



INFORME DE GUÍA PRÁCTICA

I. PORTADA

Tema:	Tratamiento de transacciones
Unidad de Organización Curricular:	PROFESIONAL
Nivel y Paralelo:	Quinto “A”
Alumnos participantes:	Bayas Nuñez Axel Nicolas
Asignatura:	Sistemas de Bases de Datos Distribuidos
Docente:	Ing. Caiza Rubén.

II. INFORME DE GUÍA PRÁCTICA

2.1 Objetivos

General:

Determinar el comportamiento de un SGBD con transacciones

Específicos:

- Comprender y aplicar los conceptos de atomicidad, consistencia, aislamiento y durabilidad (ACID) en transacciones de bases de datos distribuidas.
- Implementar y manejar transacciones utilizando comandos como COMMIT y ROLLBACK para garantizar la integridad de los datos en un entorno distribuido.

2.2 Modalidad

Presencial.

2.3 Tiempo de duración

Presenciales: 6

No presenciales: 0

2.4 Instrucciones

Acciones Previas: Verifique que está instalado SQL Server (instancia por defecto)

2.5 Listado de equipos, materiales y recursos

Listado de equipos y materiales generales empleados en la guía práctica:

- Inteligencia artificial
- TAC
- Computador
- SQL Server

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 por desarrollar

- Conéctese al motor de base de datos
- Ingrese algunos datos y verifiquemos la integridad referencial
- Creamos transacciones y probamos las características de atomicidad Commit y Rollback. Habilitamos una nueva sesión para pruebas

2.7 Resultados obtenidos

Modelo de Base de Datos Hospitalaria Distribuida

Para implementar la aplicación hospitalaria, se diseñó una base de datos relacional distribuida que refleja la operatividad de un sistema de salud con múltiples centros médicos interconectados. Esta base de datos fue fragmentada horizontalmente y replicada entre nodos virtuales ubicados en Quito, Guayaquil y Cuenca, lo cual garantiza tanto la consistencia como la alta disponibilidad de los datos.

El modelo entidad-relación contempla las siguientes entidades principales:

- **users:** Tabla que almacena la información de los usuarios del sistema, incluyendo credenciales, tipo de usuario y estado.
- **roles y userRole:** Gestionan los distintos perfiles de usuario (como administrador, médico, secretaria), permitiendo asignaciones flexibles mediante relaciones muchos-a-muchos.
- **doctor:** Contiene detalles profesionales de los médicos registrados, tales como especialidades, experiencia y biografía.
- **patient:** Almacena información detallada de los pacientes, incluyendo datos personales, historial clínico y características médicas específicas.
- **CentroMedico y Consultorio:** Representan la infraestructura médica disponible, permitiendo registrar centros médicos y los consultorios disponibles en cada uno.
- **ConsultaMedica:** Tabla central que registra las consultas realizadas, incluyendo fecha, hora, profesional responsable y consultorio asignado.
- **Operacion y ObservationOperation:** Controlan las operaciones quirúrgicas realizadas, con sus respectivas observaciones médicas.
- **Quirofono:** Define la disponibilidad y características de los quirófanos en los distintos centros médicos.

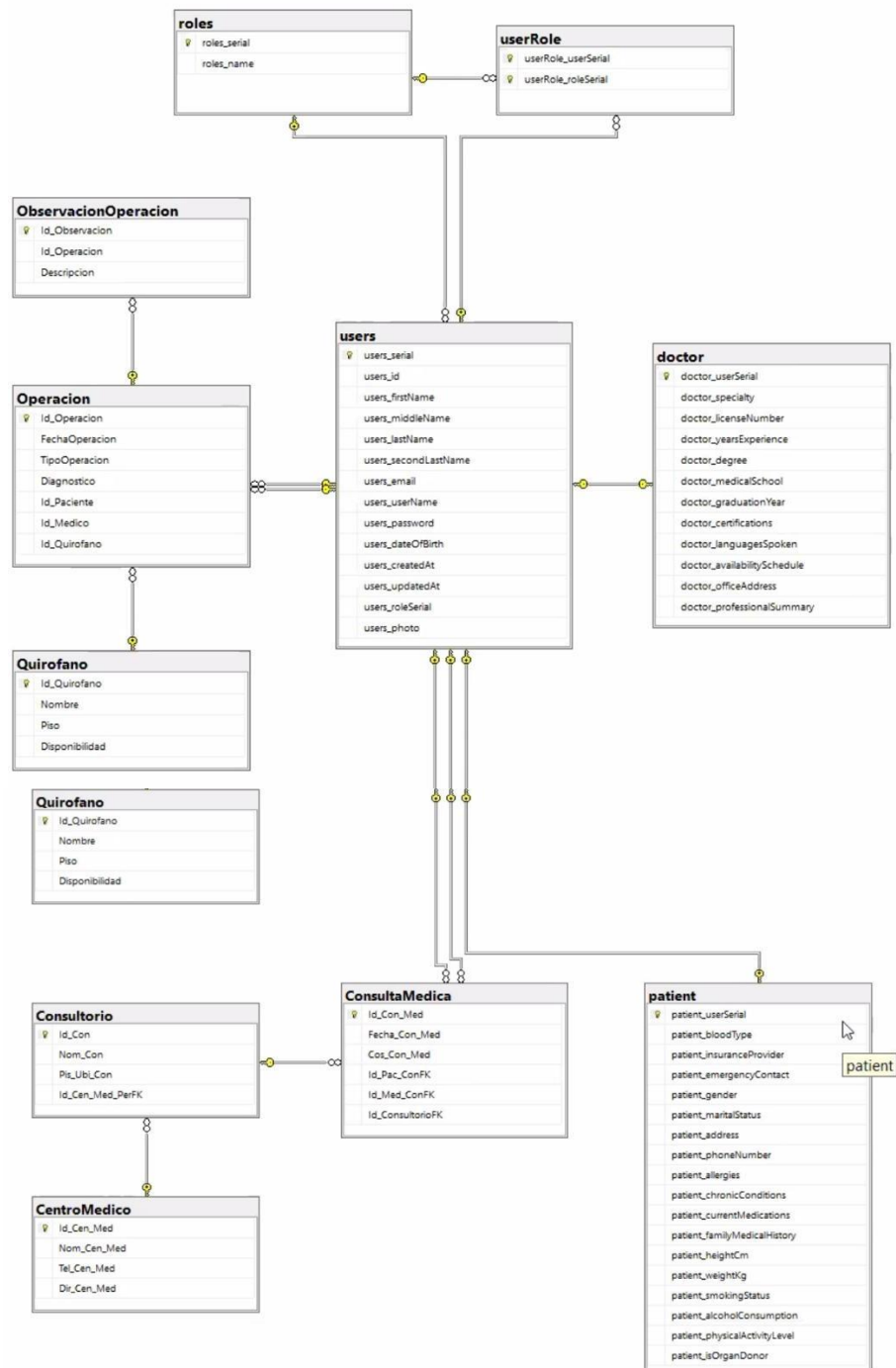
Este enfoque permite una arquitectura escalable y distribuida, HTML usando el rendimiento mediante balanceo de carga entre réplicas. Gracias a la replicación sincrónica, cualquier modificación en 1 de los nodos se sincroniza automáticamente con el demás punto. Por ejemplo, se validó la correcta replicación de las tablas, roles y sus registros en el nodo correspondiente al servidor.



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



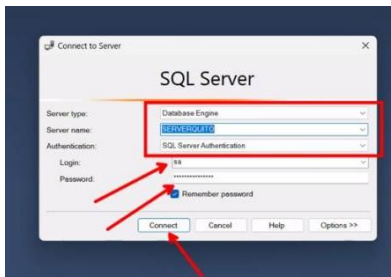
Modelo relacional:



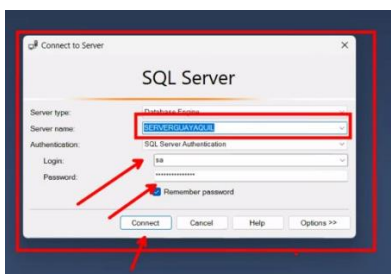


Paso 1: Conexión al servidor SQL Server

1. Inicie sesión en la instancia del servidor Quito (Maestro) a través de la interfaz gráfica de SQL Server.



2. Inicie sesión en la instancia del servidor Guayaquil a través de la interfaz gráfica de SQL Server.

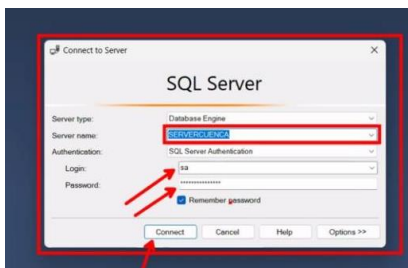




UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
CICLO ACADÉMICO: MARZO – JULIO 2025



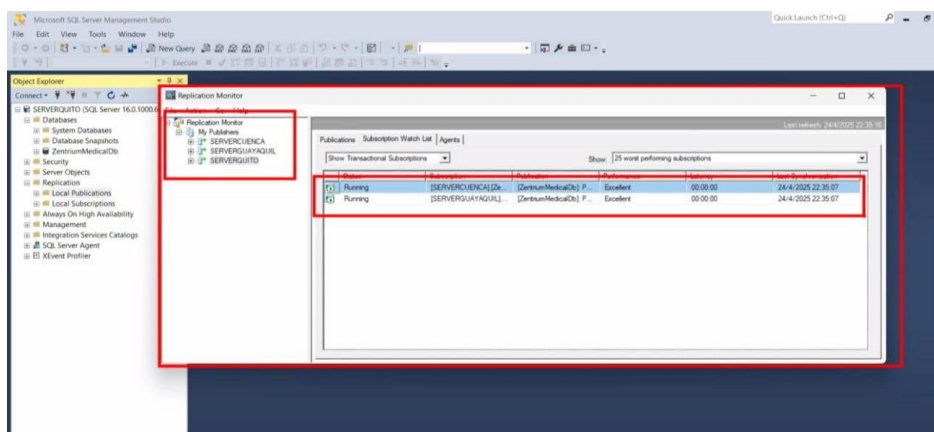
3. Inicie sesión en la instancia del servidor Guayaquil a través de la interfaz gráfica de SQL Server.



Paso 2: Verificación de la Conexión entre Servidores

Una vez que haya iniciado sesión en las instancias Guayaquil, Quito y Cuenca, verifique que los servidores se conecten entre sí para asegurar una comunicación adecuada. Esto es fundamental para el correcto funcionamiento de las transacciones distribuidas. [1]

- Los servidores Guayaquil, Quito y Cuenca deben estar conectados y sincronizando datos correctamente.
- Las publicaciones y suscripciones deben estar marcadas como Running con un estado Excelente o Good.





Paso 3: Crear base de datos

```
CREATE DATABASE ZentriumMedicalDb;  
GO  
USE ZentriumMedicalDb;  
GO
```

Paso 4: Crear tablas principales simplificadas para pruebas

-- Tabla users (usuarios del sistema)

```
CREATE TABLE users (  
    user_id CHAR(1) PRIMARY KEY,  
    nombre NVARCHAR(50)  
);
```

Tabla doctor (referenciada a users)

```
CREATE TABLE doctor (  
    doctor_id CHAR(1) PRIMARY KEY,  
    user_id CHAR(1),  
    especialidad NVARCHAR(50),  
    CONSTRAINT FK_doctor_user FOREIGN KEY (user_id) REFERENCES  
users(user_id)  
);
```

Tabla ConsultaMedica (referencia a doctor)

```
CREATE TABLE ConsultaMedica (  
    consulta_id CHAR(1) PRIMARY KEY,  
    doctor_id CHAR(1),  
    motivo NVARCHAR(100),  
    CONSTRAINT FK_consulta_doctor FOREIGN KEY (doctor_id) REFERENCES  
doctor(doctor_id)  
);
```

Paso 5: Insertar datos válidos

```
INSERT INTO users VALUES ('1', 'Dr. Pérez');  
INSERT INTO doctor VALUES ('A', '1', 'Cardiología');  
INSERT INTO ConsultaMedica VALUES ('X', 'A', 'Chequeo general');
```

Paso 6: Probar integridad referencial (esto falla si las claves no existen)

```
-- INSERT INTO doctor VALUES ('B', '9', 'Pediatría'); -- Falla: user_id '9' no existe
```

Paso 7: Crear una transacción con COMMIT

```
BEGIN TRAN;  
    INSERT INTO users VALUES ('2', 'Dra. Torres');  
    INSERT INTO doctor VALUES ('B', '2', 'Pediatría');  
COMMIT;
```

Paso 8: Crear una transacción con error y usar ROLLBACK

```
BEGIN TRAN;  
    INSERT INTO users VALUES ('3', 'Dr. Ruiz');  
    INSERT INTO doctor VALUES ('C', '9', 'Neurología'); -- user_id '9' NO existe  
ROLLBACK; -- Revertimos todo
```

Paso 9: Manejo de errores y abortos automáticos

```
SET XACT_ABORT ON;
```



```
BEGIN TRY
BEGIN TRAN;
    INSERT INTO users VALUES ('4', 'Dra. Gómez');
    INSERT INTO doctor VALUES ('D', '4', 'Dermatología');
COMMIT;
END TRY
BEGIN CATCH
    ROLLBACK;
    PRINT 'Error en la transacción: ' + ERROR_MESSAGE();
END CATCH;
```

Paso 10: Verificar datos

```
SELECT * FROM users;
SELECT * FROM doctor;
SELECT * FROM ConsultaMedica;
```

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

- Durante la ejecución de transacciones, el uso adecuado de instrucciones, Commit y Rollback. Resultó clave para controlar el estado de las operaciones punto. Estas herramientas permiten confirmar los cambios de revertirlos en casos de errores, manteniendo la coherencia de los datos.
- Se comprendió la importancia de las propiedades de atomicidad, consistencia, aislamiento y durabilidad como pilares fundamentales para asegurar la integridad de los datos en entornos distribuidos.

2.10 Recomendaciones

- Verifica que SQL Server esté correctamente instalado antes de iniciar las prácticas.
- Asegúrate de mantener la integridad referencial al realizar transacciones entre tablas relacionadas.
- Recuerda que toda transacción debe finalizar con COMMIT o ROLLBACK para garantizar un estado consistente de la base de datos.



- Considera la complejidad adicional al trabajar con sistemas transaccionales distribuidos y planifica cuidadosamente la replicación y fragmentación de datos.

2.11 Referencias bibliográficas

- [1] M. Build, "ROLLBACK TRANSACTION (Transact-SQL)," 02 01 2025. [Online]. Available: <https://learn.microsoft.com/es-es/sql/t-sql/language-elements/rollback-transaction-transact-sql?view=sql-server-ver16>. [Accessed 25 04 2025].