



**AURORA**  
**S.A**



**AURORA S.A.**

**BASES DE DATOS APLICADAS - 3641**

**TP INTEGRADOR – DOCUMENTACION**



**Universidad Nacional de La Matanza**

**Docente: Hnatiuk, Jair Ezequiel**

*Comisión 1353*

- Rodríguez Gonzalo (46418949)
- Francisco Vladimir (46030072)
- Vuono Gabriel (42134185)

## **Documentación General Del Proyecto**

### **Sobre la Instalación y Configuración de SQL Server**

**Configure las ubicaciones de almacenamiento:**

- **Base de datos:** SSD o almacenamiento de alto rendimiento en:  
***D:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\DATA***  
Las bases de datos del sistema (master, model, msdb), también se encuentran aquí.

- **Archivos de registro de la base de datos:** Disco separado con alta velocidad de escritura en:

***E:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\DATA***

**Archivos de registro del servidor (ERRORLOG):** mismo caso que el anterior, pero en la siguiente ruta:

***E:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Log***

- **Backups:** Unidad con espacio considerable y redundancia en:

***F:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup***

**Aclaración:** Los nombres de las particiones son totalmente ilustrativos, lo que no quita que los archivos deben distribuirse según las consideraciones mencionadas.

### **Configuración de Puertos y Protocolos**

**SQL Server Configuration Manager**, se accede a través de:

***C:\Windows\SysWOW64\SQLServerManager16.msc***

Nota: Si su sistema operativo se encuentra en otra partición o la ruta difiere por razones de configuración, ajuste la ruta en consecuencia.

- El protocolo TCP/IP se encuentra activado:  
Instancia predeterminada: Puerto 1433  
Instancia nombrada: Configurar puerto fijo
- El protocolo TLS/SSL se encuentra habilitado para garantizar conexiones seguras.

- Named pipes se encuentra habilitado, será necesario para realizar importaciones de archivos externos, su uso se detalla más adelante.

### Configuración de memoria

- **Memoria Mínima:** 25% de la memoria RAM; 16000 MB.
- **Memoria Máxima:** 80% de la memoria RAM (Para evitar un consumo excesivo); 51200 MB.

## Modo de autenticación

Modo mixto: (Windows + SQL Server Authentication)

## Disposición de los esquemas

- **Esquema 'Empresa':** contiene los objetos relacionados con las entidades que tienen una pertenencia con la empresa, como las tablas de sucursales y empleados.
- **Esquema 'Ventas':** contiene tablas relacionadas a ciertas entidades que intervienen a la hora de que se produzca una venta, como clientes, facturas y detalle de facturas.
- **Esquema 'Inventario':** en él se encuentran las tablas que alojan productos y sus respectivas líneas, el inventario con el que cuenta la empresa para competir en el mercado.
- **Esquema 'Utilidades':** un esquema utilizado para almacenar procedimientos funciones que fueron programadas para facilitar ciertas tareas, como vaciar las tablas rápidamente a la hora de hacer pruebas.
- **Esquema 'Reportes':** contiene los procedimientos destinados a la generación de reportes temporales en formato XML.
- **Esquema 'Seguridad':** contiene algunos procedimientos destinados a tareas de seguridad, como encriptación y generación de notas sólo por determinados roles.

## Nomenclaturas

- Nombres de tablas como sustantivos singulares.
- La utilización de los prefijos PK y FK para la creación de restricciones en claves primarias y foráneas.
- Las tablas, esquemas y procedimientos comienzan con letra mayúscula.
- Las campos y variables declaradas comienzan en minúscula.
- Los procedimientos cuentan con el sufijo \_sp.
- Se emplea la sentencia RAISERROR para expresar errores en las validaciones.
- Los índices se escriben en minúscula, con el prefijo ix\_, seguido de una breve descripción del campo que indexan.

## Procedimientos y configuración de Importación

El sistema permite la importación de datos desde archivos CSV y Excel, empleando tablas temporales y SQL dinámico, permitiendo transformar, limpiar y normalizar los datos antes de insertarlos en las tablas definitivas. Para realizar estos procedimientos es necesaria una previa configuración del motor SQL Server

1. Habilidad de Consultas Distribuidas Ad Hoc
  - EXEC sp\_configure 'show advanced options', 1;
  - EXEC sp\_configure 'Ad Hoc Distributed Queries', 1;
2. Configuración del Proveedor OLE DB para Excel:
  - EXEC sp\_MSset\_oledb\_prop 'Microsoft.ACE.OLEDB.12.0', 'AllowInProcess', 1;
  - EXEC sp\_MSset\_oledb\_prop 'Microsoft.ACE.OLEDB.12.0', 'DynamicParameters', 1;
3. Activación de Named Pipes:
  - Se habilita en SQL Server Configuration Manager, dentro de Configuración de Protocolo.

## Generación de datos faltantes

- Se generaron combinaciones de valores (ej. CUIL de empleados).
- Se definieron valores fijos para clientes.
- Se utilizaron funciones como CHECKSUM() en lugar de RAND() para valores aleatorios en inserciones masivas.

## Validaciones con RAISERROR

El segundo argumento que observará en la sentencia RAISERROR será repetidamente el entero 16. Este argumento remite al nivel de severidad del error; 16 corresponde a errores de usuario, como lo podrían violaciones de restricciones o manejo de datos inválidos. Es por ello que lo empleamos en todas las validaciones de los procedimientos o funciones codificadas.

Bibliografía: <https://learn.microsoft.com/es-es/sql/t-sql/language-elements/raiserror-transact-sql?view=sql-server-ver16>

## Orden en las importaciones

La ejecución de los procedimientos destinados importaciones masivas de archivos no es arbitraria; debe respetar un orden por las evidentes conexiones existentes entre los datos almacenados en las tablas. Véase los siguientes casos;

- Previo a la importación del catálogo es necesario que la tabla línea producto se encuentre cargada para poder normalizar correctamente las clasificaciones de los productos, de acuerdo a como fue decidido.
- Al importar las ventas registradas, para garantizar la validez de los datos es precondition que las siguientes tablas se encuentren cargadas:
  - ✓ Cliente
  - ✓ Empleado
  - ✓ Sucursal
  - ✓ Producto

## Inconsistencias en las equivalencias de productos

En el archivo “productos\_importados” aparecen determinadas categorías de productos que no coinciden con lo existente en las equivalencias encontradas en “Información\_complementaria”, pero que si pueden ser agrupadas dentro de ellas empleando algo de sentido común. Este criterio fue empleado de manera personal por el equipo de trabajo, un ejemplo sería: ‘Carnes’, ‘Frutas/verduras’, ‘Pescado/Marisco’ agrupados todos en la categoría padre ‘Frescos’.

## Intercalación

La intercalación elegida al momento de la creación de la base de datos del proyecto fue: *Modern\_Spanish\_CS\_AS*, que difiere de la predeterminada por la instancia en la instalación *Modern\_Spanish\_CS\_AI*.

Las tablas temporales utilizadas de apoyo para las importaciones masivas emplearan esta última intercalación por almacenarse en la base de datos del sistema tempdb; por ende, en las sentencias de creación de las tablas se empleó de manera explícita la sentencia COLLATE *Modern\_Spanish\_CS\_AS* para posibilitar las comparaciones de texto en sentencias del tipo WHERE o juntas.

## Nivel de aislamiento

Se emplean transacciones para englobar cada importación de archivos como una unidad lógica. El nivel de aislamiento elegido en Read Committed, a causa de las siguientes razones:

### a) Consistencia sin excesivo bloqueo

READ COMMITTED es el nivel por defecto en SQL Server y garantiza que cada transacción solo vea datos que han sido confirmados por otras transacciones. Esto resulta ideal para operaciones de importación, ya que cada archivo se procesa en su propia transacción, evitando que se lean datos “sucios” mientras se está realizando la carga.

### b) Performance en operaciones de carga

READ COMMITTED ofrece un equilibrio al proteger la integridad de los datos sin comprometer demasiado la concurrencia y el rendimiento, algo crucial cuando se manejan archivos con decenas de miles de registros.

### c) Manejo de errores

Dado que cada importación se encapsula en una transacción con un bloque TRY-CATCH, en caso de error, la transacción se revierte y se evita que datos incompletos o inconsistentes se comprometan en la base de datos.

## Índices

La creación de índices no clúster tiene el objetivo de optimizar las consultas, ya que permiten que el motor de base de datos acceda a los registros de forma más rápida sin tener que recorrer la totalidad de la tabla. Esto es

fundamental cuando se espera que el volumen de datos crezca significativamente, pues en escenarios con grandes cantidades de registros, un índice bien diseñado puede reducir drásticamente los tiempos de respuesta en búsquedas y filtros.

Sin embargo, en entornos de prueba o cuando se trabaja con conjuntos de datos pequeños (como el caso de los juegos de prueba diseñados), el beneficio de los índices no es tan evidente. De hecho, puede llegar a ser contraproducente por su costo en operaciones de escritura y lectura. La decisión de implementarlos se sustenta en una estrategia a futuro: Si en algún momento se integran archivos con mayores volúmenes de datos, contar con índices no clúster permitirá que las consultas sigan siendo eficientes, evitando exploraciones completas de tabla que podrían ralentizar considerablemente el rendimiento.

## **Función NEWID y comportamiento en SQL Server**

En SQL Server, la función NEWID() se utiliza comúnmente para obtener un ordenamiento aleatorio, por ejemplo, al seleccionar un registro de forma “aleatoria” usando TOP 1.

Justamente con el fin de obtener un registro aleatorio, pero que cumpla con determinadas condiciones, se emplea la función en el proyecto. Ocurre que cuando se emplea ORDER BY NEWID() en una subconsulta que forma parte de un CROSS APPLY dentro de un INSERT INTO ... SELECT, puede ocurrir que el motor de SQL Server evalúe esa expresión solo una vez en lugar de re-evaluarla para cada fila de la consulta externa. Esto se debe a cómo funciona la compilación y optimización del plan de ejecución.

Para solucionar este inconveniente simplemente agregamos arbitrariamente un campo adicional en la sentencia ORDER BY. Esto se podrá observar explícito en el código fuente (véase, por ejemplo, en el procedimiento de importación que carga la tabla factura).

**Bibliografía:** <https://learn.microsoft.com/es-es/sql/t-sql/functions/newid-transact-sql?view=sql-server-ver16>

## **Políticas de respaldo para ventas y reportes**

Se ha definido una estrategia de respaldo que garantiza la disponibilidad y protección de los datos en caso de fallos, ataques cibernéticos o corrupción de la base de datos.

## **Tipos de Respaldo**

### **Respaldo Diario**

- Se realiza todas las noches al finalizar el día operativo.
- Se almacena en un servidor seguro dentro de la red corporativa.
- Permite recuperar las ventas del día en caso de pérdida de datos.

### **Respaldo Semanal**

- Se ejecuta cada domingo para almacenar toda la información de la semana.
- Se guarda en un almacenamiento seguro de mayor capacidad.
- Sirve como un punto de recuperación más estable, evitando la restauración de múltiples respaldos diarios.

## **Ubicación y Seguridad de los Respaldos**

- Los respaldos diarios se almacenan en servidores internos de la empresa.
- Los respaldos semanales se almacenan en la nube o en un centro de datos externo, protegiendo la información ante desastres físicos en la oficina (incendios, robos, fallos de hardware).

### **Medidas de seguridad:**

- Todos los respaldos están cifrados para evitar accesos no autorizados.
- Solo administradores de bases de datos pueden restaurar los respaldos.
- Periódicamente se verifica que los respaldos sean funcionales y puedan ser restaurados en caso de emergencia.

### **Política de Retención**

- Los respaldos diarios se conservan por 7 días
- Los respaldos semanales por 90 días.

Una vez que se supera el tiempo de retención, los respaldos más antiguos se eliminan para optimizar el almacenamiento, manteniendo únicamente los más recientes.

De esta forma, se asegura que nunca se pierda más de un día de ventas, reduciendo el impacto de un fallo, se tiene la capacidad de restaurar información exacta en caso de errores humanos, ataques o corrupción de datos y se cumplen estándares de seguridad y auditoría requeridos para la gestión de información financiera.



## Estrategia de Backup y Recuperación de Base de Datos

### Modelo de recuperación

Se ha optado por el modelo **Bulk-Logged**, ya que el software maneja importaciones masivas de archivos externos, algunos con más de 50,000 registros. Este modelo permite que cada importación se registre como una única transacción, reduciendo el crecimiento excesivo del log de transacciones y mejorando el rendimiento del sistema. El modelo full registraría una transacción por cada uno de los registros que son insertados.

Las ventajas clave de esta elección son:

- Registra todas las transacciones, incluidas las importaciones masivas, como una unidad lógica.
- Permite restauraciones a un punto en el tiempo (siempre que se cuente con copias de seguridad adecuadas).
- Minimiza el riesgo de pérdidas significativas de datos en caso de fallos.

### Tipos de Backup

- Backup Completo
  - Se ejecuta semanalmente los domingos a la medianoche.
  - Contiene una copia completa de la base de datos.
  - Se almacena en servidores seguros y en la nube o un centro de datos externo.
- Backup Diferencial
  - Se realiza diariamente al finalizar la jornada laboral.
  - Guarda cambios desde el último backup completo.
  - Se almacena en servidores internos y unidades externas.
- Backup del Log de Transacciones
  - Se ejecuta cada 30 minutos.
  - Permite restaurar la base de datos en un punto exacto.
  - Evita crecimiento excesivo del log de transacciones.

### Plan de Restauración

- Fallo total: Restaurar último backup completo, aplicar diferencial y luego log de transacciones.

- Pérdida parcial: Restaurar backup en una base de prueba y extraer datos afectados.
- Recuperación punto en el tiempo: Aplicar logs hasta antes del error.

**Seguridad y Retención de Backups**

- Cifrado de backups para evitar accesos no autorizados.
- Restauración solo por administradores de bases de datos.
- Pruebas periódicas de recuperación.

**Política de Retención**

- Backups completos: 90 días.
- Backups diferenciales: 7 días.
- Backups de logs de transacciones: 48 horas tras un backup exitoso.

Con esta estrategia, Aurora S.A. minimiza la pérdida de datos y garantiza la recuperación rápida ante incidentes.