UNIVERSIDAD PERUANA LOS ANDES

FACULTAD INGENIERIA INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

MANUAL

SEGURIDAD Y CONTROL DE ACCESO

ASIGNATURA: BASE DE DATOS II

DOCENTE: FERNÁNDEZ BEJARANO RAUL

ESTUDIANTE: Bonifacio Hilario Erick

CÓDIGO: S01238F

HUANCAYO-2025

1. AUTENTICACIÓN SQL Y WINDOWS

Definición

La autenticación es el proceso mediante el cual SQL Server verifica la identidad del usuario antes de permitirle el acceso.
Existen dos tipos principales:

- **Autenticación de Windows:** utiliza las credenciales del sistema operativo, es más segura porque no almacena contraseñas dentro de SQL Server.
- **Autenticación de SQL Server:** requiere un usuario y contraseña propios del servidor, útil cuando no se maneja un dominio de Windows.

Diferencias

- **Windows:** usa la cuenta del sistema operativo y permite inicio de sesión sin necesidad de volver a escribir contraseña.
- **SQL Server:** requiere usuario y contraseña definidos dentro del motor SQL.
- **Windows:** más segura y fácil de administrar en redes corporativas.
- **SQL Server:** más flexible para conexiones externas o aplicaciones web.

Buenas prácticas

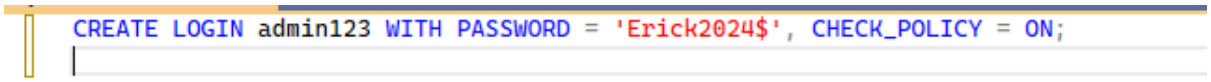
- Utilizar preferentemente autenticación de Windows en entornos empresariales.
- Si se usa autenticación SQL, establecer contraseñas seguras.
- Deshabilitar el inicio de sesión “sa” o cambiarle el nombre para evitar ataques.
- Restringir el número de inicios de sesión con permisos administrativos.

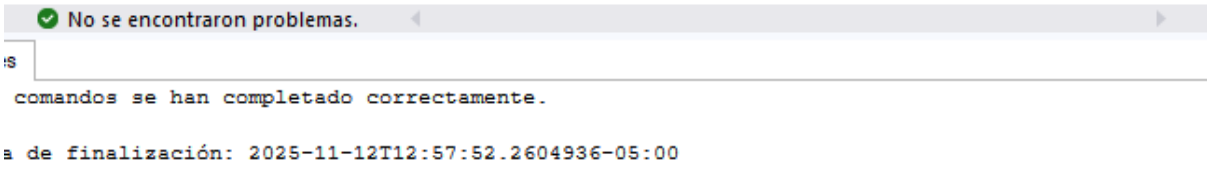
Paso a paso

1. Abrir **SQL Server Management Studio (SSMS)**.

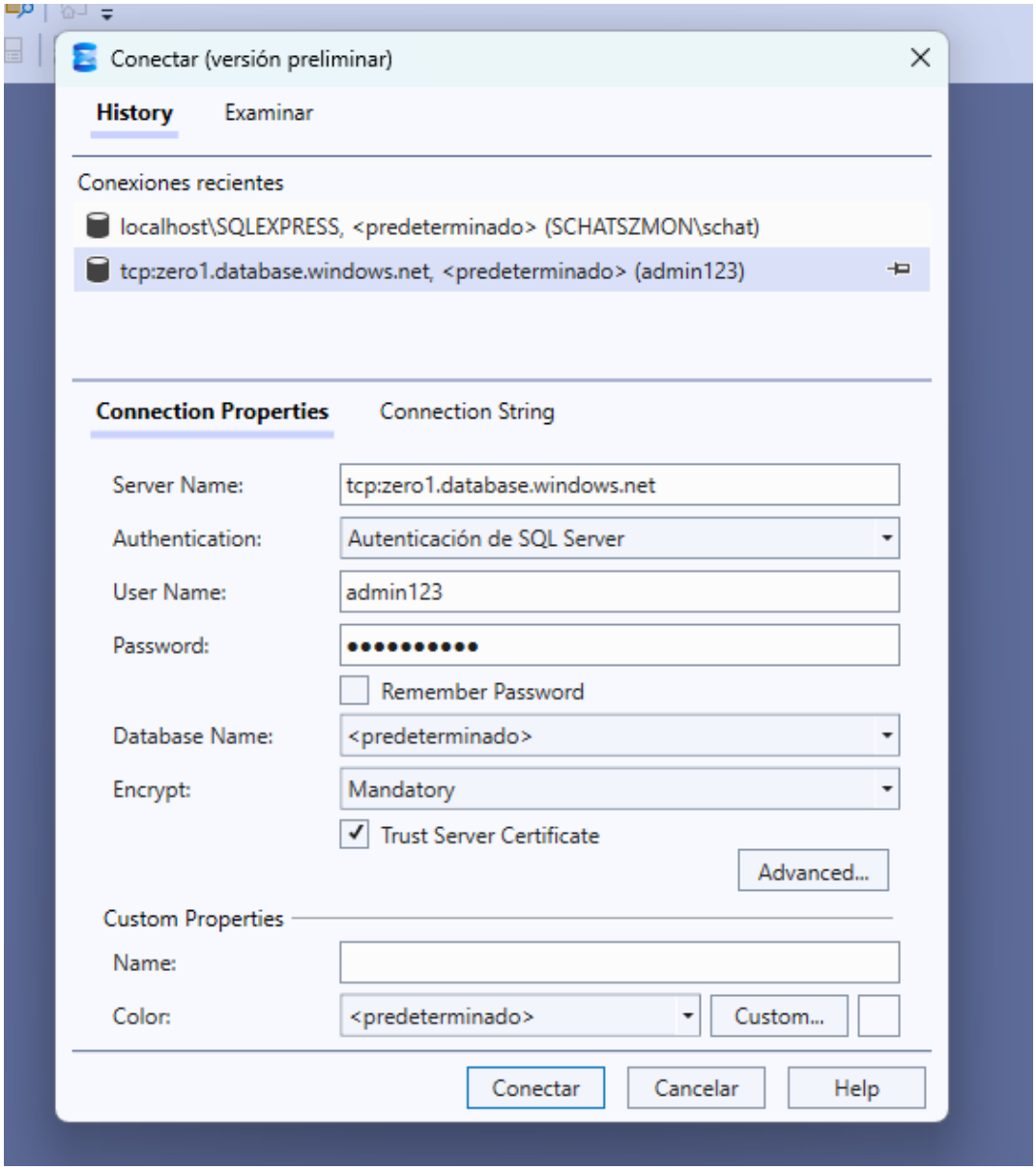
2. Conectarse al servidor con autenticación de Windows.
3. Para cambiar el modo de autenticación:
    a. Clic derecho sobre el servidor → *Propiedades*.
    b. Ir a la pestaña **Seguridad**.
    c. Seleccionar “Autenticación de Windows” o “SQL Server y Windows”.
4. Crear un nuevo inicio de sesión SQL:

```
CREATE LOGIN admin123 WITH PASSWORD = 'Erick2024$',
CHECK_POLICY = ON;
```





5. Probar conexión usando ese nuevo usuario.



Explicación

SQL Server autentica usuarios para evitar accesos no autorizados. Configurar correctamente este modo es clave para la seguridad del sistema. El paso a paso permite practicar ambos métodos y entender sus diferencias en escenarios reales.

2. CUENTAS DE SERVICIO Y CONFIGURACIÓN DEL SERVIDOR

Definición

Las cuentas de servicio son las que utiliza SQL Server para ejecutar sus procesos internos en el sistema operativo. Controlan la forma en que el motor accede a archivos, redes y otros servicios.

Buenas prácticas

- Usar cuentas de servicio **dedicadas** (no personales).
- No usar cuentas de administrador local.
- Aplicar el principio de **mínimo privilegio**.
- Cambiar contraseñas periódicamente.

Paso a paso

1. Abrir el **SQL Server Configuration Manager**.
2. En el panel izquierdo, elegir **SQL Server Services**.
3. Clic derecho sobre el servicio "SQL Server (MSSQLSERVER)" → *Propiedades*.
4. En la pestaña **Log On**, seleccionar "This Account" y especificar una cuenta de servicio con permisos limitados.
5. Reiniciar el servicio para aplicar los cambios.

Explicación

Definir correctamente la cuenta de servicio protege la base de datos de accesos indebidos al sistema operativo. Este control básico reduce los riesgos de intrusión o daño a los archivos de datos.

3. CREACIÓN DE ROLES FIJOS Y PERSONALIZADOS

Definición

Los roles son agrupaciones de permisos que simplifican la administración de usuarios. SQL Server incluye roles fijos (como `db_datareader`, `db_datawriter`, `db_owner`) y permite crear roles personalizados según las necesidades.

Buenas prácticas

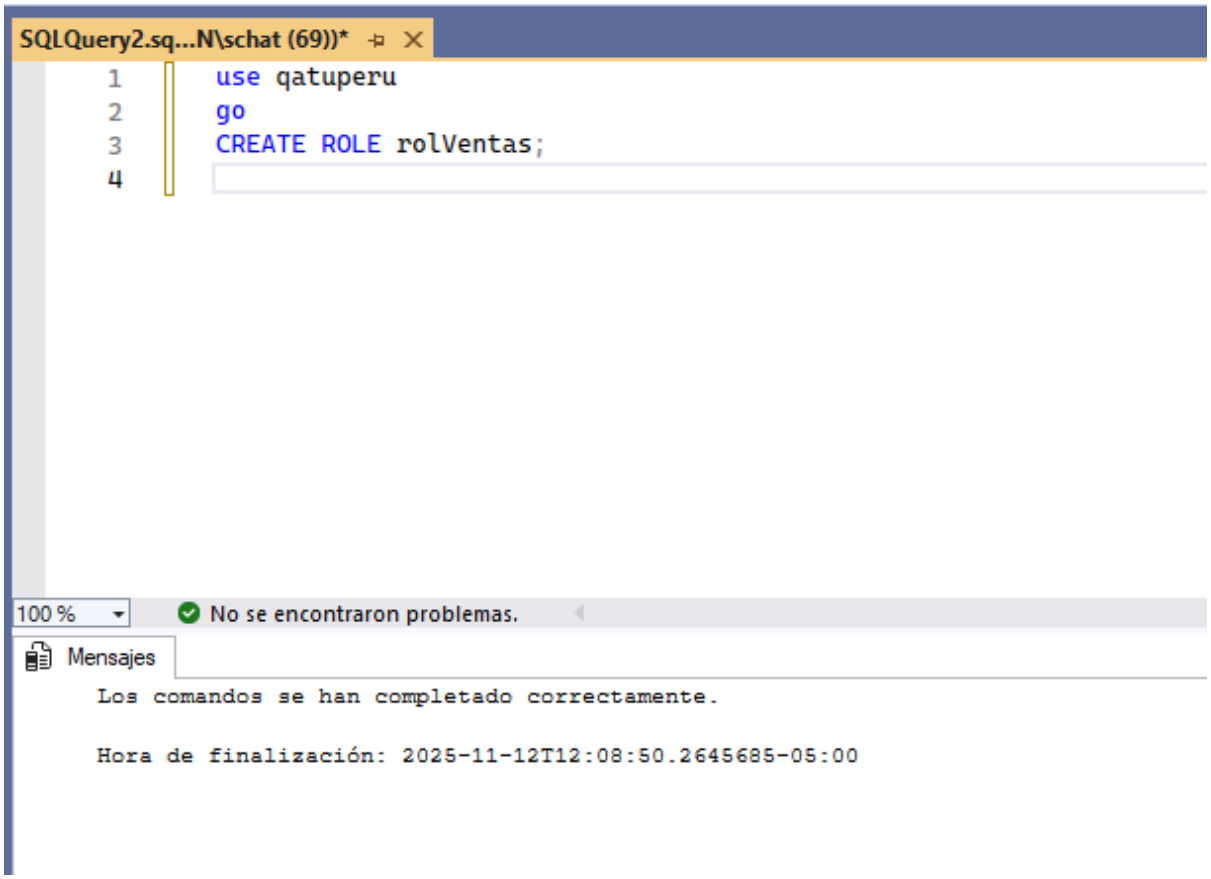
- No asignar permisos directamente a usuarios; usar roles.

- Evitar usar `db_owner` salvo en casos de administración.
- Revisar periódicamente los permisos de cada rol.

Paso a paso

1. Abrir **SSMS** y conectarse a la base de datos deseada.
2. Crear un rol personalizado:

```
CREATE ROLE rolVentas;
```

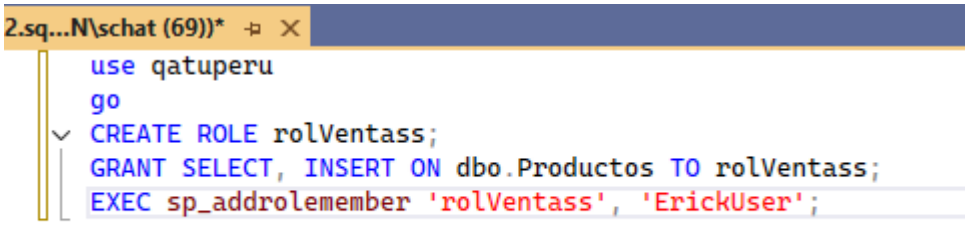


3. Otorgarle permisos:

```
GRANT SELECT, INSERT ON dbo.Productos TO rolVentas;
```

4. Asignar el rol a un usuario:

```
EXEC sp_addrolemember 'rolVentas', 'ErickUser';
```



Explicación

Usar roles facilita la administración de permisos al manejar grupos de usuarios con funciones similares. Esto evita errores y mejora la trazabilidad del control de acceso.

4. CONTROL DE ACCESO CON GRANT, DENY Y REVOKE