

BACHILLERATO EN INGENIERÍA EN TELEMÁTICA

BIT-28 SISTEMAS OPERATIVOS II

Proyecto de Investigación

Sistema de Monitoreo en Tiempo Real en Power BI

Profesor

Carlos Méndez Rodríguez

Estudiante

Liliam Castillo Brenes

Fecha de entrega

26 de abril

I Cuatrimestre, 2024

Introducción

En la actualidad con toda la tecnología que se ha estado implementando al pasar de los años ha llevado a las personas y empresas a que se adapten a los nuevos cambios en la tecnología y uno de ellos son los sistemas en tiempo real, estos son un sistema informático capaz de interactuar con un entorno físico y registrar los datos en un entorno digital, permitiendo visualizar los datos de manera continua e inmediata.

Por lo tanto, este proyecto hace referencia a la implementación de un Sistema de Monitoreo en Tiempo Real de los Servicios de TI utilizando Power BI, una base de datos en SQL y Python (Visual Studio Code) y así poder conectar estas tres aplicaciones de ciencia de datos.

Al implementar este sistema en tiempo real se configuró primero Python para leer la información del rendimiento de los servicios de TI, luego en la base de datos se insertan esos mismos datos y Power BI se encargará de mostrar la información en un dashboard en tiempo real.

Planteamiento del Problema

El problema que actualmente presentan muchas pequeñas empresas es que no tiene suficiente presupuesto para contratar un proveedor que les brinde una plataforma que monitoree los Servicios de TI sea los enlaces de red, rendimiento de servidores, los internet entre otros.

Por lo que se implementa un sistema en tiempo real con aplicaciones licenciadas que tiene la empresa como lo es Microsoft Office 365, SQL y Python (Visual Studio Code) que es gratuita y así pueden ahorrarse el presupuesto para un sistema de monitoreo para los servicios de TI y aprovechan todos los beneficios que tienen de las aplicaciones de Office 365.

Objetivo general

- Implementar un Sistema de Monitoreo en Tiempo Real de la Infraestructura de TI.

Objetivos específicos

- Configurar el sistema de monitoreo para que muestre en tiempo real el rendimiento de los servicios de TI.
- Utilizar Power BI, SQL y Python utilizando la terminal de Visual Studio Code para crear el sistema de monitoreo conectando estas tres aplicaciones.

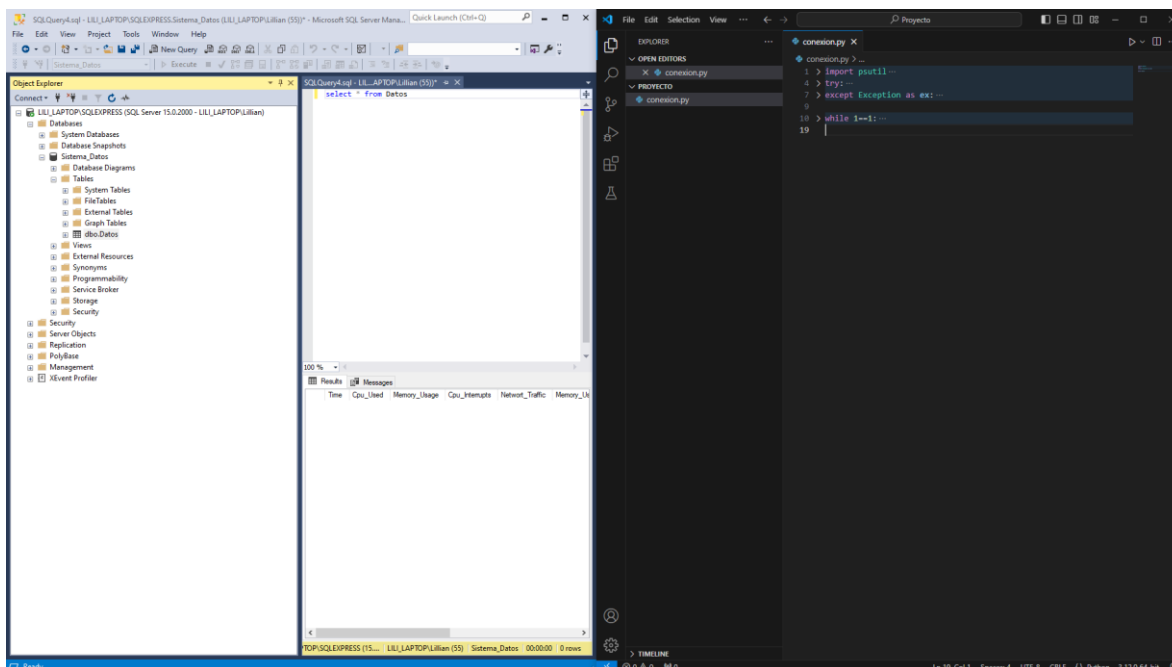
Desarrollo

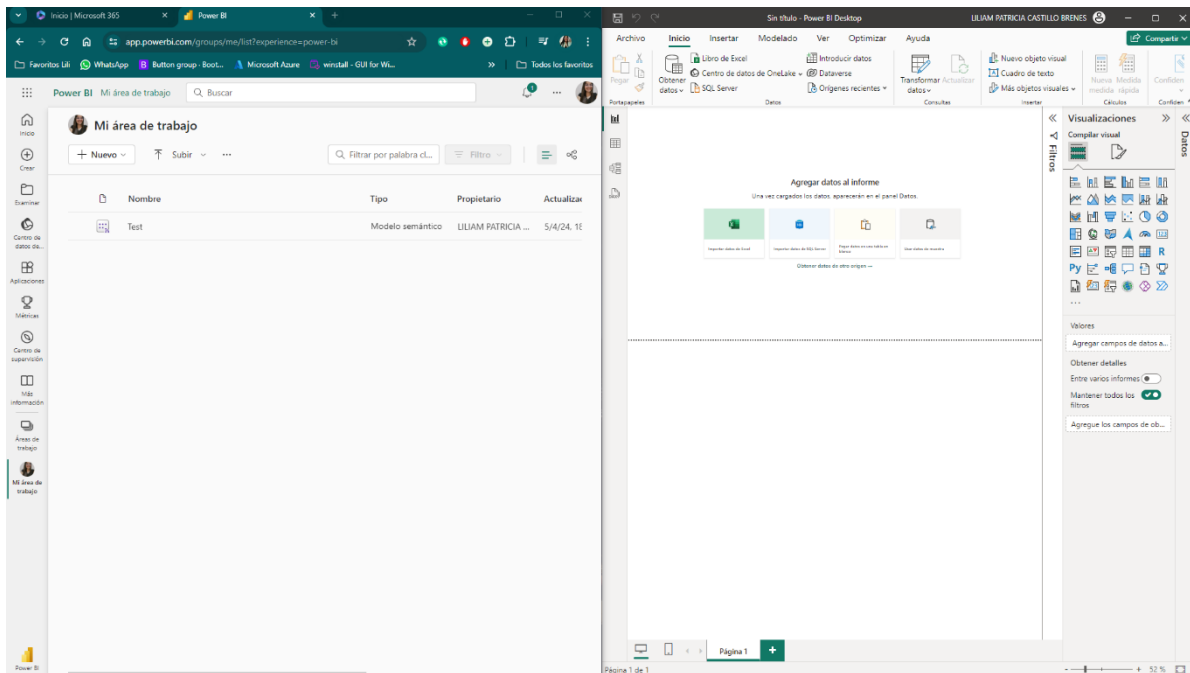
Para implementar el Sistema de Monitoreo en Tiempo Real los Servicio de TI, se utilizó Visual Studio Code para editar el código de Python en donde se importaron las bibliotecas necesarias, una de las más importantes es la biblioteca “**pyodbc**” que permite hacer la conexión con la base de datos de SQL Server.

A continuación, se muestra el paso a paso de la implementación de este sistema.

➤ PASO 1

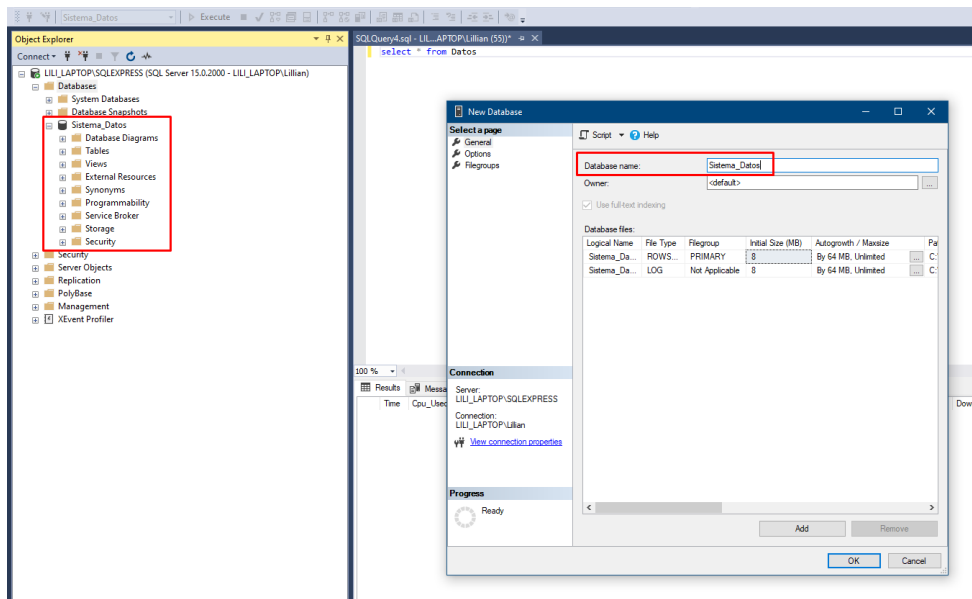
Se instala las aplicaciones de Visual Studio Code para que sea el editor de código de Python, Microsoft SQL Server Management Studio y el Power BI si fuera el Desktop o si no se ingresa al portal de office 365 y se abre la aplicación desde ahí.





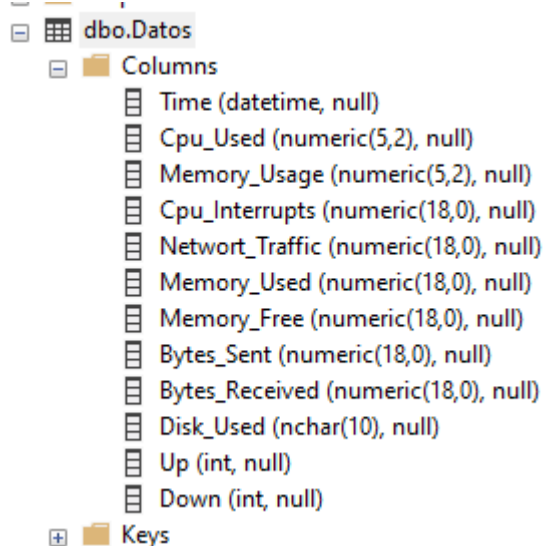
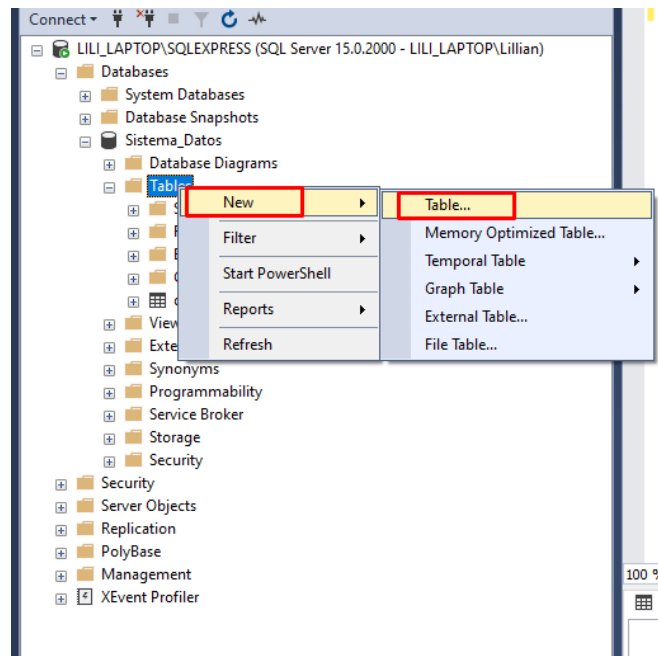
➤ PASO 2

Se selecciona el servidor correspondiente “**LILI_LAPTOP/SQL EXPRESS**” y crea la base de datos en Microsoft SQL Server Management Studio llamada “**Sistema_Datos**”.



➤ PASO 3

En la base de datos que se creó con el nombre de “**Sistema_Datos**” se crean la tabla “**Datos**” con todos los datos que se van a monitorear.



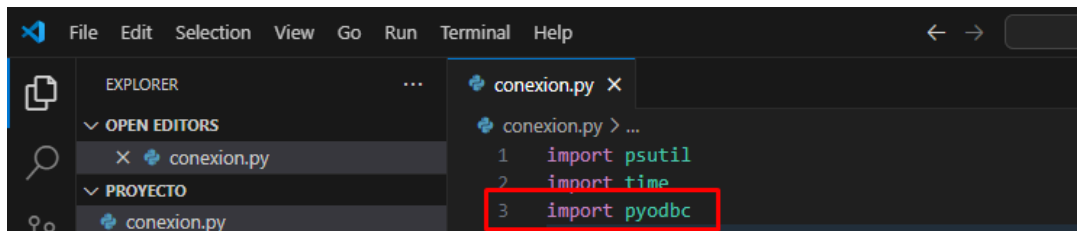
➤ PASO 4

Luego de realizar la base de datos y la tabla de los datos que se van a monitorear, se usara el editor Visual Studio Code para Python y así importar los datos de la tabla “**Datos**” y hacer la conexión con la base de datos.

- Se debe validar en CMD que se tenga instalado el paquete **pyodbc**, que es el que permite hacer la conexión de Python con el SQL Server.

```
C:\Users\Lillian\AppData\Local\Programs\Python\Python312>pip list
Package Version
-----
pip      23.2.1
psutil   5.9.8
pyodbc   5.1.0
```

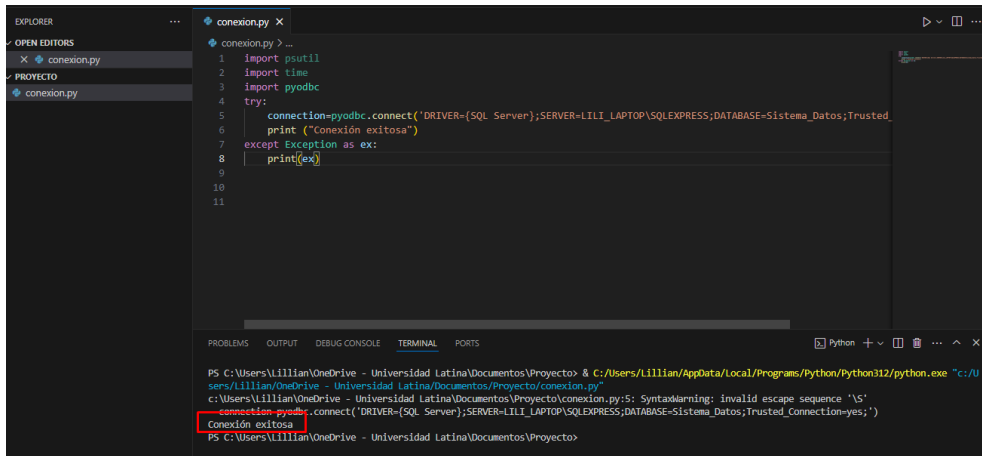
- Se debe importar la biblioteca **pyodbc** en Python (Visual Studio Code) y otros dos que se necesitan para este monitoreo.



```
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
  X conexion.py
PROYECTO
  conexion.py
conexion.py X
conexion.py > ...
1 import psutil
2 import time
3 import pyodbc
```


➤ PASO 5

Luego de importar las bibliotecas se ingresa los comandos para hacer la conexión a la base de datos y se utiliza el comando **print("conexión exitosa")** para asegurar nuestra si se conectó a la base de datos.



The screenshot shows a VS Code editor with a file named `conexion.py`. The code imports `psutil`, `time`, and `pyodbc`. It attempts to connect to a SQL Server database using `pyodbc.connect()` with specific driver and server details. A `try-except` block handles the connection, printing "Conexión exitosa" if successful or the exception if not. The terminal at the bottom shows the command to run the script and the output "Conexión exitosa".

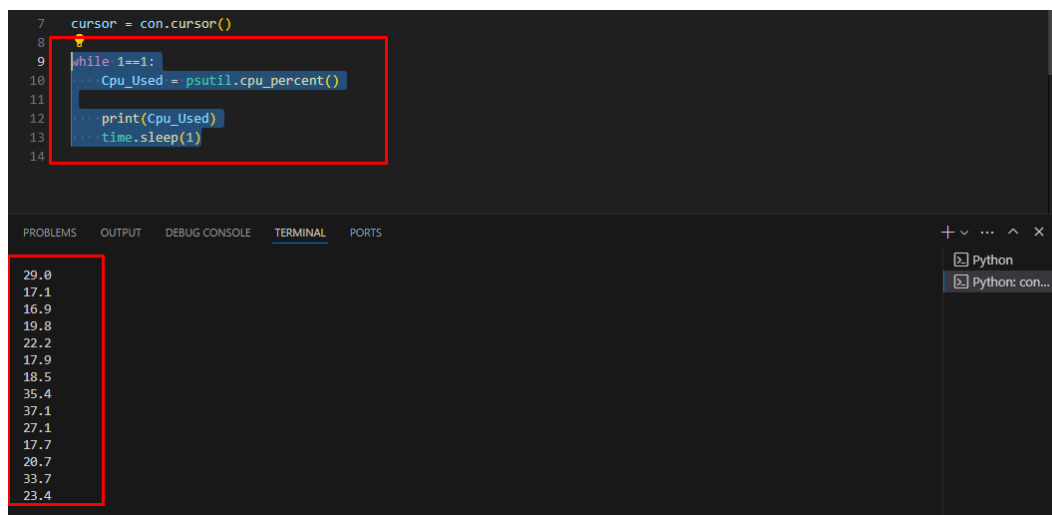
```
1 import psutil
2 import time
3 import pyodbc
4
5 try:
6     connection=pyodbc.connect('DRIVER={SQL Server};SERVER=LILI_LAPTOP\SQLEXPRESS;DATABASE=Sistema_Datos;Trusted_
7     print("Conexión exitosa")
8 except Exception as ex:
9     print(ex)
10
11
```

Terminal Output:

```
PS C:\Users\Lillian\OneDrive - Universidad Latina\Documentos\Proyecto> & C:\Users\Lillian\AppData\Local\Programs\Python\Python312\python.exe "c:/U
c:\Users\Lillian\OneDrive - Universidad Latina\Documentos\Proyecto\conexion.py"
c:\Users\Lillian\OneDrive - Universidad Latina\Documentos\Proyecto\conexion.py:5: SyntaxWarning: invalid escape sequence '\S'
connection=pyodbc.connect('DRIVER={SQL Server};SERVER=LILI_LAPTOP\SQLEXPRESS;DATABASE=Sistema_Datos;Trusted_connection=yes;')
Conexión exitosa
PS C:\Users\Lillian\OneDrive - Universidad Latina\Documentos\Proyecto>
```

➤ PASO 6

Una vez valida la conexión hacia la base datos, se empieza a ingresar los comandos de monitoreo de la tabla “**Datos**”, se prueba el monitoreo en porcentaje del CPU y así se van ingresando todos los datos de la tabla.



The screenshot shows a VS Code editor with a Python script. A `while` loop is defined with the condition `1==1`. Inside the loop, `psutil.cpu_percent()` is used to get the CPU usage percentage, which is then printed and a `time.sleep(1)` is called. The terminal at the bottom shows the output of the script, displaying a list of CPU usage percentages.

```
7 cursor = con.cursor()
8
9 while 1==1:
10     Cpu_Used = psutil.cpu_percent()
11
12     print(Cpu_Used)
13     time.sleep(1)
14
```

Terminal Output:

```
29.0
17.1
16.9
19.8
22.2
17.9
18.5
35.4
37.1
27.1
17.7
20.7
33.7
23.4
```

```
conexion.py
6
7 cursor = con.cursor()
8
9 while 1==1:
10     Cpu_Used = psutil.cpu_percent()
11     Memory_Usage = psutil.virtual_memory() [2]
12     Cpu_Interrupts = psutil.cpu_stats() [1]
13
14     Memory_Used = psutil.virtual_memory () [3]
15     Memory_Free = psutil.virtual_memory () [4]
16     Bytes_Sent = psutil.net_io_counters () [0]
17     Bytes_Received = psutil.net_io_counters () [1]
18     Disk_Used = psutil.disk_usage ('/') [3]
19
20
21 print(Cpu_Used)

time.sleep(1)
KeyboardInterrupt
PS C:\Users\Lillian\OneDrive - Universidad Latina\Documentos\Proyecto> & C:/Users/Lillian/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/Lillian/OneDrive - Universidad Latina\Documentos\Proyecto/conexion.py"
c:\Users\Lillian\OneDrive - Universidad Latina\Documentos\Proyecto\conexion.py:5: SyntaxWarning: invalid escape sequence '\S'
con= pyodbc.connect("DRIVER={SQL Server};SERVER=LILLI_LAPTOP\SQLEXPRESS;DATABASE=Sistema_Datos;Trusted_connection=yes;")
42.3
69.2
351000790
21.8
69.4
351030832
29.5
69.5
0
```

➤ PASO 7

Se ingresa la información requerida que se ha recopilado del sistema y se utiliza el comando **“insert”** para insertar los datos de la tabla **“Datos”** que se creo en la base de datos de SQL Server y que los datos se carguen con la fecha y hora actual.

```
cursor.execute('insert into Datos values (GETDATE(), '
+ str(Cpu_Used)+ ', '
+ str(Memory_Usage)+ ', '
+ str(Cpu_Interrupts)+ ', '
+ str(Memory_Used)+ ', '
+ str( (variable) Bytes_Received: int
+ str(Bytes_Received)+ ', '
+ str(Disk_Used)+ ')')
con.commit()
```

➤ PASO 8

Luego de ingresar los datos de la tabla, se utiliza la función “**commit**” para confirmar el punto de conexión para finalizar el comando anterior y hacer que la base de datos SQL sea accesible para otras aplicaciones.

```
cursor.execute('insert into Datos values (GETDATE(),'+
+ str(Cpu_Used)+ ','
+ str(Memory_Usage)+ ','
+ str(Cpu_Interrupts)+ ','
+ str(Memory_Used)+ ','
+ str(Memory_Free)+ ','
+ str(Bytes_Sent)+ ','
+ str(Bytes_Received)+ ','
+ str(Disk_Used)+ ')')
con.commit()
```

➤ PASO 9

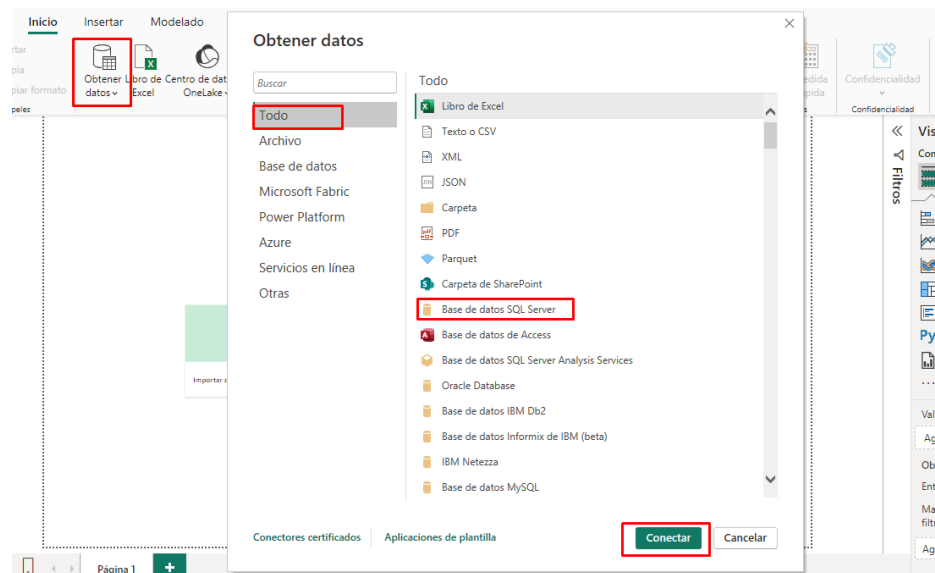
Se valida que se haya hecho la conexión correctamente, para validar el código en Python se ejecuta y luego se ejecuta un script en la base datos SQL Server que debe generar los datos que van a monitorear.

The screenshot displays two windows. The top window is Microsoft SQL Server Enterprise Manager, showing a query result for 'select * from Datos'. The bottom window is a Python IDE (VS Code) with a file named 'conexion.py'. The Python code includes imports for 'psutil' and 'cursor', and a function to insert data into the 'Datos' table. The output of the script is visible in the terminal, showing a list of data points.

Time	Cpu_Used	Memory_Usage	Cpu_Interrupts	Memory_Used	Memory_Free	Bytes_Sent	Bytes_Received	Disk_Used
2024-04-19 21:22:04.100	24.40	64.00	72503605	8100050476	4559401950	5550101250	4854102150	82.3
2024-04-19 21:22:07.210	18.80	64.00	725658685	8178905888	4607029248	355193114	4854161198	82.3
2024-04-19 21:22:10.303	19.20	63.90	725632021	8167999136	4618039200	355194803	4854214118	82.3
2024-04-19 21:22:13.367	20.80	63.90	725709161	816489056	4620411584	355196751	4854264273	82.3
2024-04-19 21:22:16.443	10.60	63.90	725721665	8172396544	4613124096	355199581	4854316041	82.3
2024-04-19 21:22:19.533	13.70	63.90	725746213	8170295296	4619634844	355201299	4854370883	82.3
2024-04-19 21:22:22.640	7.40	63.80	725766867	8162226176	4623284464	355203023	4854422549	82.3
2024-04-19 21:22:25.710	10.70	63.80	725789423	8161107968	4624412872	355204623	4854474021	82.3
2024-04-19 21:22:28.803	6.00	63.80	725789894	8159682560	4626247880	355206445	4854526807	82.3
2024-04-19 21:22:31.870	8.40	63.80	725789410	8156708864	4628221376	355207850	4854581202	82.3
2024-04-19 21:22:34.990	13.00	63.80	725813757	8155869184	4630091056	355209704	4854634388	82.3
2024-04-19 21:22:38.070	11.10	63.80	725829515	815309888	4632436736	355212126	4854687352	82.3
2024-04-19 21:22:41.160	8.10	63.80	725836555	8154419200	4631470880	355212763	4854738584	82.3
2024-04-19 21:22:44.253	7.50	63.90	725847600	817548096	4613382144	355221394	4854796653	82.3
2024-04-19 21:22:47.327	13.10	63.80	725860078	815784704	4628537344	355222039	4854848194	82.3
2024-04-19 21:22:50.380	14.30	64.40	725893866	823946688	4557201488	355229577	4854900638	82.3
2024-04-19 21:22:53.470	9.40	63.90	725898058	8177623040	4608495860	355231855	4854959573	82.3
2024-04-19 21:22:56.570	7.00	63.90	725908736	8166887424	4619042816	355233168	4855011349	82.3
2024-04-19 21:22:59.675	13.10	63.80	725937365	8162478672	4621481364	355234576	4855064136	82.3

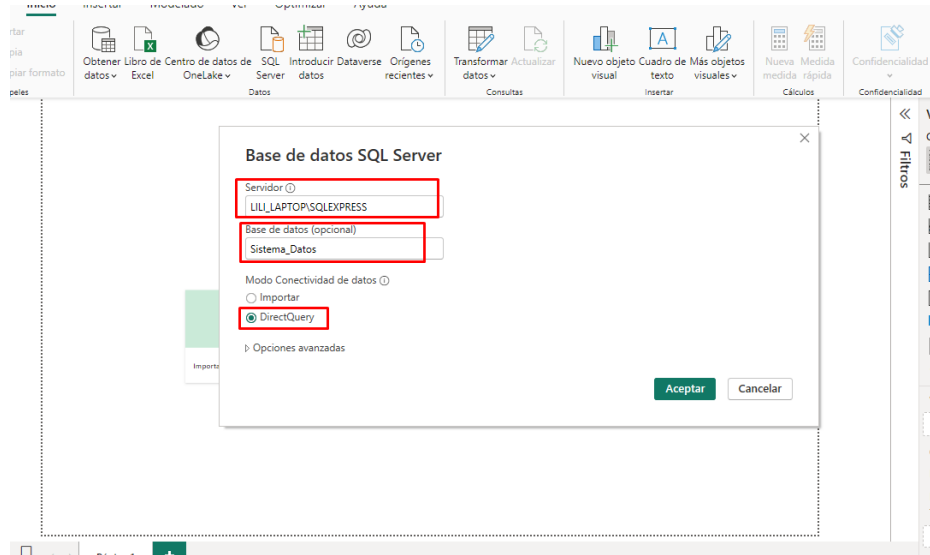
➤ PASO 10

Al validar la conexión, se realizaría la conexión de Power BI con SQL Server. Primero se abre la aplicación Power BI y se le da clic en el icono que dice **“Obtener Datos”** aparece una ventana se selecciona como data el **“SQL Server database”** y se le da en conectar.



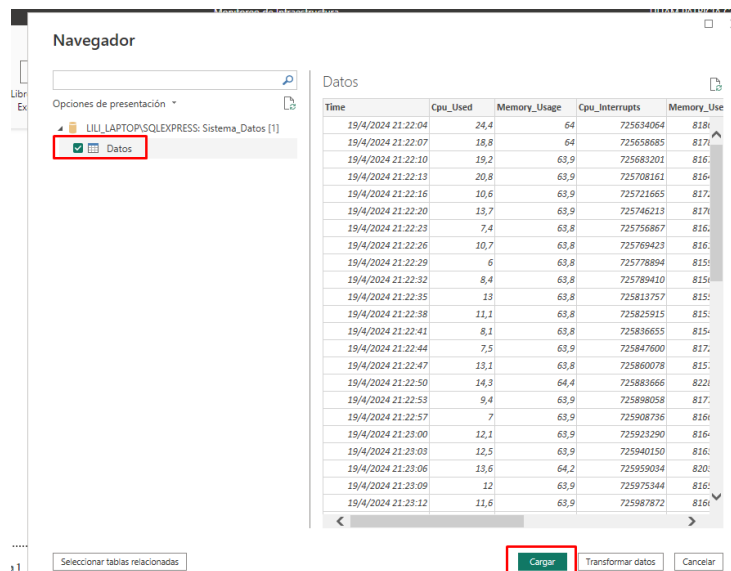
➤ PASO 11

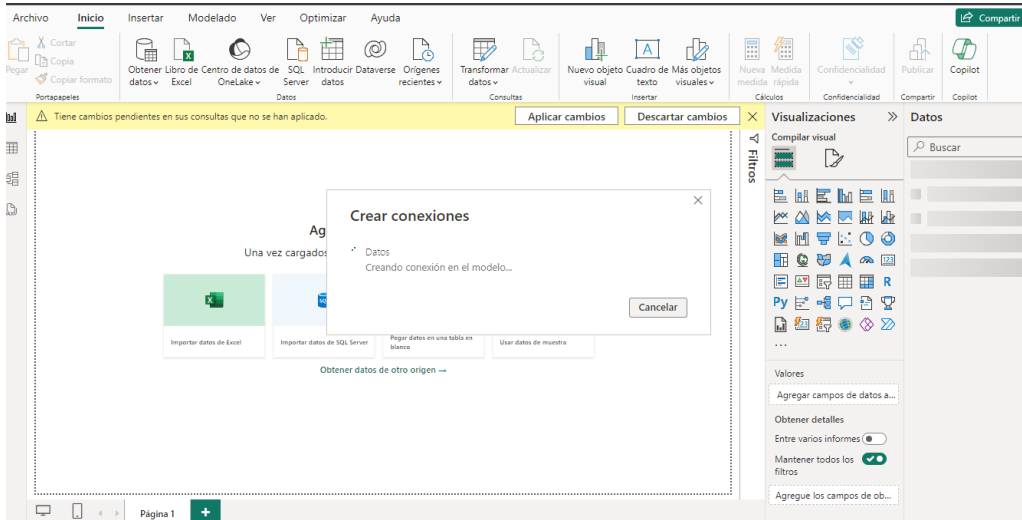
Luego de darle clic en “**Conectar**” aparece la ventana para colocar la información del servidor, base de datos y escoger el modo de conectividad de datos en este caso seria “**DirectQuery**” y darle clic en aceptar.



➤ PASO 12

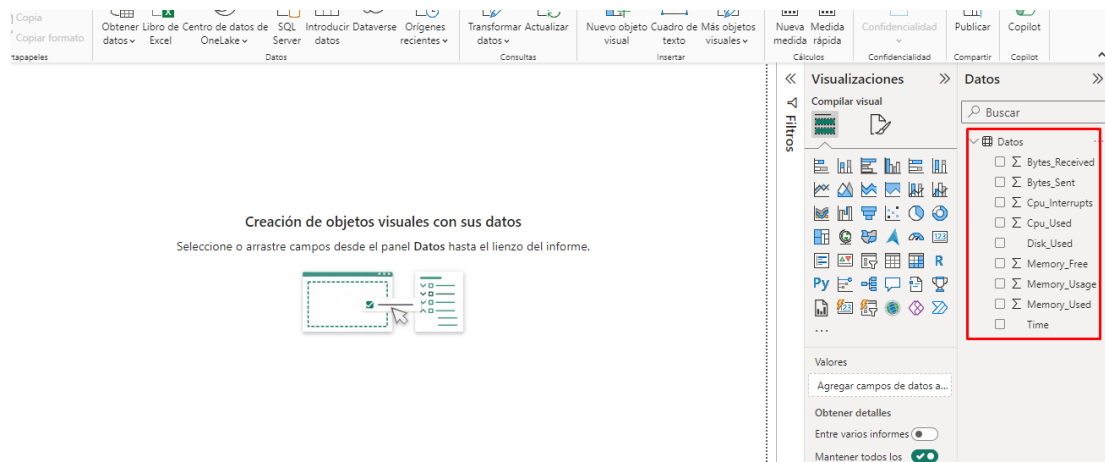
Al dar le clic en Aceptar aparece otra ventana donde se debe seleccionar la tabla de la base de datos y se le da en cargar.





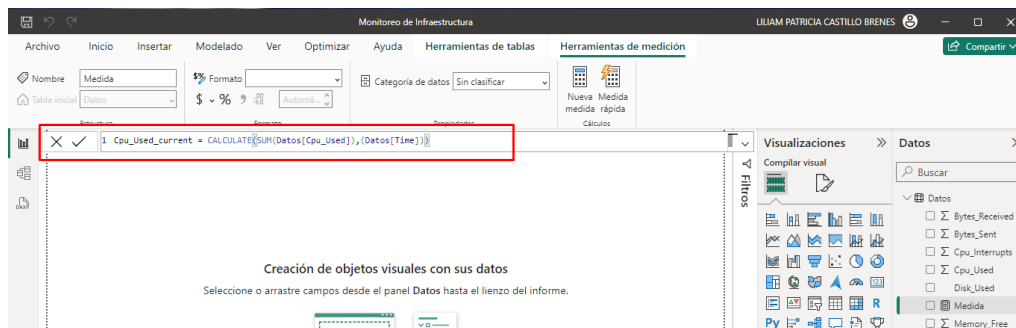
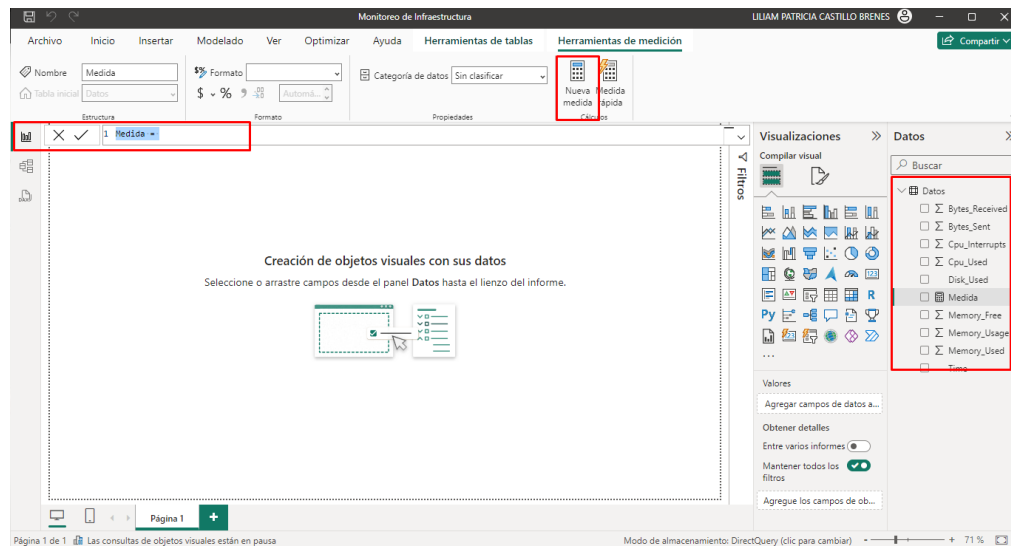
➤ PASO 13

Al cargar la base datos, mostrara la tabla “**Datos**” que se creó en Microsoft SQL Server Management Studio, con eso indica que Power BI ya está conectado directamente a la base de datos de SQL y todos los campos creados están disponibles para crear el dashboard.



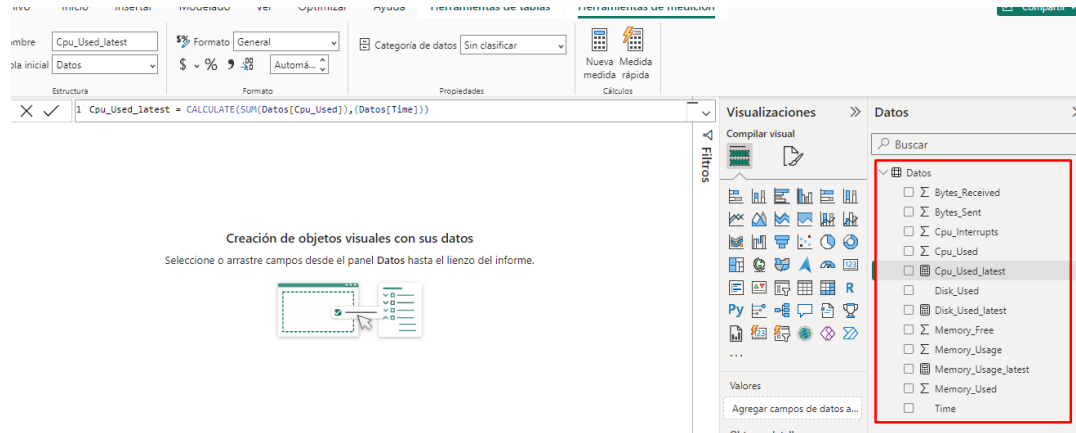
➤ PASO 14

Antes de crear el dashboard, se deben crear las medidas usando DAX para calcular los valores actuales del uso del CPU, memoria y disco duro en comparación con el ultimo valor de fecha y hora.



➤ PASO 15

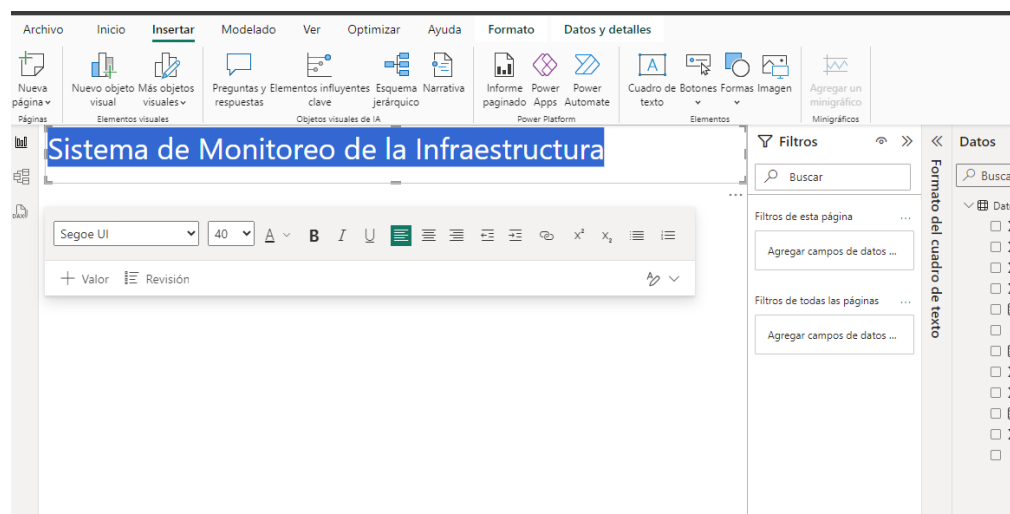
Se agregan las medidas a uso de CPU, disco duro y memoria.



➤ PASO 16

Se valida que se esté conectado directamente a la base de datos del servidor SQL y se crea el dashboard para mostrar la información de rendimiento de la infraestructura en tiempo real.

- Se agrega el título del dashboard en una caja de texto.

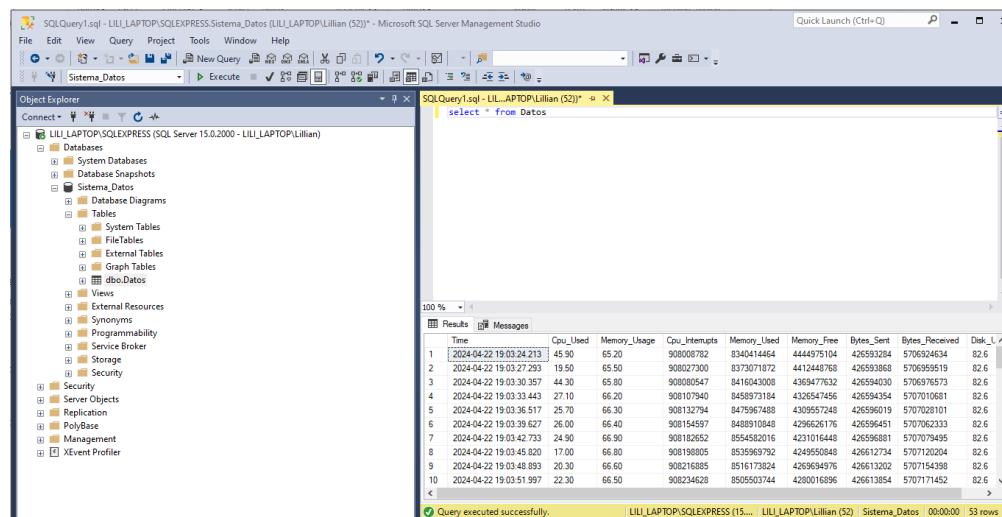


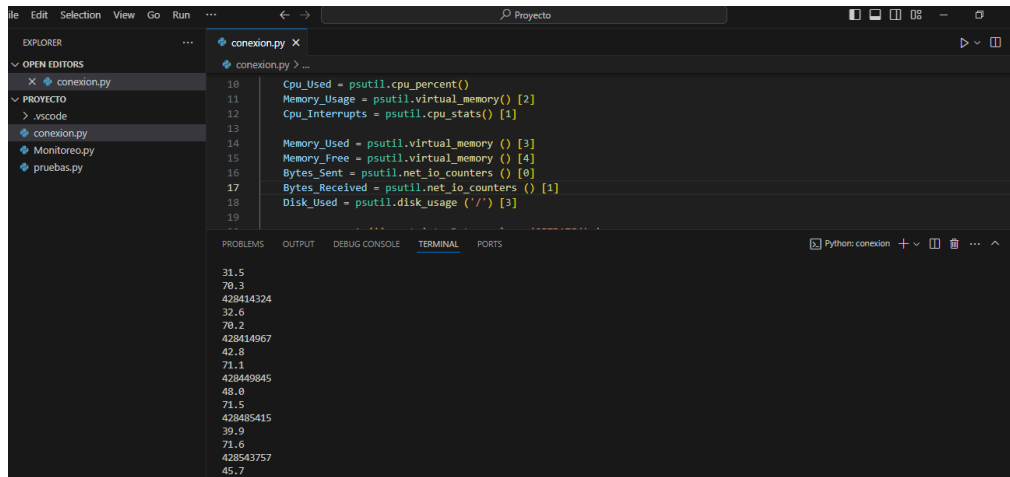
- Se agrega un indicador visual para mostrar el uso del CPU, bytes enviados y memoria y se va personalizando cada gráfico.



➤ PASO 17

Para terminar cuando ya se tengan los gráficos que se quieren usar para monitorear, en la base de datos se coloca el query “**Select * from (tabla) Datos**” en Python se ejecuta el código.



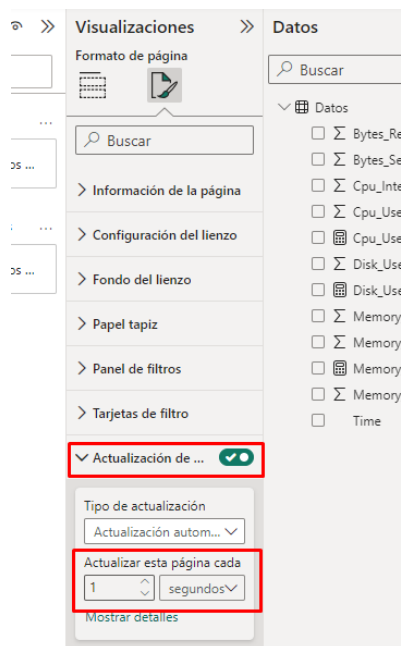


```
10 Cpu_Used = psutil.cpu_percent()
11 Memory_Usage = psutil.virtual_memory() [2]
12 Cpu_Interrupts = psutil.cpu_stats() [1]
13
14 Memory_Used = psutil.virtual_memory () [3]
15 Memory_Free = psutil.virtual_memory () [4]
16 Bytes_Sent = psutil.net_io_counters () [0]
17 Bytes_Received = psutil.net_io_counters () [1]
18 Disk_Used = psutil.disk_usage ('/') [3]
19
```

31.5
70.3
428414324
32.6
70.2
428414967
42.8
71.1
428449845
48.0
71.5
428485415
39.9
71.6
428543757
45.7

➤ PASO 18

Por ultimo se configura en Power BI como se va a actualizar los gráficos y así se crea un simulador en tiempo real.



Conclusiones

Al concluir este proyecto del sistema de monitoreo en tiempo real se determino que si se puede monitorear la información que se requiera sin invertir tanto dinero de una forma sencilla.

Al implementar un sistema en tiempo real se deben tener varias configuraciones ya a nivel de empresa para que sea un sistema de monitoreo oportuno y preciso.

Los beneficios de realizar este sistema es tener eficiencia ya que se tendría un mejor manejo del funcionamiento de los servicios de TI y además se configurarían alertas más específicas para los administradores del sistema e incluso se podría tener personal 24/7 para que realicen el monitoreo, además se puede considera tener un control que dos días a la semana generar un reporte del estado de los servicios.

Por ser un prototipo se tuvieron varias limitaciones, pero se investigó que hay otras maneras de implementar un sistema de monitoreo en tiempo real con Power BI y conectarlo con aplicaciones más robusta.

Y por ultimo se determino que al implementar este tipo de sistemas se tiene varios riesgos como la seguridad de los datos ya que pueden ser datos sensibles o críticos recopilados y transmitidos por el sistema que pueden ser recolectados por personas no autorizadas, si no se implementan medidas adecuadas de validación y verificación existe el riesgo de que los datos recopilados y almacenados no sean precisos o incompletos, como se mencionó anteriormente el sistema se debe diseñar lo suficiente robusto y escalable para que no existan interrupciones no planificadas o fallos que afectan la disponibilidad también existe el riesgo que el rendimiento no sea satisfactorio y la incompatibilidad con los otros sistemas.