



Taller Transacciones

I. PORTADA

Tema:	Tema de la guía práctica proporcionada por el docente
Unidad de Organización Curricular:	PROFESIONAL
Nivel y Paralelo:	Quinto - A
Alumnos participantes:	Guachamin Shiguango Josue Andres Rodriguez Romero Jhon Wilmer Vallejo Rengifo John David Villacis Silva Stefano Mateo Yugcha Quishpe Erick Geomar
Asignatura:	Sistema de bases de datos distribuidos
Docente:	Ing. José Caiza, Mg.

II. INFORME DE GUÍA PRÁCTICA

2.1 Objetivos

General:

Establecer una nueva instancia en SQL Server

Específicos:

- Instalar y configurar SQL Server en Ubuntu, y comprobar que está funcionando correctamente.
- Vincular SQL Server en Ubuntu con el programa de gestión de SQL Server Management Studio (SSMS) en Ventanas.
- Elaborar y revisar una vista en la base de datos "Flotilla" para verificar la interacción entre los dos servidores

2.2 Modalidad

Presencial.

2.3 Tiempo de duración

Presenciales: 2

No presenciales: 0

2.4 Instrucciones

- Establecer una conexión con el SQL Server Management Studio (SSMS).
- Desplegar los scripts de SQL que se ofrecen en la guía.
- Registrar los resultados y las capturas de pantalla.
- Antes de seguir adelante, confirmar cada paso con el profesor

2.5 Listado de equipos, materiales y recursos

- Computador con Windows/Linux/macOS.
- SQL Server Management Studio (SSMS) instalado.
- Acceso a la base de datos CentroMedicoDB.

TAC (Tecnologías para el Aprendizaje y Conocimiento) empleados en la guía práctica:

- ☒ Plataformas educativas
- ☒ Simuladores y laboratorios virtuales
- ☐ Aplicaciones educativas
- ☒ Recursos audiovisuales
- ☐ Gamificación
- ☐ Inteligencia Artificial

Otros (Especifique): _____



2.6 Actividades desarrolladas

- **Verificar conexión y estructura de la base de datos**
 - Abrimos SQL Server Management Studio (SSMS) y nos conectamos al servidor.
 - Seleccionamos la base de datos “CentroMedicoDB”.
 - Verificamos las tablas existentes ejecutando:
SELECT * FROM INFORMATION_SCHEMA.TABLES;

The screenshot shows the SQL Server Management Studio (SSMS) interface. The top pane displays a query in the 'SQLQuery2.s...SONAL (70))*' window. The query is:

```
1  USE CentroMedicoDB;
2  GO
3  SELECT * FROM INFORMATION_SCHEMA.TABLES;
4
```

The bottom pane shows the 'Results' tab with a table of results. The table has five columns: TABLE_CATALOG, TABLE_SCHEMA, TABLE_NAME, and TABLE_TYPE. The results show eight tables in the CentroMedicoDB database, all of which are BASE TABLES.

	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE
1	CentroMedicoDB	dbo	Especialidades	BASE TABLE
2	CentroMedicoDB	dbo	AsignacionEspecialidades	BASE TABLE
3	CentroMedicoDB	dbo	Empleados	BASE TABLE
4	CentroMedicoDB	dbo	Consultas	BASE TABLE
5	CentroMedicoDB	dbo	Medicos	BASE TABLE
6	CentroMedicoDB	dbo	CentroMedicos	BASE TABLE
7	CentroMedicoDB	dbo	Clientes	BASE TABLE
8	CentroMedicoDB	dbo	sysdiagrams	BASE TABLE

Ilustración 1: CentroMedicoDB



- **Probar integridad referencial con datos existentes**
 - Ejemplo 1: Insertamos un médico con una especialidad inexistente:

```
1 USE CentroMedicoDB;
2 GO
3
4 INSERT INTO Medicos (Nombre, Apellido, IDEspecialidades, IDCentro, Email, Teléfono)
5 VALUES ('Erick', 'Yugcha', 2, 1, 'erick@mail.com', '987654389');
6
7 SELECT * FROM Medicos
```

IDMedico	Nombre	Apellido	IDEspecialidades	IDCentro	Email	Teléfono
1	Erick	Yugcha	2	1	erick@mail.com	987654389

Ilustración 2: Especialidades existentes

```
1 USE CentroMedicoDB;
2 GO
3
4 INSERT INTO Medicos (Nombre, Apellido, IDEspecialidades, IDCentro, Email, Teléfono)
5 VALUES ('Erick', 'Yugcha', 3, 1, 'erick@mail.com', '987654389');
6
```

Msg 547, Level 16, State 0, Line 4
The INSERT statement conflicted with the FOREIGN KEY constraint "FK_Medicos_Especialidades". The conflict occurred in database "CentroMedicoDB". The statement has been terminated.
Completion time: 2025-10-03T10:36:43.0942342-05:00

Ilustración 3: Error en inserción de Médico con especialidad inexistente

- Ejemplo 2: Intentar eliminar una especialidad en uso (Pediatria, ID = 1)

```
8
9 DELETE FROM Especialidades WHERE IDEspecialidades = 2;
```

Msg 547, Level 16, State 0, Line 9
The DELETE statement conflicted with the REFERENCE constraint "FK_Medicos_Especialidades". The conflict occurred in database "CentroMedicoDB". The statement has been terminated.
Completion time: 2025-10-03T10:40:07.3169385-05:00

Ilustración 4: Error en intentar eliminar una especialidad en uso



- **Transacciones con atomicidad (Commit y Rollback)**
 - **Transacción exitosa:** Crear un centro médico y asignar un médico.

The screenshot shows a SQL query window with two tabs: 'SQLQuery13.s...ERSONAL (73))' and 'SQLQuery2.s...SONAL (63))*'. The active tab contains the following SQL code:

```
11
12 -- 1. DECLARAR la variable al inicio
13 DECLARE @NuevoCentroID INT;
14
15 INSERT INTO CentroMedicos (Nombre, Ciudad, Direccion, Teléfono)
16 VALUES ('Clínica Norte', 'Quito', 'Av. Principal 123', '02222222');
17 -- 2. ASIGNAR el valor del ID recién creado a la variable
18 SET @NuevoCentroID = SCOPE_IDENTITY();
19 -- 3. Usar la variable en el siguiente INSERT
20 INSERT INTO Medicos (Nombre, Apellido, IDEspecialidades, IDCentro, Email, Teléfono)
21 VALUES ('Caro', 'Vega', 2, @NuevoCentroID, 'caro@clinicanorte.ec', '0978345276');
22 COMMIT TRANSACTION
23
24 SELECT * FROM CentroMedicos WHERE Nombre = 'Clínica Norte';
25 SELECT * FROM Medicos WHERE Nombre = 'Caro';
26
```

Below the query, the 'Results' tab shows two tables. The first table, 'CentroMedicos', has one row with IDCentro 6, Nombre 'Clínica Norte', Ciudad 'Quito', Direccion 'Av. Principal 123', and Teléfono '02222222'. The second table, 'Medicos', has one row with IDMedico 12, Nombre 'Caro', Apellido 'Vega', IDEspecialidades 2, IDCentro 6, Email 'caro@clinicanorte.ec', and Teléfono '0978345276'.

Ilustración 5: Creación de centro Médico y asignación de Médico

- **Transacción fallida:** Intentar asignar un médico a un centro inexistente.

The screenshot shows a SQL query window with a single tab containing the following SQL code:

```
31
32 select * from CentroMedicos
```

Below the query, the 'Results' tab shows a table with two rows of data for 'CentroMedicos':

	IDCentro	Nombre	Ciudad	Direccion	Teléfono
1	1	Huachi	Ambato	Sur	0967453623
2	6	Clínica Norte	Quito	Av. Principal 123	02222222

Ilustración 6: Centros Médicos existentes



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
CICLO ACADÉMICO: AGOSTO 2025 – ENERO 2026



```
26
27  INSERT INTO Medicos (Nombre, Apellido, IDEspecialidades, IDCentro, Email, Teléfono)
28  VALUES ('Jhon', 'Vallejo', 3, 2, 'jhon@email.com', '987654376')
```

100 % No issues found Ln: 28 Ch: 22 TABS CRLF

Messages

Msg 208, Level 16, State 1, Line 27
Invalid object name 'Medicos'.

Completion time: 2025-10-03T11:27:06.6364998-05:00

Ilustración 7: Error en asignación de un Médico a un centro inexistente

- **Pruebas concurrentes**
 - **Sesión 1 (Actualización)**

```
33
34  BEGIN TRANSACTION;
35  UPDATE Medicos SET Email = 'caro.uno@cl clinicanorte.ec' WHERE Nombre = 'Caro';
36
37
```

100 % No issues found

Messages

(1 row affected)

Completion time: 2025-10-03T11:37:00.3886265-05:00

Ilustración 8: Actualización de Email

- **Sesión 2 (Lectura)**

```
36
37  SELECT * FROM Medicos WHERE Nombre = 'Caro';
38
```

100 % No issues found

Results Messages

	IDMedico	Nombre	Apellido	IDEspecialidades	IDCentro	Email	Teléfono
1	12	Caro	Vega	2	6	caro.uno@cl clinicanorte.ec	0978345276

Ilustración 9: Vista Médicos

- **Manejo de errores con TRY...CATCH**
 - Ejemplo: Insertar una consulta con un ClienteID inexistente.sql

```
38
39  BEGIN TRY
40  BEGIN TRANSACTION;
41  INSERT INTO Consultas (IDMedico, IDCliente, FechaConsulta)
42  VALUES (1, 9999, GETDATE());
43  COMMIT TRANSACTION;
44  END TRY
45  BEGIN CATCH
46  ROLLBACK TRANSACTION;
47  PRINT 'Error: ' + ERROR_MESSAGE();
48  END CATCH;
49
```

100 % 2 0 Ln: 49 Ch: 1 TABS CRLF

Messages

(0 rows affected)

Error: Cannot insert the value NULL into column 'Tratamiento', table 'CentroMedicoDB.dbo.Consultas'; column does not allow nulls. INSERT f

Completion time: 2025-10-03T11:42:27.9001626-05:00

Ilustración 10: Manejo de errores con TRY CATCH



- **Verificación final**
Consultar datos nuevos:
 - Verificar el nuevo centro médico.
 - Verificar el médico asignado al nuevo centro.

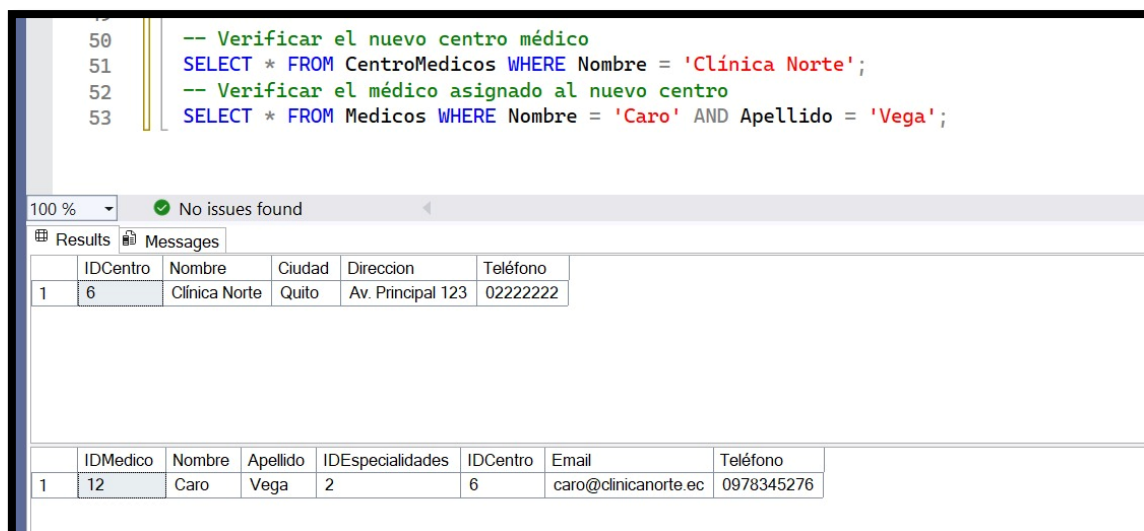


Ilustración 11: Verificación final

2.7 Resultados obtenidos

- Se incorporó la clínica "Clínica Norte" y se designó a la doctora "Marta Vega".
- Se cumplió la atomicidad de la transacción: se llevaron a cabo las dos operaciones como una unidad única.

2.8 Habilidades blandas empleadas en la práctica

- ☐ Liderazgo
- ☒ Trabajo en equipo
- ☐ Comunicación asertiva
- ☐ La empatía
- ☒ Pensamiento crítico
- ☐ Flexibilidad
- ☒ La resolución de conflictos
- ☒ Adaptabilidad
- ☒ Responsabilidad

2.9 Conclusiones

- Las transacciones aseguran que se realicen múltiples operaciones de forma atómica evitando situaciones inconsistentes.
- La utilización de SCOPE_IDENTITY() garantiza que las tablas que tienen claves se relacionen adecuadamente extranjeras.

2.10 Recomendaciones

- Es preciso comprobar siempre la presencia de claves foráneas antes de añadir datos.
- Emplear transacciones en operaciones delicadas que incluyan varias tablas.



2.11 Referencias bibliográficas

- [1] Microsoft Docs, "Claves foráneas en SQL Server", 2023. [En línea].
Disponible: <https://docs.microsoft.com/es-es/sql/relational-databases/tables/createforeign-key-relationships>.
- [2] SQL Server Tutorial, "Manejo de errores con TRY CATCH", 2023. [En línea].
Disponible: <https://www.sqlservertutorial.net/>.

2.12 Anexos

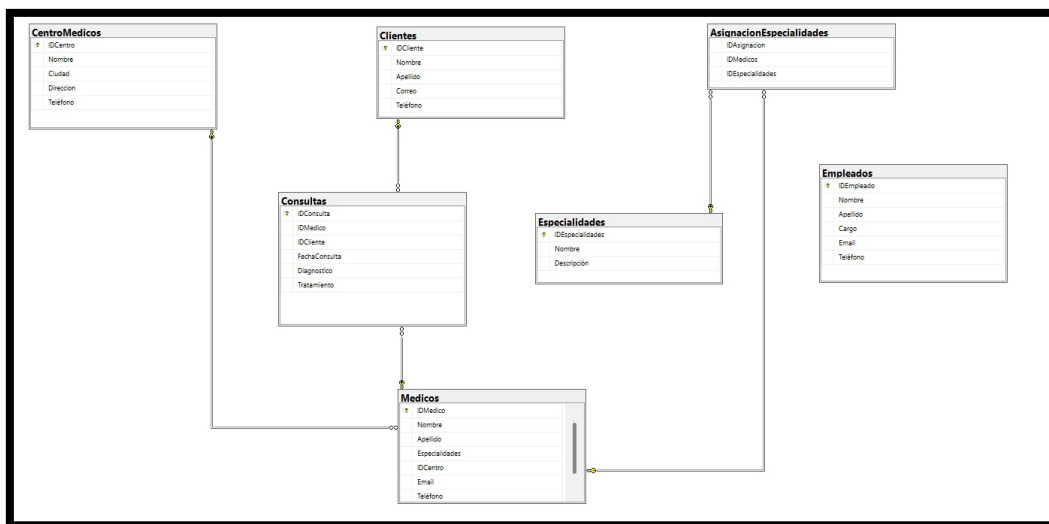


Ilustración 12: Diagrama de relaciones de tablas

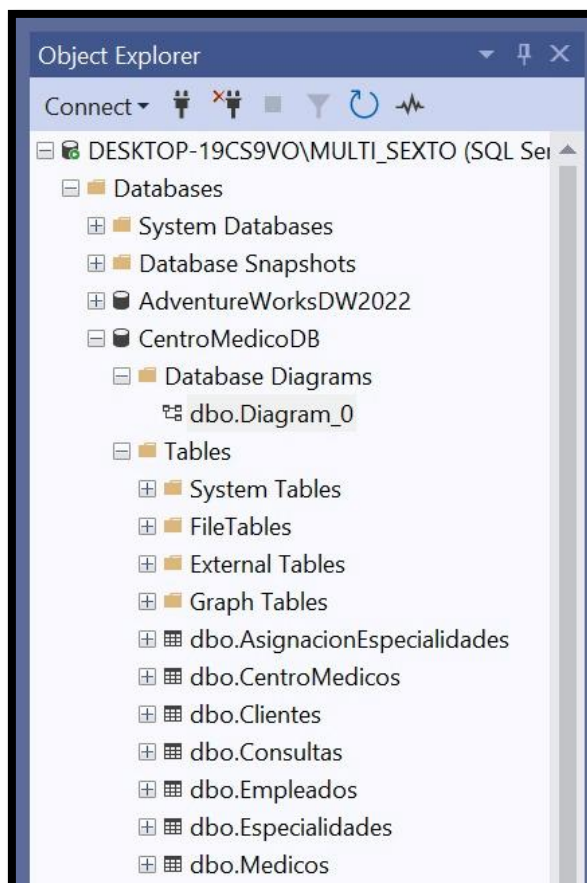


Ilustración 13: BD "CentroMedicoDB"