

Especificación de la arquitectura técnica, el modelo de datos y las decisiones de diseño.

**DISEÑO TÉCNICO Y ESTRUCTURA DE DATABASE**

Contenido

[Introducción 2](#_Toc193035288)

[Especificaciones del Sistema 2](#_Toc193035289)

[Arquitectura del Sistema 2](#_Toc193035290)

[Plataforma Tecnológica e Integraciones 3](#_Toc193035291)

[Diseño de la base de datos 4](#_Toc193035292)

[Diagrama Entidad Relación (ERD) 4](#_Toc193035293)

[Modelo Lógico 4](#_Toc193035294)

[El modelo lógico está diseñado en SQL Server y las tablas se organizan en esquemas según su funcionalidad: 4](#_Toc193035295)

[Normalización 5](#_Toc193035296)

[Diccionario de datos 5](#_Toc193035297)

[Procedimientos y funcionalidades 9](#_Toc193035298)

[Procedimientos ABM 9](#_Toc193035299)

[Procedimientos para Sucursales 9](#_Toc193035300)

[Procedimientos para Empleados 9](#_Toc193035301)

[Procedimientos para Clientes 9](#_Toc193035302)

[Procedimientos para Inventario y Productos 10](#_Toc193035303)

[Procedimientos para Ventas 10](#_Toc193035304)

[Procedimientos de Seguridad 11](#_Toc193035305)

[Otros procedimientos 11](#_Toc193035306)

[Juegos de Prueba 12](#_Toc193035307)

[Justificaciones Técnicas 13](#_Toc193035308)

[Motor SQL Server 13](#_Toc193035309)

[Índices 13](#_Toc193035310)

[Características de código 14](#_Toc193035311)

[Nomenclatura 14](#_Toc193035312)

[Consideraciones a la hora de importar 14](#_Toc193035313)

# Introducción

Este documento proporciona una descripción detallada de la arquitectura, el diseño y la implementación del Sistema de Gestión de Ventas para Aurora S.A. Está dirigido a desarrolladores, administradores de bases de datos y personal técnico involucrado en la implementación y mantenimiento del sistema.

El documento abarca todos los aspectos técnicos del sistema, incluyendo la arquitectura de la base de datos, procedimientos almacenados, seguridad, optimización y políticas de respaldo. Se enfoca en la implementación técnica de los requerimientos especificados en el documento SRS.

El sistema está implementado como una solución de base de datos centralizada en Microsoft SQL Server, utilizando T-SQL para la implementación de la lógica de negocio a través de procedimientos almacenados, funciones y triggers. El diseño sigue principios de modularidad, seguridad y rendimiento.

# Especificaciones del Sistema

## Arquitectura del Sistema

El sistema se basa en una arquitectura cliente-servidor centralizada con Microsoft SQL Server como motor de base de datos principal. La arquitectura se compone de una capa de almacenamiento, una capa lógica y una capa de integración y seguridad.

La capa de almacenamiento comprende el conjunto de objetos de base de datos que residen físicamente en el servidor. Se incluyen tablas, índices, restricciones, claves primarias y foráneas, y demás objetos relacionados con el almacenamiento de datos. Utilizamos una organización mediante esquemas para agrupar lógicamente los objetos según su funcionalidad.

La capa lógica de negocio se implementa a través de procedimientos almacenados, funciones y triggers. Esta capa se encarga de las validaciones, el manejo de transacciones y la transformación e inserción de datos.

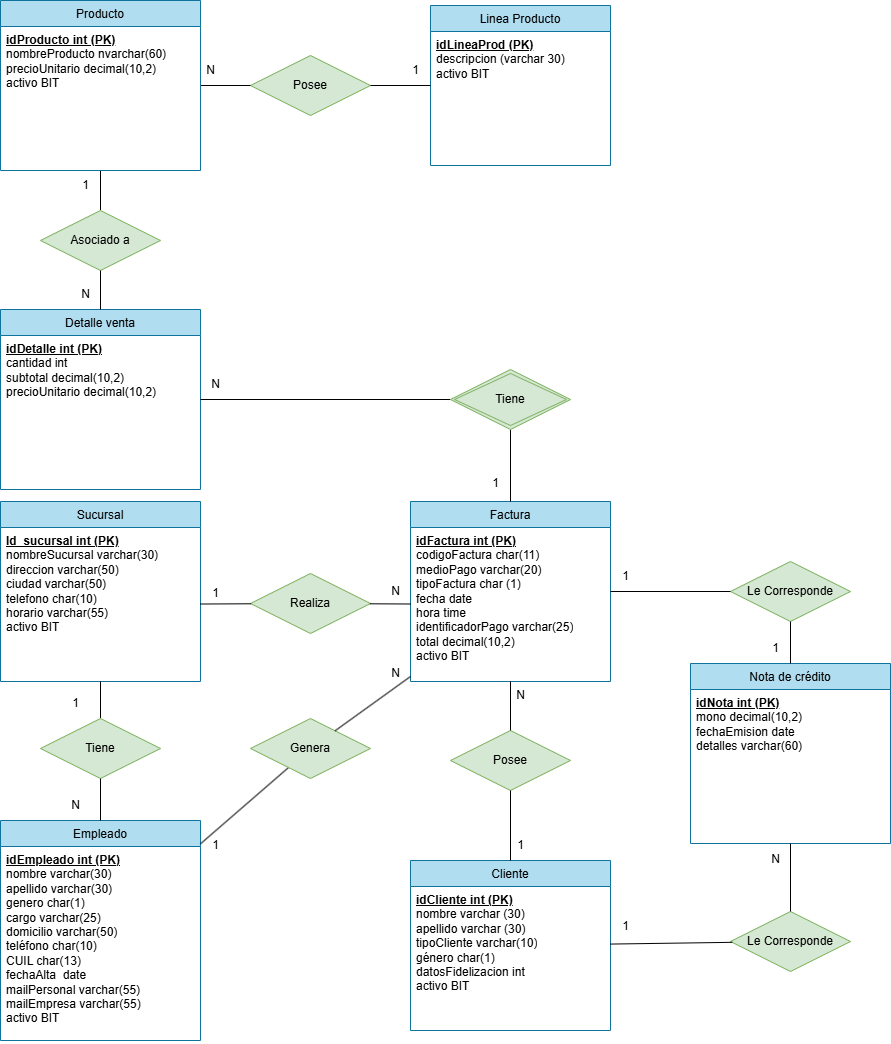
La capa de integración y seguridad se encarga de la interacción con fuentes externas (por ejemplo, importaciones masivas de archivos CSV y Excel) y de aplicar las políticas de seguridad del sistema. Incluye la configuración de proveedores OLE DB, la ejecución de consultas distribuidas y la implementación de roles y encriptación de datos sensibles.

## Plataforma Tecnológica e Integraciones

* Motor de Base de Datos: Microsoft SQL Server (versión más reciente disponible)
* Lenguajes de Programación: T-SQL para procedimientos almacenados y funciones
* Importación de Datos: Integración con archivos XLXS y CSV para la carga de catálogos.
* Exportación de Reportes: Generación de reportes en formato XML.

# Diseño de la base de datos

## Diagrama Entidad Relación (ERD)



## Modelo Lógico

## El modelo lógico está diseñado en SQL Server y las tablas se organizan en esquemas según su funcionalidad:

* **Esquema Empresa**: Contiene los objetos referidos a recursos humanos y materiales referidos al funcionamiento de la empresa. Engloba las tablas de Empleado y Sucursal.
* **Esquema Ventas**: Contiene los objetos referidos al procedimiento de venta. Engloba las tablas de Cliente, Factura, DetalleVenta y NotaCredito.
* **Esquema Inventario**: Contiene los objetos referidos al inventario de productos de la empresa. Engloba las tablas de Producto y LineaProducto.
* **Esquema Utilidades**: Contiene todos los componentes de software de apoyo, como funciones o procedimientos utilizados para resultados secundarios o auxiliares.
* **Esquema Reportes**: Contiene todos los componentes de software utilizados en la creación de reportes.
* **Esquema ‘Seguridad’**: Contiene todos los componentes de software destinados a tareas de seguridad.

## Normalización

Las tablas están diseñadas siguiendo los principios de la Tercera Forma Normal (3FN) para minimizar la redundancia y mantener la integridad de los datos.

Se ha aplicado desnormalización en las tablas de factura y detalle de factura al incluir el precio unitario en el detalle, para mantener un registro histórico del precio al momento de la venta, independientemente de cambios posteriores en el catálogo.

## Diccionario de datos

**Tabla: Empleados**

| **Columna** | **Tipo de Dato** | **Descripción** | **Restricciones** |
| --- | --- | --- | --- |
| IdEmpleado | INT | Identificador único del empleado | PK, NOT NULL |
| Nombre | VARCHAR(50) | Nombre del empleado | NOT NULL |
| Apellido | VARCHAR(50) | Apellido del empleado | NOT NULL |
| CUIL | VARCHAR(20) | CUIL del empleado (encriptado) | NOT NULL, UNIQUE |
| Domicilio | VARCHAR(200) | Domicilio del empleado (encriptado) | NOT NULL |
| Telefono | VARCHAR(20) | Teléfono del empleado (encriptado) | NOT NULL |
| Email | VARCHAR(100) | Email del empleado | NULL |
| FechaAlta | DATE | Fecha de ingreso | NOT NULL |
| Estado | BIT | Estado del empleado (1=Activo, 0=Inactivo) | NOT NULL, DEFAULT 1 |
| IdSucursal | INT | Sucursal a la que pertenece | FK, NOT NULL |
| IdRol | INT | Rol asignado al empleado | FK, NOT NULL |

**Tabla: Sucursales**

| **Columna** | **Tipo de Dato** | **Descripción** | **Restricciones** |
| --- | --- | --- | --- |
| IdSucursal | INT | Identificador único de la sucursal | PK, NOT NULL |
| Nombre | VARCHAR(100) | Nombre de la sucursal | NOT NULL, UNIQUE |
| Direccion | VARCHAR(200) | Dirección física | NOT NULL |
| Ciudad | VARCHAR(100) | Ciudad donde está ubicada | NOT NULL |
| Telefono | VARCHAR(20) | Teléfono de contacto | NOT NULL |
| FechaApertura | DATE | Fecha de apertura | NOT NULL |

**Tabla: LineasProducto**

| **Columna** | **Tipo de Dato** | **Descripción** | **Restricciones** |
| --- | --- | --- | --- |
| IdLinea | INT | Identificador único de la línea | PK, NOT NULL |
| Nombre | VARCHAR(100) | Nombre de la línea de producto | NOT NULL, UNIQUE |
| Descripcion | VARCHAR(200) | Descripción de la línea | NULL |

**Tabla: Productos**

| **Columna** | **Tipo de Dato** | **Descripción** | **Restricciones** |
| --- | --- | --- | --- |
| IdProducto | INT | Identificador único del producto | PK, NOT NULL |
| Codigo | VARCHAR(50) | Código de producto | NOT NULL, UNIQUE |
| Nombre | VARCHAR(200) | Nombre del producto | NOT NULL |
| Descripcion | VARCHAR(500) | Descripción del producto | NULL |
| PrecioUnitario | DECIMAL(10,2) | Precio unitario actual | NOT NULL |
| IdLinea | INT | Línea de producto a la que pertenece | FK, NOT NULL |
| FechaAlta | DATETIME | Fecha de alta en el sistema | NOT NULL, DEFAULT GETDATE() |
| Estado | BIT | Estado del producto (1=Activo, 0=Inactivo) | NOT NULL, DEFAULT 1 |
| Origen | VARCHAR(50) | Archivo de origen del producto | NOT NULL |

**Tabla: Facturas**

| **Columna** | **Tipo de Dato** | **Descripción** | **Restricciones** |
| --- | --- | --- | --- |
| IdFactura | INT | Identificador único de la factura | PK, NOT NULL |
| NumeroFactura | VARCHAR(20) | Número de factura | NOT NULL, UNIQUE |
| IdTipoFactura | CHAR(1) | Tipo de factura | FK, NOT NULL |
| Fecha | DATETIME | Fecha y hora de emisión | NOT NULL |
| IdCliente | INT | Cliente asociado | FK, NOT NULL |
| IdEmpleado | INT | Empleado que realizó la venta | FK, NOT NULL |
| IdSucursal | INT | Sucursal donde se realizó | FK, NOT NULL |
| IdMedioPago | INT | Medio de pago utilizado | FK, NOT NULL |
| MontoTotal | DECIMAL(12,2) | Monto total de la factura | NOT NULL |
| Estado | VARCHAR(20) | Estado de la factura (Pagada, Anulada, etc.) | NOT NULL, DEFAULT 'Pagada' |

**Tabla: DetallesFactura**

| **Columna** | **Tipo de Dato** | **Descripción** | **Restricciones** |
| --- | --- | --- | --- |
| IdDetalle | INT | Identificador único del detalle | PK, NOT NULL |
| IdFactura | INT | Factura a la que pertenece | FK, NOT NULL |
| IdProducto | INT | Producto vendido | FK, NOT NULL |
| Cantidad | INT | Cantidad vendida | NOT NULL |
| PrecioUnitario | DECIMAL(10,2) | Precio unitario al momento de la venta | NOT NULL |
| Subtotal | DECIMAL(12,2) | Subtotal (Cantidad \* PrecioUnitario) | NOT NULL |

**Tabla: NotasCredito**

| **Columna** | **Tipo de Dato** | **Descripción** | **Restricciones** |
| --- | --- | --- | --- |
| IdNotaCredito | INT | Identificador único de la nota de crédito | PK, NOT NULL |
| NumeroNC | VARCHAR(20) | Número de nota de crédito | NOT NULL, UNIQUE |
| IdFactura | INT | Factura asociada | FK, NOT NULL |
| Fecha | DATETIME | Fecha y hora de emisión | NOT NULL |
| IdEmpleado | INT | Supervisor que generó la nota | FK, NOT NULL |
| Motivo | VARCHAR(500) | Motivo de la devolución | NOT NULL |
| MontoTotal | DECIMAL(12,2) | Monto total de la nota | NOT NULL |

**Tabla: Clientes**

| **Columna** | **Tipo de Dato** | **Descripción** | **Restricciones** |
| --- | --- | --- | --- |
| IdCliente | INT | Identificador único del cliente | PK, NOT NULL |
| Nombre | VARCHAR(50) | Nombre del cliente | NOT NULL |
| Apellido | VARCHAR(50) | Apellido del cliente | NOT NULL |
| Genero | VARCHAR(10) | Género del cliente | NULL |
| Ciudad | VARCHAR(100) | Ciudad de residencia | NULL |
| Telefono | VARCHAR(20) | Teléfono de contacto | NULL |
| Email | VARCHAR(100) | Email del cliente | NULL |
| IdTipoCliente | INT | Tipo de cliente | FK, NOT NULL |
| FechaRegistro | DATE | Fecha de registro | NOT NULL, DEFAULT GETDATE() |
| PuntosAcumulados | INT | Puntos del programa de fidelización | NOT |

# Procedimientos y funcionalidades

A continuación se presenta una lista de los procedimientos desarrollados para el proyecto, junto con una breve explicación de su funcionamiento:

## Procedimientos ABM

### Procedimientos para Sucursales

* **Empresa.InsertarSucursal\_sp:** Permite insertar una nueva sucursal en la base de datos. Valida los datos ingresados (por ejemplo, formato del teléfono) y registra la sucursal en el esquema *Empresa*.
* **Empresa.ActualizarSucursal\_sp:** Actualiza los datos de una sucursal existente. Se verifica la existencia de la sucursal (y su estado activo) antes de proceder a la modificación.
* **Empresa.EliminarSucursal\_sp:** Realiza un borrado l\u00f3gico de la sucursal, actualizando el estado a inactivo. Esto permite mantener un historial de operaciones sin eliminar la información por completo.

### Procedimientos para Empleados

* **Empresa.InsertarEmpleado\_sp:** Inserta un nuevo empleado validando datos críticos (CUIL, teléfono, email, etc.). Se verifican las restricciones (como la existencia previa de legajo) y se aplican validaciones. Se incluye encriptación de datos sensibles.
* **Empresa.ActualizarEmpleado\_sp:** Actualiza la información de un empleado existente, aplicando las mismas validaciones que en la inserción. Se asegura la integridad referencial con la sucursal a la que pertenece.
* **Empresa.EliminarEmpleado\_sp:** Realiza el borrado lógico de un empleado, marcando su registro como inactivo en lugar de eliminarlo, lo que permite mantener la trazabilidad.

### Procedimientos para Clientes

* **Ventas.InsertarCliente\_sp:** Registra un nuevo cliente en el sistema, verificando el formato de los datos y asignando valores por defecto (por ejemplo, para el programa de fidelización.
* **Ventas.ActualizarCliente\_sp:** Permite la modificación de los datos del cliente, validando que la información sea consistente.
* **Ventas.EliminarCliente\_sp:** Realiza un borrado lógico del cliente, conservando la información para informes históricos.

### Procedimientos para Inventario y Productos

* **Inventario.InsertarLineaProducto\_sp:** Inserta nuevas líneas de producto en el catálogo, garantizando que no se dupliquen y normalizando la información.
* **Inventario.InsertarProducto\_sp:** Permite la inserción de nuevos productos, asociándolos a una línea de producto existente y validando el precio unitario y otros datos.
* **Inventario.ActualizarProducto\_sp:** Actualiza la información de un producto, permitiendo modificaciones en nombre, precio o categoría.
* **Inventario.EliminarProducto\_sp:** Realiza un borrado lógico de un producto para preservar el historial de ventas.

### Procedimientos para Ventas

* **Ventas.InsertarFactura\_sp:** Registra una factura nueva en el sistema, asociando datos de cliente, empleado y sucursal. Se valida el formato del código de factura y se calcula el total.
* **Ventas.ActualizarFactura\_sp:** Permite la actualización de los datos de una factura existente, con validaciones correspondientes.
* **Ventas.EliminarFactura\_sp:** Realiza un borrado lógico de la factura, manteniendo la integridad de los registros históricos.
* **Ventas.InsertarDetalleVenta\_sp:** Inserta el detalle de productos asociados a una factura, calculando subtotales y relacionando cada producto con su respectivo registro en el catálogo.
* **Ventas.ActualizarDetalleVenta\_sp** y **Ventas.EliminarDetalleVenta\_sp**: Permiten la modificación o eliminación de los detalles de venta, asegurando la coherencia de los totales y la integridad referencial.

### Procedimientos de Seguridad

* **Seguridad.GenerarNotaCredito\_sp:** Permite generar una nota de crédito para una factura pagada. Este procedimiento está restringido para usuarios con el rol Supervisor, validando que la factura esté activa antes de generar la nota.
* **Seguridad.EncriptarEmpleado\_sp:** Procedimiento para encriptar datos sensibles de empleados (como CUIL, domicilio, teléfono y email personal), actualizando la estructura de la tabla para almacenar los datos en forma segura. Se incluyen pasos para agregar las columnas encriptadas, actualizar la información y renombrar las columnas para mantener la nomenclatura original.

## Otros procedimientos

* **Inventario.CargarProductosCatalogoCSV\_sp:** Este procedimiento importa el catálogo de productos desde un archivo CSV y un archivo de equivalencias (Excel). Utiliza tablas temporales para cargar y transformar los datos, deduplicando registros y normalizando las categorías. Posteriormente, inserta los productos en la tabla definitiva, asignando el precio unitario ajustado por el valor del dólar.
* **Inventario.CargarProductosElectronicos\_sp:** Se encarga de importar productos electrónicos desde un archivo Excel. Verifica la existencia de la línea de producto “Electrónico” y, si no existe, la crea. Luego carga los datos, los transforma (aplicando el factor de conversión según el valor del dólar) y los inserta en la tabla de productos.
* **Inventario.CargarProductosImportados\_sp:** Importa productos desde un archivo Excel (Productos\_importados). Aplica transformaciones sobre la categoría (por ejemplo, agrupa ‘Carnes’, ‘Frutas/Verduras’ y ‘Pescado/Marisco’ en ‘Frescos’; ‘Condimentos’, ‘Granos/Cereales’, ‘Lácteos’ y ‘Repostería’ en ‘Almacen’). Utiliza una CTE para deduplicar los productos (dejando solo la versión más reciente según el precio) y, finalmente, inserta los datos en la tabla de productos.
* **Empresa.ImportarSucursales\_sp:** Procedimiento que importa la información de sucursales desde un archivo Excel. Carga los datos en una tabla temporal, y luego inserta las sucursales en la tabla definitiva, sin aplicar transformaciones sobre los datos, ya que se asume que están correctos.
* **Empresa.ImportarEmpleados\_sp:** Importa la información de empleados desde un archivo Excel. Incluye validaciones para evitar duplicados (por ejemplo, verificando que el DNI no se repita) y utiliza una CTE para deduplicar registros. Además, asocia a cada empleado con la sucursal correspondiente y, opcionalmente, genera valores aleatorios para algunos campos (como el teléfono) en caso de ser necesario.
* **Ventas.CargarClientesAleatorios\_sp:** Procedimiento que genera un número determinado de clientes aleatorios para el programa de fidelización. Utiliza tablas temporales y funciones aleatorias para seleccionar nombres y apellidos de listas predefinidas, insertándolos en la tabla de clientes.
* **Ventas.ImportarVentas\_sp:** Se encarga de importar las ventas registradas desde un archivo CSV. Crea tablas temporales para cargar los datos, transforma la información (por ejemplo, actualiza la ciudad según reglas predefinidas), agrupa los datos para calcular el total por factura y, finalmente, inserta registros en las tablas de Factura y DetalleVenta. La generación de cliente se realiza mediante CROSS APPLY para asignar de forma aleatoria un cliente adecuado a cada factura.
* **Seguridad.GenerarNotaCredito\_sp:** Este procedimiento permite generar una nota de crédito para una factura determinada, pero solo puede ser ejecutado por usuarios con el rol Supervisor. Verifica que la factura esté activa y, de cumplirse la condición, inserta la nota de crédito en la tabla correspondiente, registrando la fecha de emisión y los detalles de la devolución.
* **Seguridad.EncriptarEmpleado\_sp:** Procedimiento destinado a encriptar datos sensibles de empleados (como CUIL, domicilio, teléfono y email personal). Este proceso se realiza en múltiples pasos: se agregan nuevas columnas encriptadas, se actualizan los registros en la tabla de Empleado utilizando EncryptByPassPhrase, se eliminan las columnas originales y se renombran las columnas encriptadas para conservar la nomenclatura original.

## Juegos de Prueba

En el proyecto se incluye un script de “Juegos de Prueba”. En él se incluyen descripciones de los escenarios, la metodología para la ejecución de pruebas, los resultados esperados y consideraciones sobre el comportamiento del sistema. La ejecución de estos juegos de prueba permite verificar la integridad de los procesos de importación, la validez de las operaciones de ABM, la generación de reportes y la aplicación de medidas de seguridad.

# Justificaciones Técnicas

## Motor SQL Server

La elección de Microsoft SQL Server para este proyecto se fundamenta en diversas razones técnicas y de negocio. En primer lugar, SQL Server es un motor de base de datos altamente maduro y probado en entornos empresariales. Su solidez en el manejo de transacciones, integridad de datos y seguridad lo convierten en una opción confiable para un sistema de gestión de ventas centralizado.

Además, SQL Server permite escalar vertical y horizontalmente, ofreciendo opciones de optimización mediante índices, particionamiento de tablas y ajustes en la configuración del servidor. Esto es fundamental para un sistema que debe manejar el crecimiento de datos, especialmente en escenarios de importaciones masivas y reportes complejos.

Por otro lado, el motor cuenta con avanzadas características de seguridad, incluyendo autenticación mixta, cifrado de datos (TDE, encriptación a nivel de columna), y la posibilidad de definir roles y permisos granulares. Estas características son esenciales para proteger datos sensibles, como la información de empleados, y para cumplir con los requerimientos de seguridad de AuroraSA.

SQL Server se integra de forma nativa con múltiples herramientas de desarrollo y administración (SQL Server Management Studio, SQL Server Data Tools, Reporting Services, etc.). Esto facilita la creación, el mantenimiento y la documentación del proyecto. Además, el uso de SQL dinámico, procedimientos almacenados y funciones permite implementar de forma modular la lógica de negocio.

Por último, al ser uno de los motores de base de datos más usados en el mundo empresarial, SQL Server cuenta con un amplio soporte, documentación oficial y una comunidad activa, lo que facilita la resolución de problemas y la incorporación de mejoras.

## 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 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.

# Características de código

## Nomenclatura

El código se ha organizado de manera modular en procedimientos almacenados, funciones y scripts, siguiendo convenciones de nomenclatura claras: las tablas se nombran en singular; los procedimientos con el sufijo sp y los índices con el prefijo ix; las restricciones primarias y foráneas llevan el sufijo PK y FK respectivamente; las tablas esquemas y procedimientos utilizan CamelCase; los campos y las variables comienzan en minúscula; cada archivo incluye comentarios detallados que explican la funcionalidad de cada bloque y las validaciones implementadas; se han utilizado transacciones, bloques TRY/CATCH y técnicas de optimización para garantizar la integridad y el rendimiento del sistema

## Consideraciones a la hora de importar

La ejecución de los procedimientos destinados importaciones masivas de archivos no es arbitraría; 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 precondición que las siguientes tablas se encuentren cargadas: Cliente, Empleado, Sucursal, Producto.

Debido al manejo transaccional en la importación una ejecución en distinto orden no provocará errores fatales, simplemente se informará errores de referencia por claves foráneas u otros errores y se ejecutará un ROLLBACK de la transacción revirtiendo los cambios.