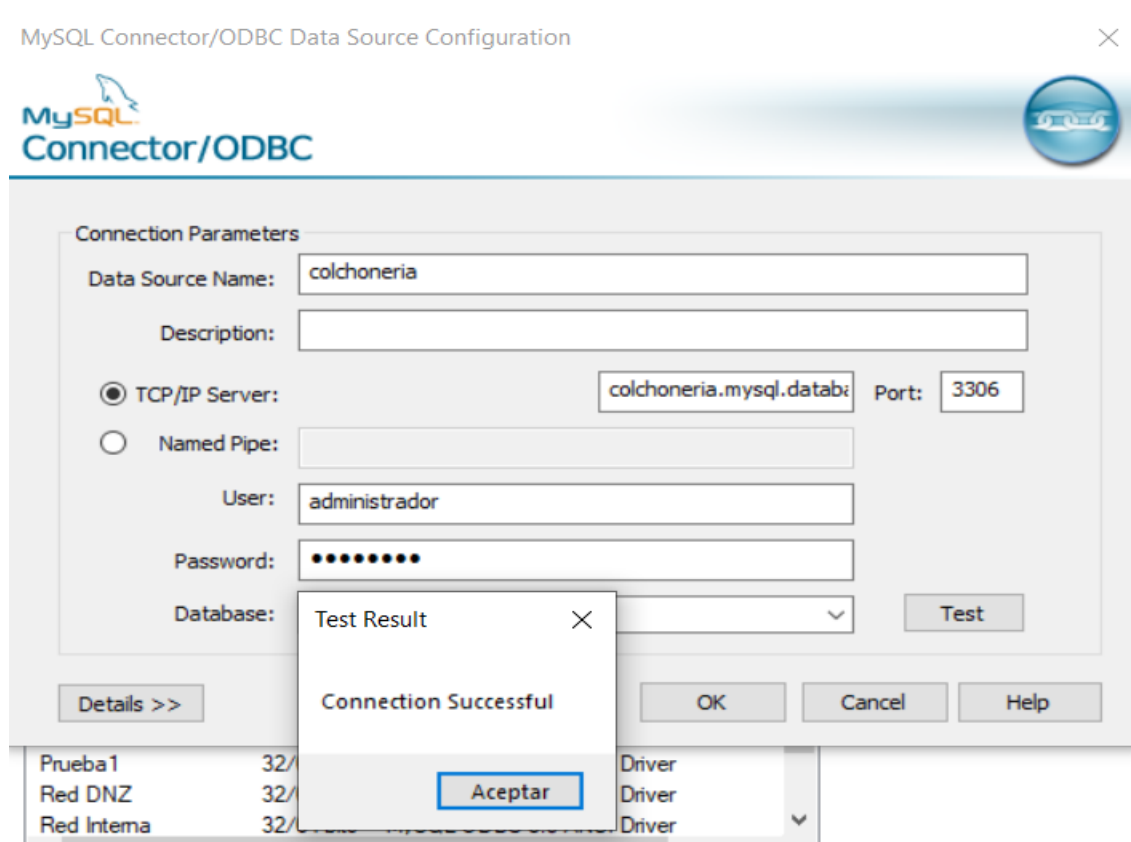
Seguido en el botón de finaliza.

La siguiente tabla la llenamos con los siguientes datos.



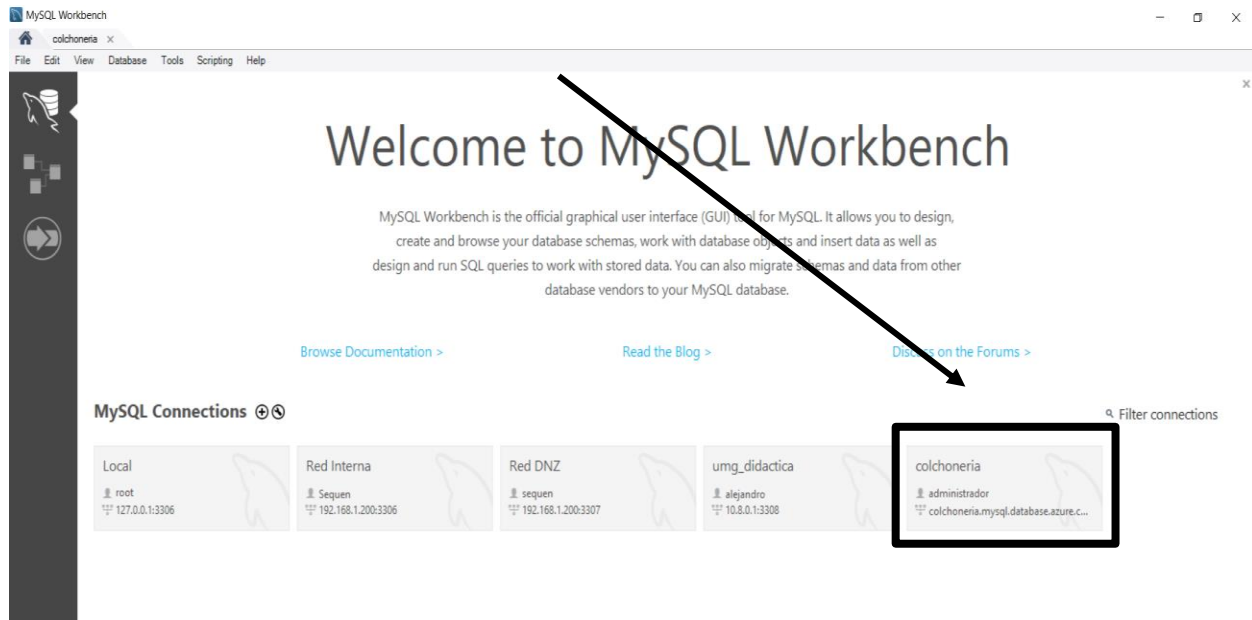


Con lo campos llenos seleccionamos el botón “test” para comprobar que la conexión ha sido exitosa.

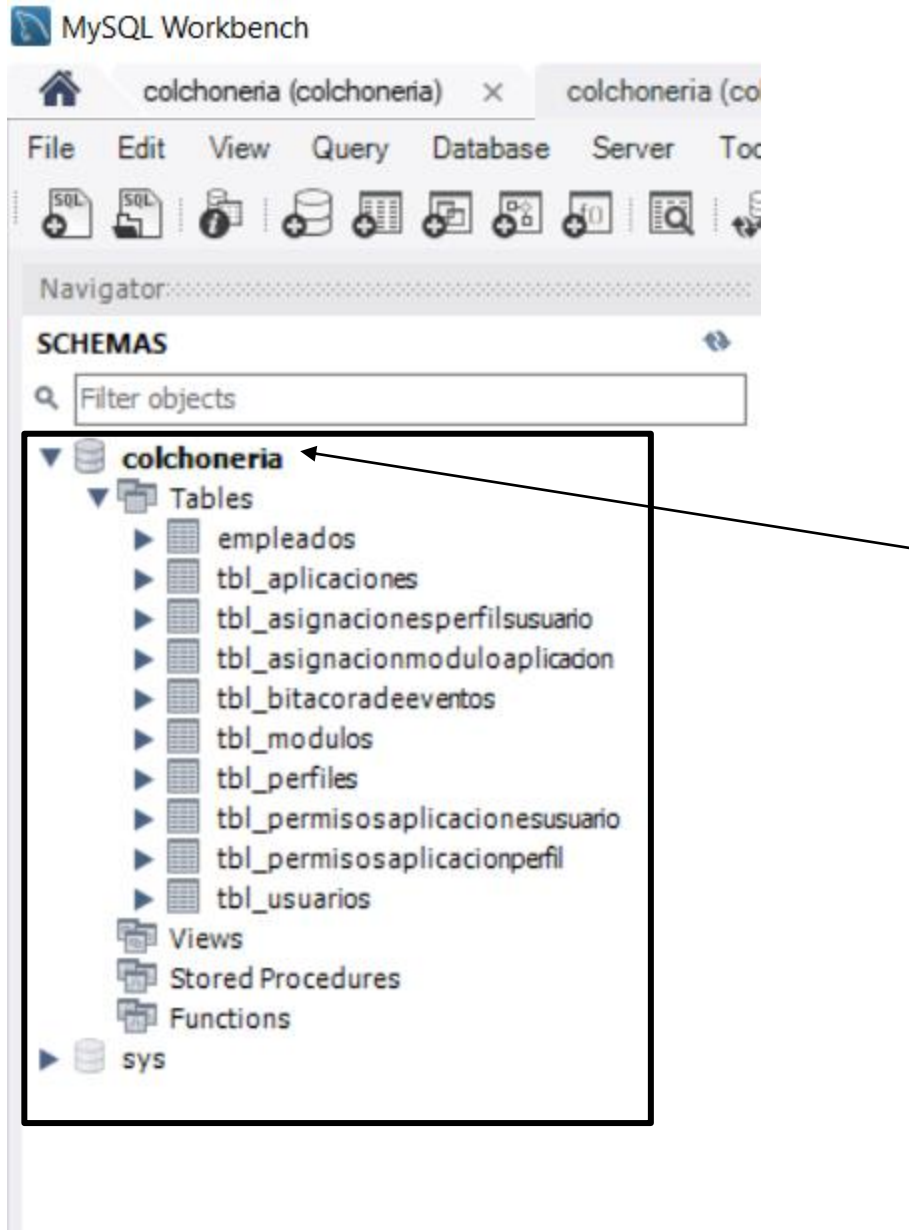




Seguido de esto nos vamos a nuestro workbench a comprobar la conexión y en inicio nos debe aparecer lo siguiente.



Seleccionamos la base de datos y nos debe aparecer lo siguiente.

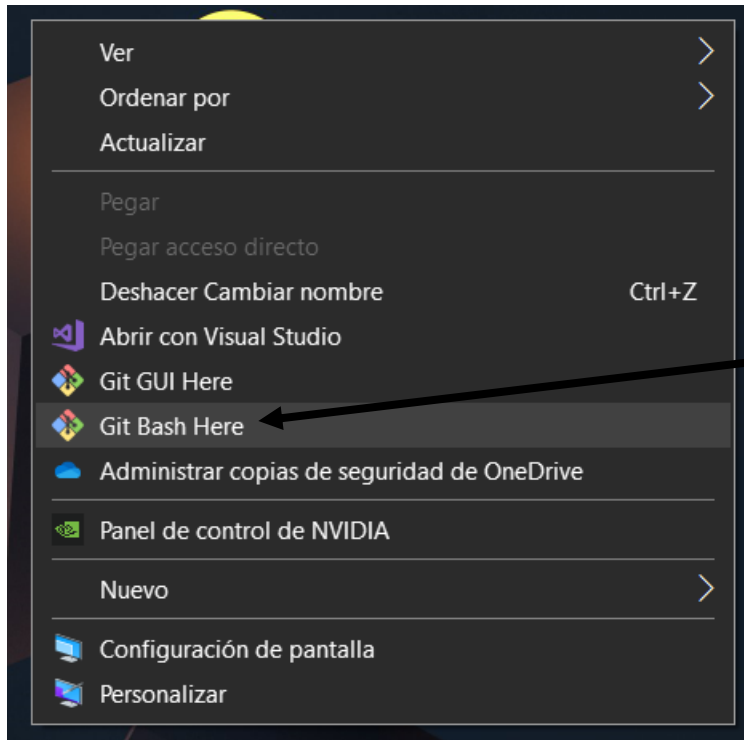


Mostrándonos la base de datos con sus respectivas tablas.

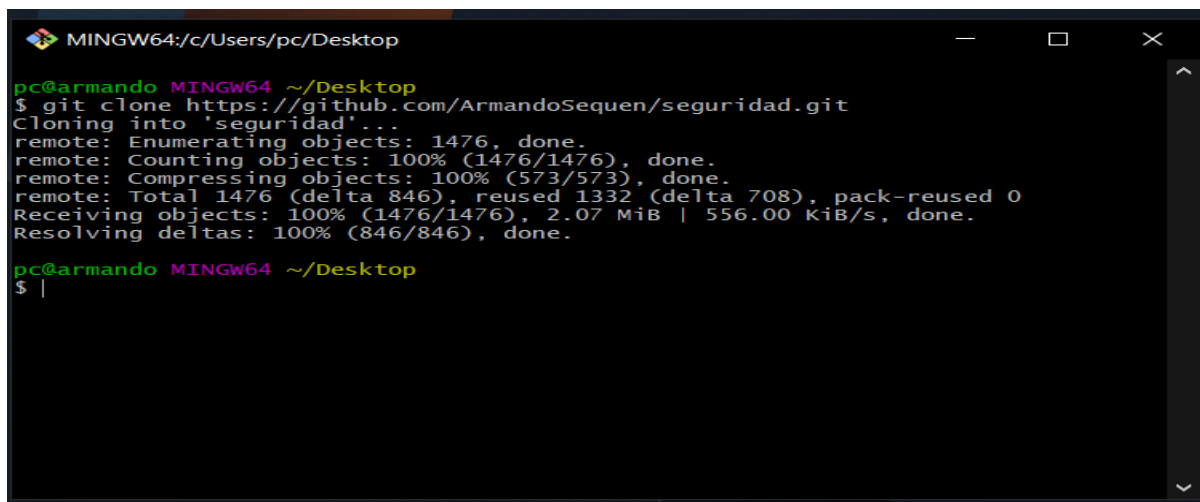
## Ejecución del software área de seguridad.

Damos click derecho sobre nuestro escritorio o bien en alguna carpeta a la que deseemos como destino de nuestro repositorio que clonaremos de GitHub.

Seleccionamos la opción de Git Bash Here.



Nos aparecerá una ventana igual a la que se mostrara a continuación.

A screenshot of a MINGW64 terminal window. The title bar reads 'MINGW64:/c/Users/pc/Desktop'. The terminal shows the execution of the command 'git clone https://github.com/ArmandoSequen/seguridad.git'. The output shows the progress of cloning the repository, including enumerating, counting, and compressing objects, and receiving the data. The terminal text is as follows:

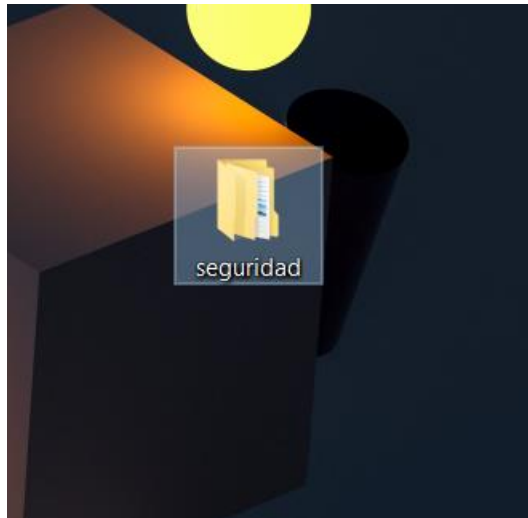
```
pc@armando MINGW64 ~/Desktop
$ git clone https://github.com/ArmandoSequen/seguridad.git
Cloning into 'seguridad'...
remote: Enumerating objects: 1476, done.
remote: Counting objects: 100% (1476/1476), done.
remote: Compressing objects: 100% (573/573), done.
remote: Total 1476 (delta 846), reused 1332 (delta 708), pack-reused 0
Receiving objects: 100% (1476/1476), 2.07 MiB | 556.00 KiB/s, done.
Resolving deltas: 100% (846/846), done.
pc@armando MINGW64 ~/Desktop
$ |
```

En la cual escribiremos el siguiente comando: git clone "link"

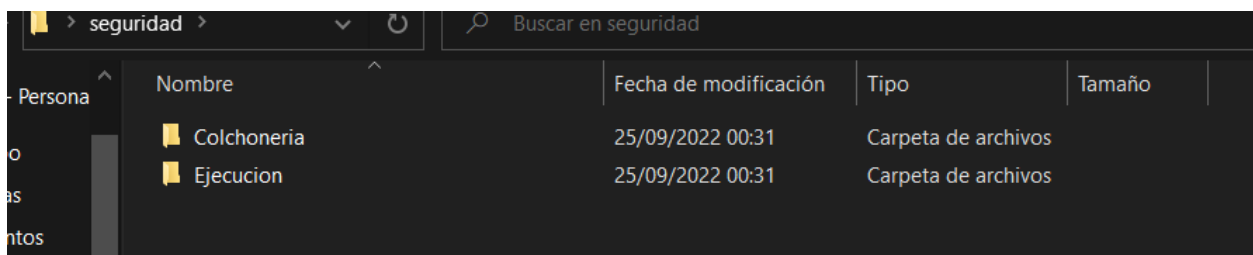
<https://github.com/JRoca00/seguridad.git>

presionamos la tecla enter y empezara a clonar el repositorio en nuestro área seleccionada inicialmente.

Dándonos como resultado la siguiente carpeta.

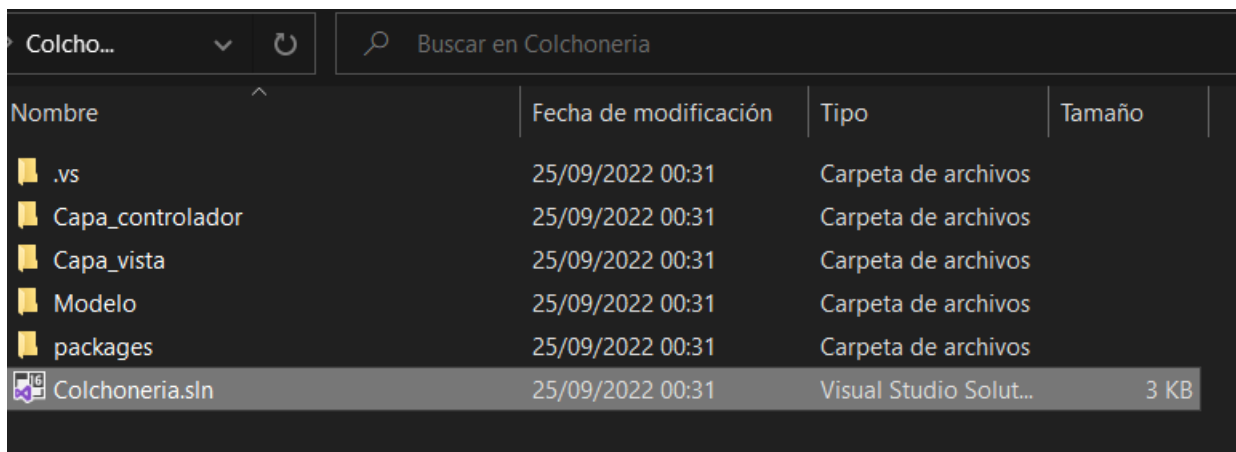



Dentro de ella se encontraran 2 sub carpetas.



Nos vamos a la carpeta colchonería.

Y damos doble click sobre el siguiente icono.






Buscar en Explorador de soluciones (Ctrl+;)

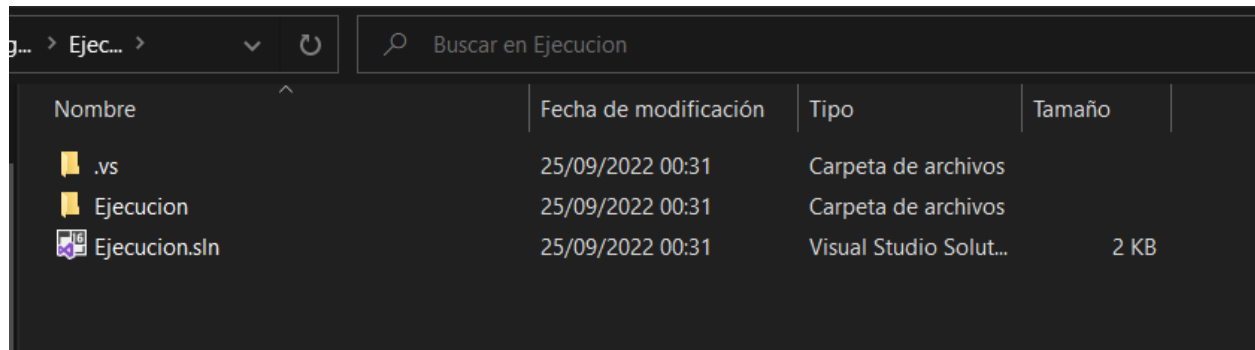
Solución "Colchoneria" (3 de 3 proyectos)

- Capa\_controlador
- Connected Services



```
Salida
Mostrar salida de: Compilación
1> Capa_controlador -> C:\Users\pc\Desktop\seguridad\Colchoneria\Capa_controlador\bin\Debug\Capa_controlador.dll
===== Compilar: 1 correctos, 0 incorrectos, 2 actualizados, 0 omitidos =====
```

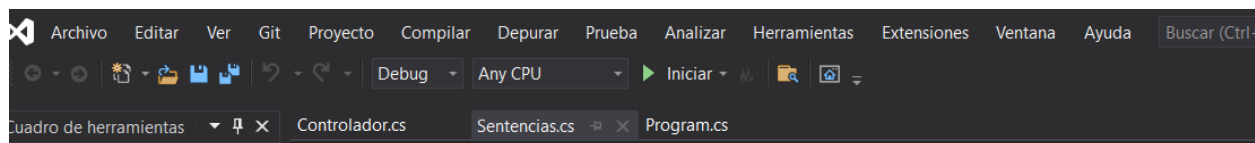
Luego nos vamos a la carpeta ejecución que es una de las 2 sub carpetas principales que se encuentran dentro de la carpeta seguridad.



Damos doble click en ejecución.sln.

Nuevamente nos abrirá el ide visual studio.

Daremos click en iniciar en la flecha verde y nos corra el programa.



Abriéndonos la interfaz de seguridad.

