

Universidad Americana
Facultad de Ingeniería y Arquitectura



Fundamentos de Bases de Datos - Grupo 1

Actividad Evaluativa #03

Estudiantes

Priscila Julieth Selva Flores - CIF: 23021745

Emma Cecilia Serrano Urroz - CIF: 23012394

Jeyni Lomary Orozco Chávez - CIF: 24010057

Docente

MSc. Yader Rodríguez López

Fecha 23/09/2025

Índice

1. Introducción.....	3
Descripción general del proyecto.....	3
Objetivos de la implementación de la base de datos.....	3
• Objetivo general.....	3
• Objetivos específicos.....	3
2. Definición del problema.....	4
3. Primera versión de la base de datos.....	5
4. Anexos.....	13
5. Documentación.....	116

1. Introducción

Descripción general del proyecto

El proyecto tiene como objetivo principal desarrollar un sistema de gestión de operaciones para la Casa Automotriz NicaMotores. Dicho sistema permitirá a los empleados, gerentes y administradores de la empresa gestionar de forma centralizada y eficiente los procesos clave del negocio, como la venta de vehículos, la administración de clientes y proveedores, la gestión de repuestos, la programación de citas de servicio, la ejecución de mantenimientos, la facturación y las órdenes de compra. La base de datos servirá como núcleo del sistema, garantizando la integridad de la información, la reducción de errores manuales, el control del inventario de repuestos, así como un seguimiento detallado de las ventas, citas programadas y servicios prestados a clientes internos y externos.

Objetivos de la implementación de la base de datos

- **Objetivo general**

Implementar una base de datos en NicaMotores para mejorar la eficiencia en la gestión de ventas, servicios, citas y abastecimiento, permitiendo un control integral de clientes, vehículos, proveedores, empleados, repuestos, facturación y órdenes de compra.

- **Objetivos específicos**

- Diseñar una estructura de datos que permita registrar de forma única y segura la información de clientes, empleados y proveedores.
- Implementar un control de inventario de repuestos que garantice la disponibilidad y trazabilidad de las piezas utilizadas en ventas y servicios.
- Optimizar la gestión de ventas de vehículos y la generación de facturas asociadas a cada transacción.

- Registrar, programar y dar seguimiento a las citas de servicio de clientes, asegurando que cada servicio realizado esté vinculado a una cita previa.
- Dar trazabilidad a los servicios realizados, tanto para vehículos vendidos por la empresa como para vehículos externos.
- Integrar la información de órdenes de compra y detalle de órdenes para mantener un control claro de las adquisiciones a proveedores.
- Asegurar la consistencia y calidad de los datos mediante la aplicación de restricciones (PK, FK, UNIQUE, CHECK).
- Proporcionar a la gerencia reportes confiables para la toma de decisiones estratégicas.

2. Definición del problema

Actualmente, NicaMotores enfrenta dificultades en la gestión de sus operaciones debido a la falta de un sistema centralizado que organice la información de clientes, ventas, vehículos, repuestos, citas y servicios. Muchos procesos se llevan a cabo de manera manual o en registros dispersos, lo que genera:

- Duplicidad y pérdida de información de clientes y empleados, dificultando el seguimiento de relaciones comerciales.
- Falta de control en el inventario de repuestos, ocasionando retrasos en las órdenes de compra y en la prestación de servicios.
- Procesos de facturación y ventas desorganizados, con riesgo de inconsistencias entre las transacciones realizadas y los documentos emitidos.
- Escasa trazabilidad en los servicios realizados, especialmente en los casos de vehículos externos, donde no existe un historial centralizado.

- Ausencia de un sistema para programar y controlar citas de servicio, lo que provoca sobrecarga de trabajo, citas traslapadas y dificultad en la asignación de recursos del taller.
- Ausencia de un control estructurado de proveedores y órdenes de compra, lo que impacta en la eficiencia del abastecimiento.

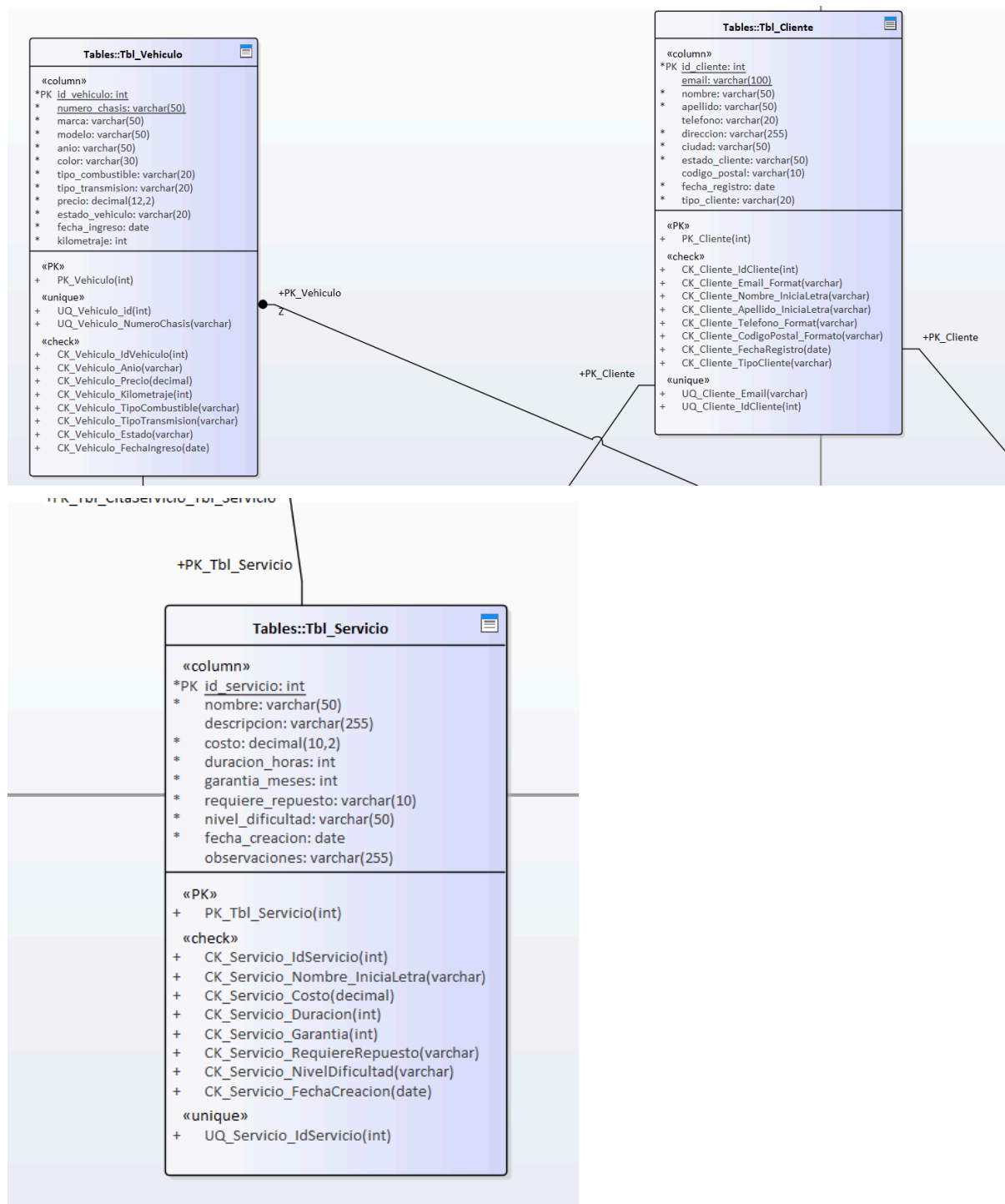
La carencia de un sistema unificado impide a la empresa tener una visión global y confiable de sus operaciones. Esto afecta tanto a la atención al cliente como a la gestión administrativa y estratégica.

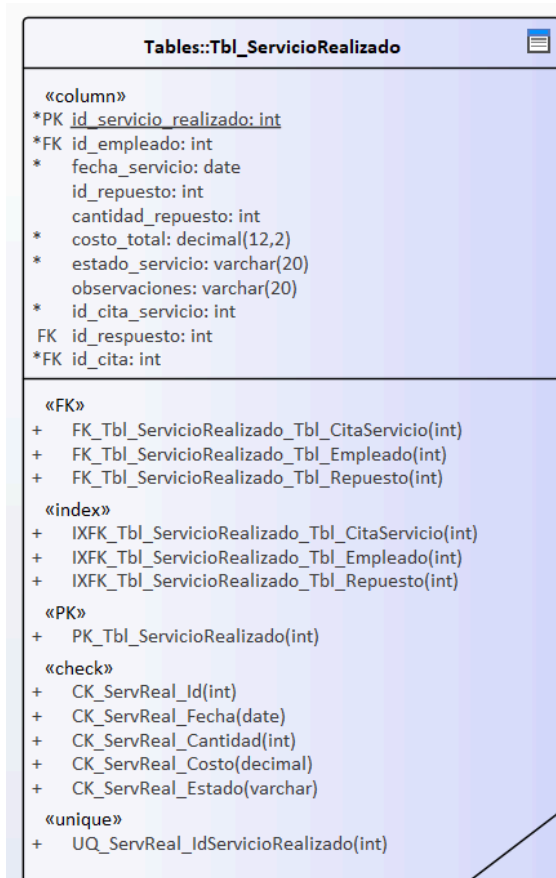
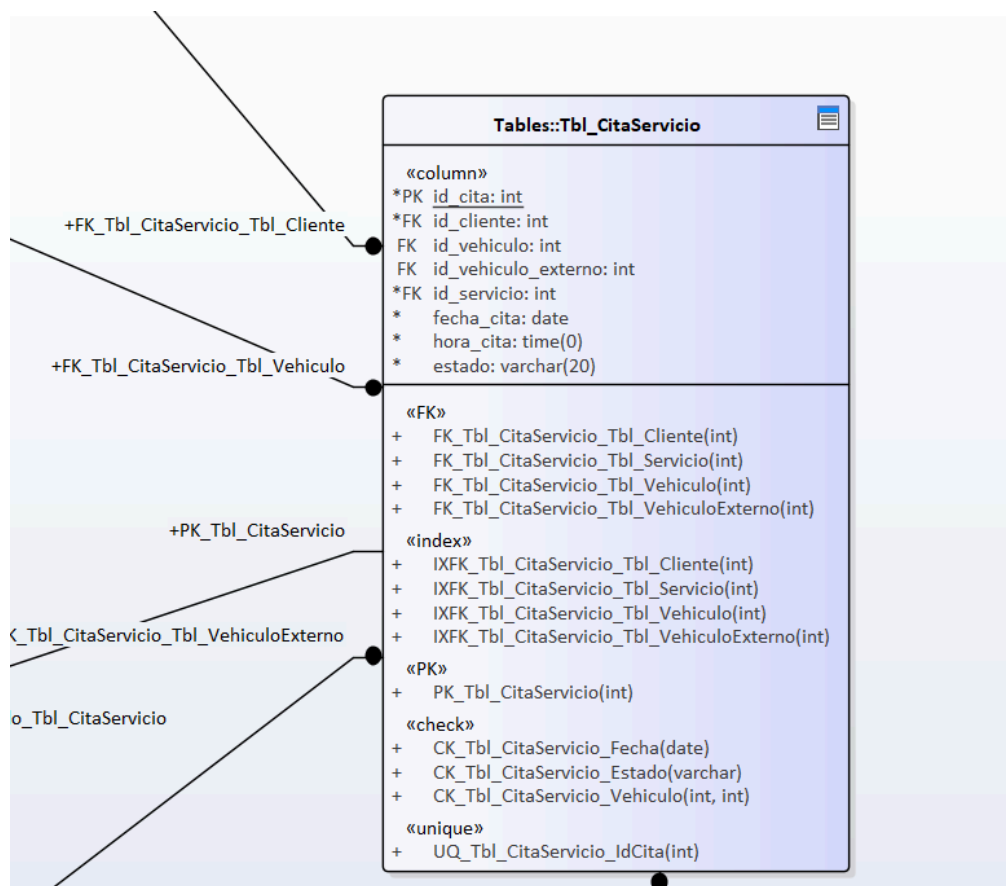
La solución a este problema es el diseño e implementación de una base de datos relacional que integre la información en 13 tablas principales:

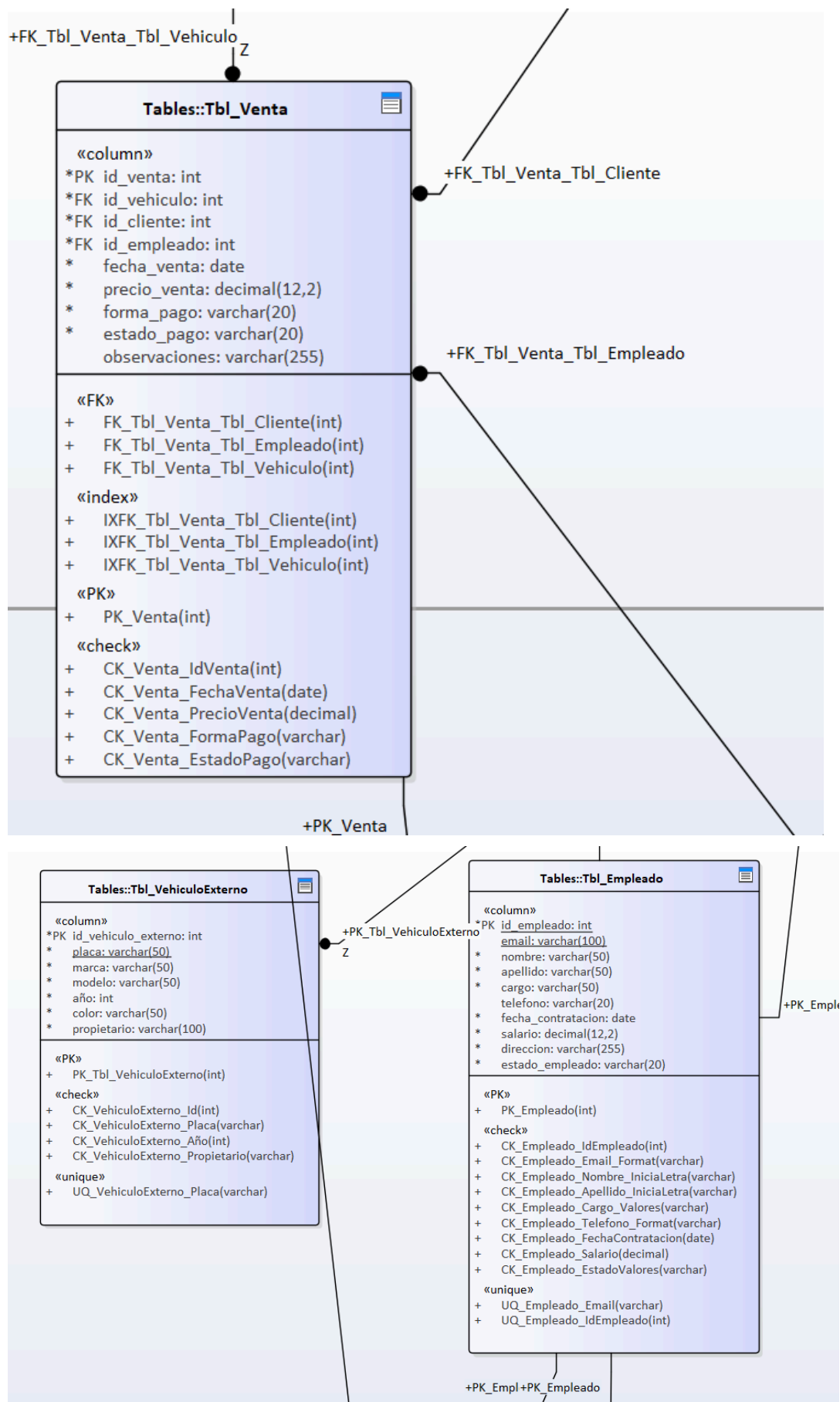
- Clientes, Empleados, Vehículos y Vehículos Externos para la gestión de actores y bienes.
- Proveedores, Repuestos, Órdenes de Compra y Detalles de Órdenes de Compra para el control del abastecimiento.
- Ventas y Facturas para la gestión de transacciones comerciales.
- Citas de Servicio, Servicios y Servicios Realizados para el control integral de la agenda, mantenimientos y reparaciones.

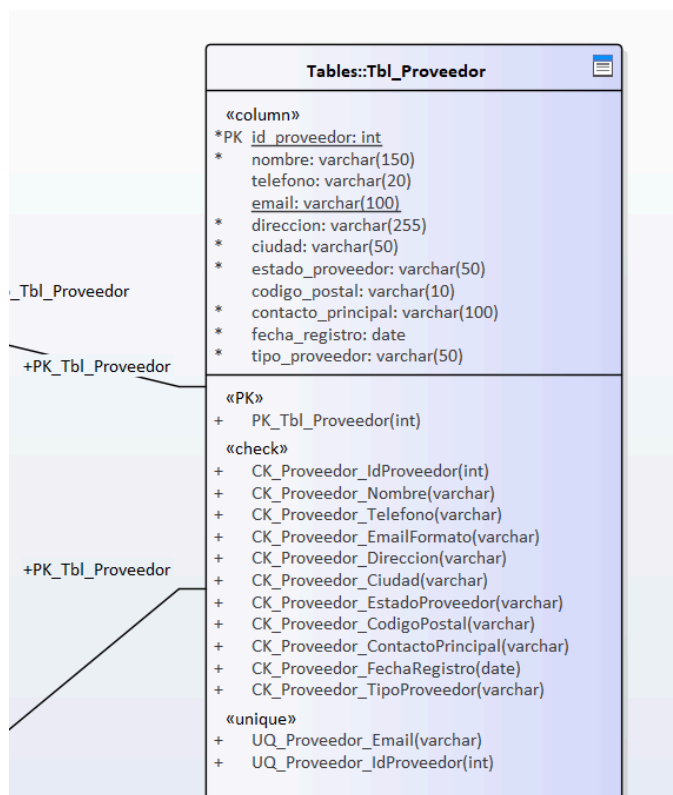
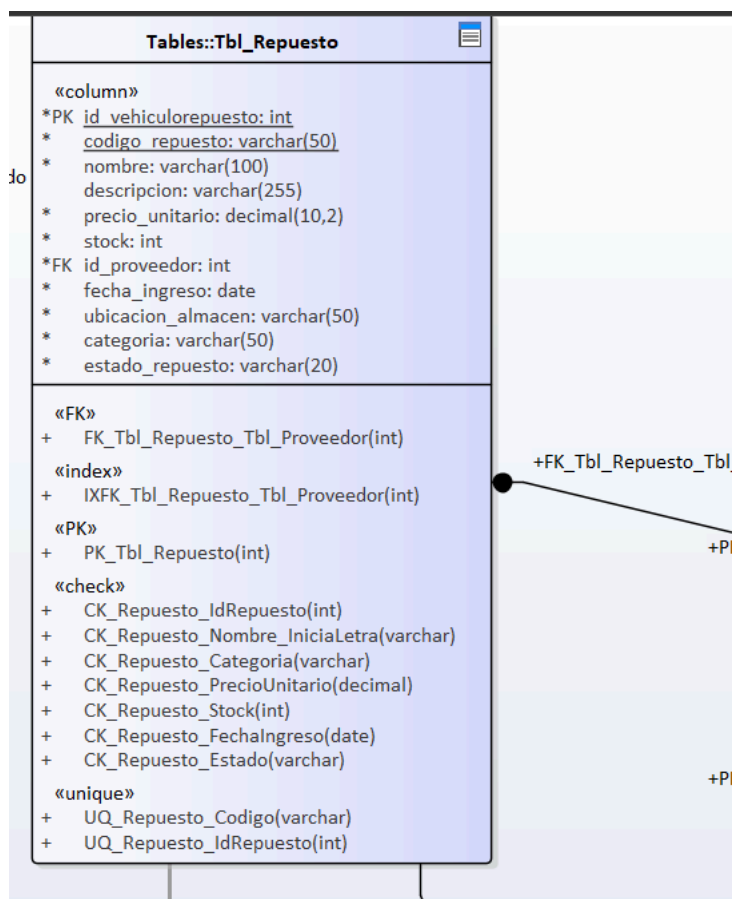
Con esta estructura, NicaMotores podrá garantizar la consistencia de los datos, mejorar la eficiencia operativa y ofrecer un servicio más ágil y confiable a sus clientes.

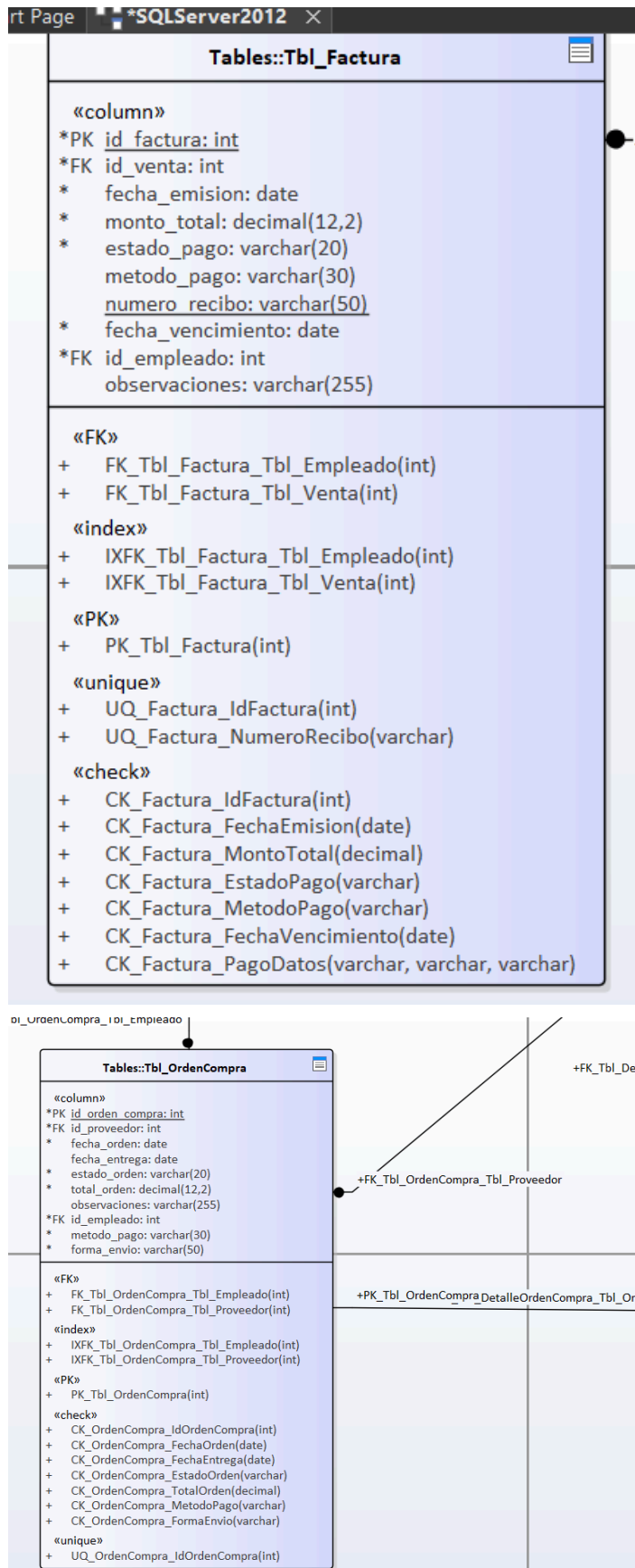
3. Primera versión de la base de datos

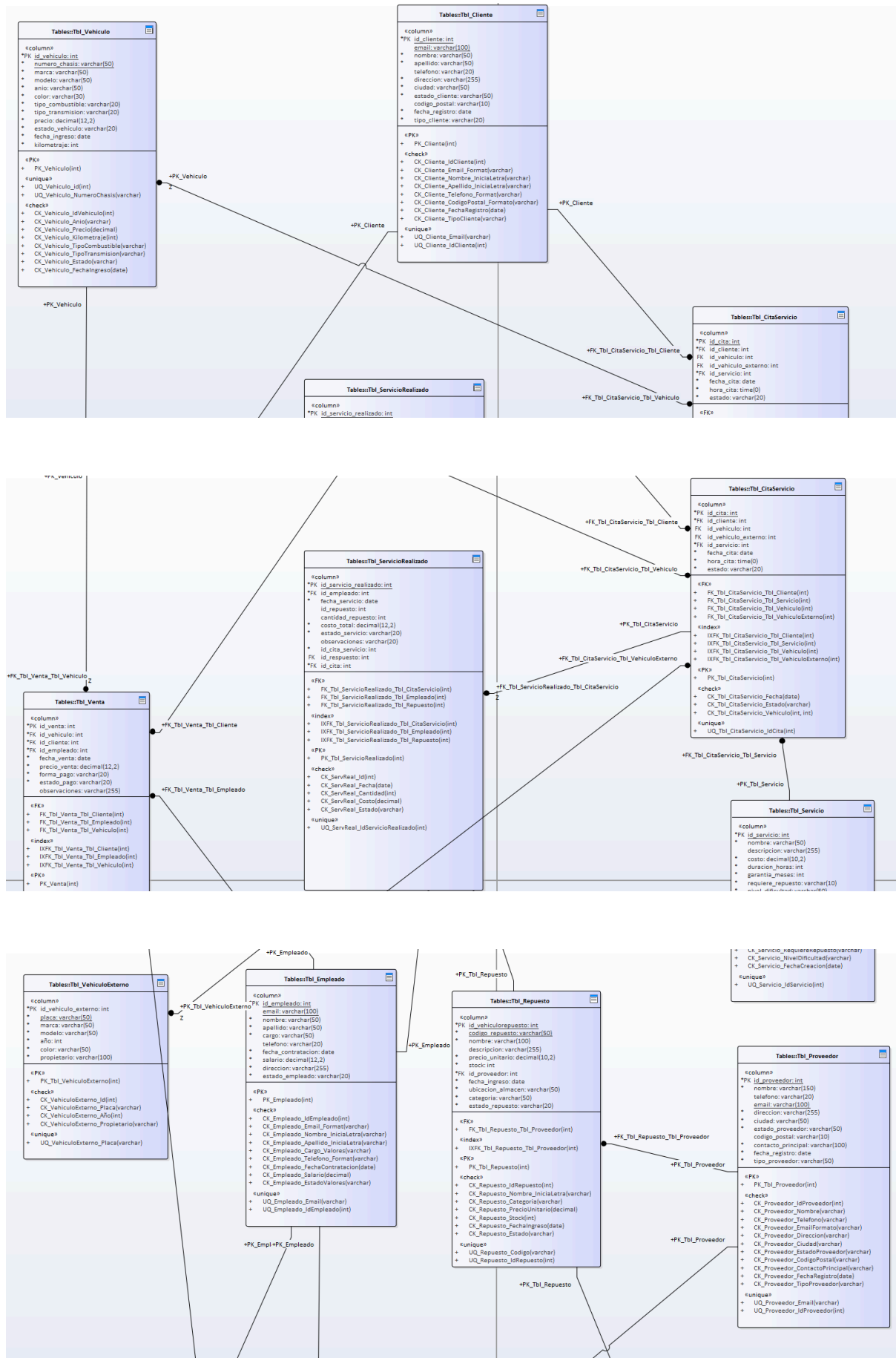


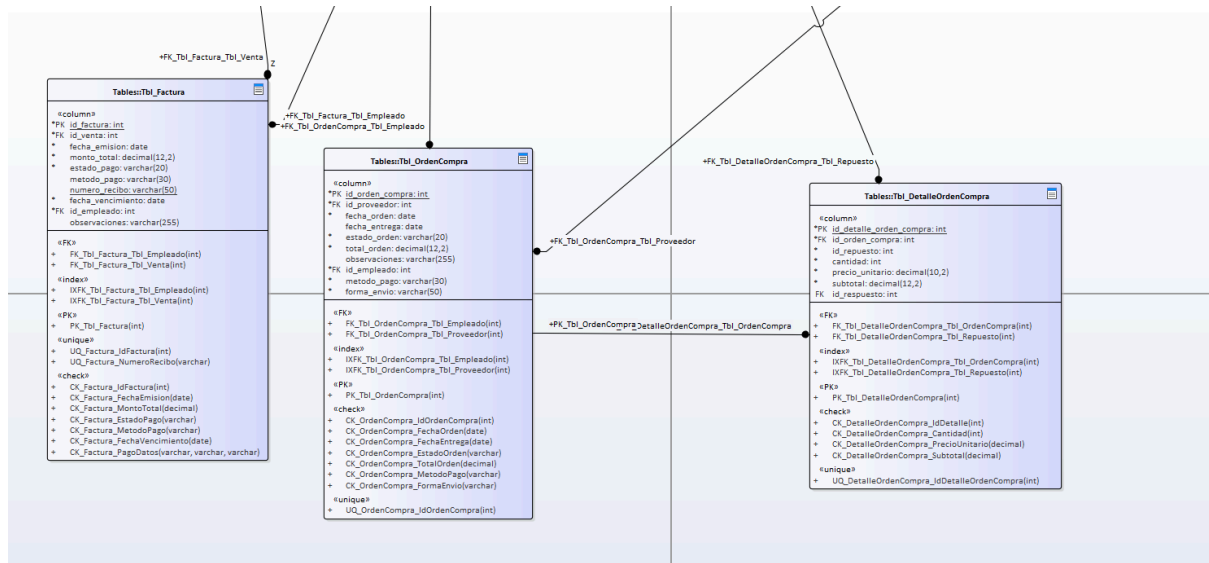












4. Anexos

```

/* ----- */
/* Generated by Enterprise Architect Version 13.5          */
/* Created On : 22-sep.-2025 11:01:38 a. m.              */
/* DBMS      : SQL Server 2012                          */
/* ----- */

/* Drop Foreign Key Constraints */

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id =
object_id(N'[FK_Tbl_CitaServicio_Tbl_Cliente]') AND OBJECTPROPERTY(id,
N'IsForeignKey') = 1)
ALTER TABLE [Tbl_CitaServicio] DROP CONSTRAINT [FK_Tbl_CitaServicio_Tbl_Cliente]
GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id =
object_id(N'[FK_Tbl_CitaServicio_Tbl_Servicio]') AND OBJECTPROPERTY(id,
N'IsForeignKey') = 1)
ALTER TABLE [Tbl_CitaServicio] DROP CONSTRAINT [FK_Tbl_CitaServicio_Tbl_Servicio]
GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id =
object_id(N'[FK_Tbl_CitaServicio_Tbl_Vehiculo]') AND OBJECTPROPERTY(id,
N'IsForeignKey') = 1)
ALTER TABLE [Tbl_CitaServicio] DROP CONSTRAINT [FK_Tbl_CitaServicio_Tbl_Vehiculo]
GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id =
object_id(N'[FK_Tbl_CitaServicio_Tbl_VehiculoExterno]') AND OBJECTPROPERTY(id,
N'IsForeignKey') = 1)
ALTER TABLE [Tbl_CitaServicio] DROP CONSTRAINT
[FK_Tbl_CitaServicio_Tbl_VehiculoExterno]
GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id =
object_id(N'[FK_Tbl_DetalleOrdenCompra_Tbl_OrdenCompra]') AND
OBJECTPROPERTY(id, N'IsForeignKey') = 1)
ALTER TABLE [Tbl_DetalleOrdenCompra] DROP CONSTRAINT
[FK_Tbl_DetalleOrdenCompra_Tbl_OrdenCompra]
GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id =
object_id(N'[FK_Tbl_DetalleOrdenCompra_Tbl_Repuesto]') AND OBJECTPROPERTY(id,
N'IsForeignKey') = 1)
ALTER TABLE [Tbl_DetalleOrdenCompra] DROP CONSTRAINT
[FK_Tbl_DetalleOrdenCompra_Tbl_Repuesto]

```

GO

```
IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id =
object_id(N'[FK_Tbl_Factura_Tbl_Empleado]') AND OBJECTPROPERTY(id,
N'IsForeignKey') = 1)
ALTER TABLE [Tbl_Factura] DROP CONSTRAINT [FK_Tbl_Factura_Tbl_Empleado]
GO
```

```
IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id =
object_id(N'[FK_Tbl_Factura_Tbl_Venta]') AND OBJECTPROPERTY(id, N'IsForeignKey') =
1)
ALTER TABLE [Tbl_Factura] DROP CONSTRAINT [FK_Tbl_Factura_Tbl_Venta]
GO
```

```
IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id =
object_id(N'[FK_Tbl_OrdenCompra_Tbl_Empleado]') AND OBJECTPROPERTY(id,
N'IsForeignKey') = 1)
ALTER TABLE [Tbl_OrdenCompra] DROP CONSTRAINT
[FK_Tbl_OrdenCompra_Tbl_Empleado]
GO
```

```
IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id =
object_id(N'[FK_Tbl_OrdenCompra_Tbl_Proveedor]') AND OBJECTPROPERTY(id,
N'IsForeignKey') = 1)
ALTER TABLE [Tbl_OrdenCompra] DROP CONSTRAINT
[FK_Tbl_OrdenCompra_Tbl_Proveedor]
GO
```

```
IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id =
object_id(N'[FK_Tbl_Repuesto_Tbl_Proveedor]') AND OBJECTPROPERTY(id,
N'IsForeignKey') = 1)
ALTER TABLE [Tbl_Repuesto] DROP CONSTRAINT [FK_Tbl_Repuesto_Tbl_Proveedor]
GO
```

```
IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id =
object_id(N'[FK_Tbl_ServicioRealizado_Tbl_CitaServicio]') AND OBJECTPROPERTY(id,
N'IsForeignKey') = 1)
ALTER TABLE [Tbl_ServicioRealizado] DROP CONSTRAINT
[FK_Tbl_ServicioRealizado_Tbl_CitaServicio]
GO
```

```
IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id =
object_id(N'[FK_Tbl_ServicioRealizado_Tbl_Empleado]') AND OBJECTPROPERTY(id,
N'IsForeignKey') = 1)
ALTER TABLE [Tbl_ServicioRealizado] DROP CONSTRAINT
[FK_Tbl_ServicioRealizado_Tbl_Empleado]
GO
```

```

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id =
object_id(N'[FK_Tbl_ServicioRealizado_Tbl_Repuesto]') AND OBJECTPROPERTY(id,
N'IsForeignKey') = 1)
ALTER TABLE [Tbl_ServicioRealizado] DROP CONSTRAINT
[FK_Tbl_ServicioRealizado_Tbl_Repuesto]
GO

```

```

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id =
object_id(N'[FK_Tbl_Venta_Tbl_Cliente]') AND OBJECTPROPERTY(id, N'IsForeignKey') =
1)
ALTER TABLE [Tbl_Venta] DROP CONSTRAINT [FK_Tbl_Venta_Tbl_Cliente]
GO

```

```

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id =
object_id(N'[FK_Tbl_Venta_Tbl_Empleado]') AND OBJECTPROPERTY(id, N'IsForeignKey')
= 1)
ALTER TABLE [Tbl_Venta] DROP CONSTRAINT [FK_Tbl_Venta_Tbl_Empleado]
GO

```

```

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id =
object_id(N'[FK_Tbl_Venta_Tbl_Vehiculo]') AND OBJECTPROPERTY(id, N'IsForeignKey') =
1)
ALTER TABLE [Tbl_Venta] DROP CONSTRAINT [FK_Tbl_Venta_Tbl_Vehiculo]
GO

```

```

/* Drop Tables */

```

```

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object_id(N'[Tbl_CitaServicio]')
AND OBJECTPROPERTY(id, N'IsUserTable') = 1)
DROP TABLE [Tbl_CitaServicio]
GO

```

```

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object_id(N'[Tbl_Cliente]') AND
OBJECTPROPERTY(id, N'IsUserTable') = 1)
DROP TABLE [Tbl_Cliente]
GO

```

```

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id =
object_id(N'[Tbl_DetalleOrdenCompra]') AND OBJECTPROPERTY(id, N'IsUserTable') = 1)
DROP TABLE [Tbl_DetalleOrdenCompra]
GO

```

```

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object_id(N'[Tbl_Empleado]')
AND OBJECTPROPERTY(id, N'IsUserTable') = 1)
DROP TABLE [Tbl_Empleado]
GO

```

```
IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object_id(N'[Tbl_Factura]') AND
OBJECTPROPERTY(id, N'IsUserTable') = 1)
DROP TABLE [Tbl_Factura]
GO
```

```
IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object_id(N'[Tbl_OrdenCompra]')
AND OBJECTPROPERTY(id, N'IsUserTable') = 1)
DROP TABLE [Tbl_OrdenCompra]
GO
```

```
IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object_id(N'[Tbl_Proveedor]')
AND OBJECTPROPERTY(id, N'IsUserTable') = 1)
DROP TABLE [Tbl_Proveedor]
GO
```

```
IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object_id(N'[Tbl_Repuesto]')
AND OBJECTPROPERTY(id, N'IsUserTable') = 1)
DROP TABLE [Tbl_Repuesto]
GO
```

```
IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object_id(N'[Tbl_Servicio]') AND
OBJECTPROPERTY(id, N'IsUserTable') = 1)
DROP TABLE [Tbl_Servicio]
GO
```

```
IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id =
object_id(N'[Tbl_ServicioRealizado]') AND OBJECTPROPERTY(id, N'IsUserTable') = 1)
DROP TABLE [Tbl_ServicioRealizado]
GO
```

```
IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object_id(N'[Tbl_Vehiculo]') AND
OBJECTPROPERTY(id, N'IsUserTable') = 1)
DROP TABLE [Tbl_Vehiculo]
GO
```

```
IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id =
object_id(N'[Tbl_VehiculoExterno]') AND OBJECTPROPERTY(id, N'IsUserTable') = 1)
DROP TABLE [Tbl_VehiculoExterno]
GO
```

```
IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object_id(N'[Tbl_Venta]') AND
OBJECTPROPERTY(id, N'IsUserTable') = 1)
DROP TABLE [Tbl_Venta]
GO
```

```
/* Create Tables */
```

```
CREATE TABLE [Tbl_CitaServicio]
```



```
(
    [id_cita] int NOT NULL, -- Aquí se almacena el identificador numérico único de
    cada cita registrada en la base de datos. No puede repetirse, debe ser un número entero
    positivo y no puede quedar vacío. Este campo sirve como clave primaria para identificar de
    manera inequívoca cada cita de servicio agendada por un cliente.
    [id_cliente] int NOT NULL, -- Este campo almacena el identificador del cliente que
    solicita la cita de servicio. Es un número entero que establece una relación con la tabla
    Tbl_Cliente, garantizando que toda cita esté asociada a un cliente registrado en el sistema.
    No puede quedar vacío, ya que toda cita debe corresponder a un cliente específico.
    [id_vehiculo] int NULL, -- Aquí se almacena el identificador del vehículo que
    pertenece a NicaMotores y que será atendido en la cita. Es una clave foránea hacia la tabla
    Tbl_Vehiculo. Este campo es opcional porque la cita puede ser para un vehículo externo.
    [id_vehiculo_externo] int NULL, -- Este atributo registra el identificador del vehículo
    externo (no comprado en NicaMotores) que será atendido en la cita. Se relaciona con la
    tabla Tbl_VehiculoExterno. Es opcional, ya que la cita también puede ser para un vehículo
    interno.
    [id_servicio] int NOT NULL, -- Este campo guarda el identificador del servicio
    solicitado en la cita. Se relaciona con la tabla Tbl_Servicio y permite registrar si se trata de
    un mantenimiento, reparación u otro tipo de servicio. No puede quedar vacío porque toda
    cita debe estar vinculada a un servicio específico.
    [fecha_cita] date NOT NULL, -- Aquí se almacena la fecha agendada para la cita.
    Debe ser una fecha igual o posterior a la fecha actual (no puede estar en el pasado). No
    puede quedar vacía ya que toda cita debe tener definida una fecha de realización.
    [hora_cita] time NOT NULL, -- Este atributo guarda la hora específica de la cita, lo
    que permite organizar el calendario de servicios en el taller. No puede estar vacío porque es
    necesario definir un horario para atender cada cita. En futuras versiones se deberá validar
    que no se programen más de dos citas en el mismo horario y que no existan traslapes de
    tiempo de acuerdo con la duración del servicio. Esto podrá implementarse mediante triggers
    o reglas de negocio en la aplicación.
    [estado] varchar(20) NOT NULL -- Este campo almacena el estado actual de la
    cita, el cual puede tomar valores como Pendiente, Confirmada, Cancelada o Atendida.
    Permite dar seguimiento al ciclo de vida de la cita, desde que es agendada hasta que se
    atiende o se cancela. No puede estar vacío.
)
GO
```

```
CREATE TABLE [Tbl_Cliente]
```

```
(
    [id_cliente] int NOT NULL, -- Aquí se almacena el identificador numérico único de
    cada cliente registrado en la base de datos. No puede repetirse, debe ser un número entero
    positivo y no puede quedar vacío. Este campo sirve como clave primaria para identificar
    cada registro de cliente.
    [email] varchar(100) NULL, -- Aquí se guarda el correo electrónico principal del
    cliente. Debe ser único, no puede repetirse entre clientes y debe tener un formato válido de
    correo electrónico (usuario@dominio). No puede iniciar ni terminar con espacios en blanco.
    [nombre] varchar(50) NOT NULL, -- Aquí se almacena el nombre completo del
    cliente. Debe contener caracteres alfabéticos, no puede quedar vacío ni comenzar con un
    carácter especial. Puede incluir espacios o guiones después de la primera letra.
```

[apellido] varchar(50) NOT NULL, -- Aquí se almacena el/los apellidos del cliente. Debe contener caracteres alfabéticos, no puede quedar vacío ni comenzar con un carácter especial. Puede incluir espacios o guiones después de la primera letra.

[telefono] varchar(20) NULL, -- Aquí se almacena el número telefónico de contacto del cliente. Puede ser nulo. Si se ingresa, solo puede contener dígitos, espacios, signos +, - o paréntesis, y debe iniciar con dígito o +.

[direccion] varchar(255) NOT NULL, -- Aquí se almacena la dirección física del cliente (calle, número, referencia). Es obligatoria, no puede quedar vacía y tiene un máximo de 255 caracteres.

[ciudad] varchar(50) NOT NULL, -- Aquí se almacena la ciudad de residencia del cliente. Es obligatoria y no puede quedar vacía ni iniciar con caracteres especiales.

[estado_cliente] varchar(50) NOT NULL, -- Aquí se almacena el estado, provincia o departamento donde reside el cliente. Es obligatorio y no puede quedar vacío.

[codigo_postal] varchar(10) NULL, -- Aquí se almacena el código postal del cliente. Es opcional; si se ingresa, debe contener solo caracteres alfanuméricos, espacios o guiones y tener máximo 10 caracteres.

[fecha_registro] date NOT NULL, -- Aquí se almacena la fecha en que se registró el cliente en el sistema. Es obligatoria, no puede quedar vacía y no puede ser mayor a la fecha actual.

[tipo_cliente] varchar(20) NOT NULL -- Aquí se almacena el tipo de cliente según su naturaleza. Solo se permiten valores predefinidos como particular o empresa. Es obligatorio y no puede quedar vacío.

)

GO

CREATE TABLE [Tbl_DetalleOrdenCompra]

(

[id_detalle_orden_compra] int NOT NULL, -- Representa el identificador único de cada línea de detalle de una orden de compra. Es un número entero positivo, obligatorio y único. Permite diferenciar cada producto o repuesto dentro de una orden de compra.

[id_orden_compra] int NOT NULL, -- Aquí se guarda el identificador de la orden de compra a la que pertenece este detalle. Es un dato obligatorio que no puede quedar vacío y funciona como clave foránea para asegurar que cada línea de detalle esté vinculada a una orden de compra existente en la base de datos.

[id_repuesto] int NOT NULL, -- Aquí se almacena el identificador del repuesto solicitado en la orden de compra. Es obligatorio, no puede quedar vacío y funciona como clave foránea para garantizar que solo se registren repuestos existentes en la base de datos.

[cantidad] int NOT NULL, -- Aquí se guarda la cantidad del repuesto solicitado en esta línea de detalle. Es un campo obligatorio, no puede quedar vacío y debe ser un número entero mayor que cero, ya que representa la cantidad a comprar.

[precio_unitario] decimal(10,2) NOT NULL, -- Aquí se almacena el precio unitario acordado para cada unidad del repuesto en esta línea de detalle. Es obligatorio, no puede quedar vacío y debe ser un número decimal mayor o igual a cero.

[subtotal] decimal(12,2) NOT NULL, -- Aquí se almacena el importe total de la línea de detalle calculado como la multiplicación entre la cantidad solicitada y el precio unitario de ese repuesto; no puede quedar vacío y debe ser mayor o igual a cero.

[id_repuesto] int NULL

)
GO

CREATE TABLE [Tbl_Empleado]

(

[id_empleado] int NOT NULL, -- Aquí se almacena el identificador numérico único de cada empleado registrado en la base de datos. No puede repetirse, debe ser un número entero positivo y no puede quedar vacío. Este campo sirve como clave primaria para identificar cada registro de empleado.

[email] varchar(100) NULL, -- Aquí se guarda el correo electrónico del empleado. Debe ser único, no puede repetirse entre empleados y debe tener un formato válido de correo electrónico (usuario@dominio). No puede iniciar ni terminar con espacios en blanco.

[nombre] varchar(50) NOT NULL, -- Aquí se almacena el nombre del empleado. Debe contener caracteres alfabéticos, no puede quedar vacío ni comenzar con un carácter especial. Puede incluir espacios o guiones después de la primera letra.

[apellido] varchar(50) NOT NULL, -- Aquí se almacena el/los apellidos del empleado. Debe contener caracteres alfabéticos, no puede quedar vacío ni comenzar con un carácter especial. Puede incluir espacios o guiones después de la primera letra.

[cargo] varchar(50) NOT NULL, -- Aquí se almacena el cargo o puesto que ocupa el empleado dentro de la empresa. Solo se permiten valores predefinidos como vendedor, mecánico, etc. Es obligatorio y no puede quedar vacío.

[telefono] varchar(20) NULL, -- Aquí se almacena el número telefónico de contacto del empleado. Puede ser nulo. Si se ingresa, solo puede contener dígitos, espacios, signos +, - o paréntesis, y debe iniciar con dígito 0 +.

[fecha_contratacion] date NOT NULL, -- Aquí se almacena la fecha en que fue contratado el empleado. Es obligatoria, no puede quedar vacía y no puede ser mayor a la fecha actual.

[salario] decimal(12,2) NOT NULL, -- Aquí se almacena el salario mensual del empleado. Debe ser un número decimal mayor o igual a cero y no puede quedar vacío.

[direccion] varchar(255) NOT NULL, -- Aquí se almacena la dirección física del empleado. Es obligatoria, no puede quedar vacía y tiene un máximo de 255 caracteres.

[estado_empleado] varchar(20) NOT NULL -- Aquí se almacena el estado laboral actual del empleado. Solo se permiten valores predefinidos como activo o inactivo. Es obligatorio y no puede quedar vacío.

)
GO

CREATE TABLE [Tbl_Factura]

(

[id_factura] int NOT NULL, -- Identificador único de cada factura emitida por la empresa. Es un número entero positivo, obligatorio y exclusivo. Garantiza la integridad en el control de facturación y permite asociar la factura con su respectiva venta y empleado.

[id_venta] int NOT NULL, -- Aquí se guarda el identificador de la venta asociada a la factura. Es obligatorio, no puede quedar vacío y funciona como clave foránea para asegurar que toda factura esté vinculada a una venta existente en la base de datos.

[fecha_emision] date NOT NULL, -- Aquí se almacena la fecha en la que se emitió la factura. Es obligatoria, no puede quedar vacía y no puede ser posterior a la fecha actual para garantizar que refleje un evento real.

[monto_total] decimal(12,2) NOT NULL, -- Aquí se guarda el monto total facturado. Es obligatorio, no puede quedar vacío y debe ser mayor o igual a cero, ya que representa la suma de los conceptos incluidos en la factura.

[estado_pago] varchar(20) NOT NULL, -- Aquí se almacena el estado actual del pago de la factura. Es obligatorio y solo puede tomar uno de los valores predefinidos ("pagado", "pendiente"), permitiendo controlar el estado financiero.

[metodo_pago] varchar(30) NULL, -- Aquí se guarda el método de pago utilizado para la factura. Es obligatorio únicamente cuando el estado de pago es "pagado"; si la factura está pendiente este campo puede quedar vacío. Solo puede tomar valores predefinidos como "efectivo" o "tarjeta".

[numero_recibo] varchar(50) NULL, -- Aquí se almacena el número único de recibo asociado a la factura. Es obligatorio únicamente cuando el estado de pago es "pagado"; si la factura está pendiente este campo puede quedar vacío. No puede repetirse para facturas pagadas.

[fecha_vencimiento] date NOT NULL, -- Aquí se almacena la fecha límite para el pago de la factura. Es obligatoria, no puede quedar vacía y debe ser igual o posterior a la fecha de emisión para asegurar coherencia en los plazos.

[id_empleado] int NOT NULL, -- Aquí se guarda el identificador del empleado que generó la factura. Es obligatorio, no puede quedar vacío y funciona como clave foránea para asegurar que cada factura esté asociada a un empleado registrado en el sistema.

[observaciones] varchar(255) NULL -- Aquí se pueden almacenar comentarios adicionales relacionados con la factura. Es un campo opcional y puede quedar vacío.

)

GO

CREATE TABLE [Tbl_OrdenCompra]

(

[id_orden_compra] int NOT NULL, -- Identificador único de cada orden de compra realizada a proveedores. No se puede repetir, debe ser entero positivo y no nulo. Permite rastrear las órdenes, asociarlas con proveedores y empleados responsables.

[id_proveedor] int NOT NULL, -- Aquí se guarda el identificador del proveedor al que se realiza la orden de compra. Es un dato obligatorio que no puede quedar vacío y funciona como clave foránea para garantizar que toda orden esté vinculada a un proveedor existente en la base de datos.

[fecha_orden] date NOT NULL, -- Aquí se almacena la fecha en la que se generó la orden de compra. Es un campo obligatorio que no puede quedar vacío ni ser posterior a la fecha actual, ya que debe reflejar el momento real en que se hizo el pedido.

[fecha_entrega] date NULL, -- Aquí se guarda la fecha estimada o real de entrega de la orden de compra. Es un campo opcional que puede quedar vacío mientras no se tenga confirmación, pero si se ingresa debe ser igual o posterior a la fecha de orden para mantener coherencia temporal.

[estado_orden] varchar(20) NOT NULL, -- Aquí se almacena el estado actual de la orden de compra. Es un campo obligatorio que solo puede tomar los valores permitidos ("pendiente" o "recibida"), para garantizar un control uniforme del avance de las órdenes.

[total_orden] decimal(12,2) NOT NULL, -- Aquí se guarda el monto total de la orden de compra. Es obligatorio, no puede quedar vacío y debe ser mayor o igual a cero, ya que representa el total comprometido para esa orden.

[observaciones] varchar(255) NULL, -- Aquí se pueden registrar comentarios adicionales relacionados con la orden de compra. Es un campo opcional y puede quedar vacío.

[id_empleado] int NOT NULL, -- Aquí se guarda el identificador del empleado que gestionó la orden de compra. Es un campo obligatorio que no puede quedar vacío y funciona como clave foránea para asegurar que cada orden esté asociada a un empleado registrado en el sistema.

[metodo_pago] varchar(30) NOT NULL, -- Aquí se almacena el método de pago acordado para la orden de compra. Es un dato obligatorio y solo puede tomar los valores permitidos ("efectivo" o "transferencia"), para clasificar y auditar correctamente los pagos.

[forma_envio] varchar(50) NOT NULL -- Aquí se guarda la forma de envío de la orden de compra. Es un dato obligatorio que solo puede tomar los valores permitidos ("terrestre", "aéreo" o "marítimo"), para reflejar el medio logístico utilizado.

)
GO

CREATE TABLE [Tbl_Proveedor]

(

[id_proveedor] int NOT NULL, -- Aquí se almacena el identificador numérico único de cada proveedor registrado en la base de datos. No puede repetirse, debe ser un número entero positivo mayor que cero y no puede quedar vacío. Este campo sirve como clave primaria para identificar de forma inequívoca cada proveedor.

[nombre] varchar(150) NOT NULL, -- Aquí se guarda el nombre oficial o razón social del proveedor. Es un dato obligatorio que no puede quedar vacío ni contener únicamente espacios, ya que se utiliza en reportes, búsquedas y facturación para reconocer de forma clara a cada proveedor.

[telefono] varchar(20) NULL, -- Aquí se almacena el número telefónico de contacto del proveedor. Este campo es opcional, pero si se ingresa debe respetar el formato habitual de números telefónicos, permitiendo dígitos, guiones, paréntesis y el signo más. Sirve como un medio directo de comunicación con el proveedor.

[email] varchar(100) NULL, -- Aquí se guarda la dirección de correo electrónico principal del proveedor. Es un campo opcional, pero cuando se proporciona debe tener un formato válido de correo electrónico y no puede repetirse, ya que se utilizará como un medio de contacto formal y para el envío de notificaciones electrónicas.

[direccion] varchar(255) NOT NULL, -- Aquí se almacena la dirección física completa del proveedor, como calle, número de casa o local, barrio o colonia. Es un dato obligatorio que no puede quedar vacío, ya que resulta indispensable para la localización geográfica, envíos de mercancía y registros oficiales.

[ciudad] varchar(50) NOT NULL, -- Aquí se guarda el nombre de la ciudad donde se ubica el proveedor. Este campo es obligatorio, no puede quedar vacío y permite clasificar a los proveedores según su ubicación geográfica para fines de reportes o segmentación.

[estado_proveedor] varchar(50) NOT NULL, -- Aquí se almacena el nombre del estado, provincia o departamento al que pertenece la ciudad donde se ubica el proveedor. Este campo es obligatorio y debe registrarse siempre, ya que facilita la organización geográfica y la generación de informes por región.

[codigo_postal] varchar(10) NULL, -- Aquí se almacena el código postal correspondiente a la dirección del proveedor. Este campo es opcional, pero cuando se proporciona debe cumplir con la longitud establecida y contener únicamente caracteres

válidos como letras, números y guiones, ya que se utiliza para fines de envío y normalización de direcciones.

[contacto_principal] varchar(100) NOT NULL, -- Aquí se almacena el nombre completo de la persona de contacto principal dentro del proveedor. Es un dato obligatorio que no puede quedar vacío, pues resulta esencial para las gestiones comerciales, consultas y negociaciones directas.

[fecha_registro] date NOT NULL, -- Aquí se almacena la fecha en la que el proveedor fue ingresado en la base de datos. Es un campo obligatorio que no puede quedar vacío ni registrar una fecha posterior a la actual, ya que se utiliza con fines de auditoría y control histórico.

[tipo_proveedor] varchar(50) NOT NULL -- Aquí se almacena la clasificación del proveedor según su origen, diferenciando si es de tipo local o internacional. Este dato es obligatorio y únicamente puede tomar uno de estos dos valores, lo que permite aplicar políticas de compra, impuestos y logística según corresponda.

)
GO

CREATE TABLE [Tbl_Repuesto]

(

[id_vehiculorepuesto] int NOT NULL, -- Aquí se almacena el identificador numérico único de cada repuesto registrado en la base de datos. No puede repetirse, debe ser un número entero positivo y no puede quedar vacío. Este campo sirve como clave primaria para identificar cada repuesto.

[codigo_repuesto] varchar(50) NOT NULL, -- Aquí se almacena el código único asignado a cada repuesto. No puede repetirse entre repuestos, debe ser obligatorio y sirve como referencia de inventario.

[nombre] varchar(100) NOT NULL, -- Aquí se almacena el nombre del repuesto. Debe contener solo caracteres alfabéticos y espacios, no puede quedar vacío ni comenzar con un carácter especial.

[descripcion] varchar(255) NULL, -- Aquí se almacena una breve descripción del repuesto. Es opcional, puede quedar vacío, pero si se ingresa debe tener un máximo de 255 caracteres.

[precio_unitario] decimal(10,2) NOT NULL, -- Aquí se almacena el precio de venta por unidad del repuesto. Debe ser un número decimal mayor o igual a cero y no puede quedar vacío.

[stock] int NOT NULL, -- Aquí se almacena la cantidad disponible del repuesto en inventario. Debe ser un número entero mayor o igual a cero y no puede quedar vacío.

[id_proveedor] int NOT NULL, -- Aquí se almacena el identificador del proveedor asociado al repuesto. Es un dato obligatorio y corresponde a una clave foránea que enlaza con la tabla Proveedor. Debe existir previamente en la tabla Proveedor.

[fecha_ingreso] date NOT NULL, -- Aquí se almacena la fecha en que el repuesto ingresó al almacén o al inventario de la empresa. Debe ingresarse obligatoriamente (NO puede quedar vacío), debe ser una fecha válida y no puede ser una fecha futura (es decir, = la fecha actual).

[ubicacion_almacen] varchar(50) NOT NULL, -- Aquí se almacena la ubicación física del repuesto dentro del almacén. Es opcional, puede quedar vacío, y se usará para identificar el estante o pasillo correspondiente.

[categoria] varchar(50) NOT NULL, -- Aquí se almacena la categoría a la que pertenece el repuesto (ejemplo: motor, suspensión, eléctrico). Es obligatoria y debe contener únicamente valores predefinidos.

[estado_repuesto] varchar(20) NOT NULL -- Aquí se almacena el estado actual del repuesto en el inventario. Solo se permiten valores predefinidos como disponible o agotado. Es obligatorio y no puede quedar vacío.

)
GO

CREATE TABLE [Tbl_Servicio]

(

[id_servicio] int NOT NULL, -- Aquí se almacena el identificador numérico único de cada servicio registrado en la base de datos. No puede repetirse, debe ser un número entero positivo y no puede quedar vacío. Este campo sirve como clave primaria para identificar cada registro de servicio.

[nombre] varchar(50) NOT NULL, -- Aquí se almacena el nombre del servicio ofrecido por la empresa (por ejemplo: "Cambio de aceite", "Alineación y balanceo"). Debe contener caracteres alfabéticos, no puede quedar vacío ni iniciar con un carácter especial.

[descripcion] varchar(255) NULL, -- Aquí se almacena una descripción breve y clara del servicio. Es opcional, puede quedar vacío, pero si se ingresa debe tener un máximo de 255 caracteres.

[costo] decimal(10,2) NOT NULL, -- Aquí se almacena el costo base del servicio. Debe ser un número decimal mayor o igual a cero y no puede quedar vacío.

[duracion_horas] int NOT NULL, -- Aquí se almacena la duración aproximada del servicio expresada en minutos. Debe ser un número entero mayor que cero y no puede quedar vacío.

[garantia_meses] int NOT NULL, -- Aquí se almacena el tiempo de garantía que ofrece el servicio (en meses). Debe ser un número entero mayor o igual a cero. Según el tipo de servicio.

[requiere_repuesto] varchar(10) NOT NULL, -- Aquí se indica si el servicio requiere repuestos para ser realizado. Solo se permiten valores predefinidos: Sí o No. Es obligatorio.

[nivel_dificultad] varchar(50) NOT NULL, -- Aquí se almacena el nivel de dificultad del servicio. Solo se permiten valores predefinidos como básico, medio o avanzado. Es obligatorio.

[fecha_creacion] date NOT NULL, -- Aquí se almacena la fecha en que el servicio fue creado en el sistema. Es obligatoria, no puede quedar vacía y no puede ser mayor a la fecha actual.

[observaciones] varchar(255) NULL -- Aquí se almacena cualquier comentario adicional sobre el servicio. Este campo es opcional y puede quedar vacío.

)
GO

CREATE TABLE [Tbl_ServicioRealizado]

(

[id_servicio_realizado] int NOT NULL, -- Aquí se almacena el identificador numérico único de cada servicio realizado en el taller. Es la clave primaria de la tabla y permite distinguir de manera única cada registro de servicio. Debe ser un número entero positivo mayor que cero y no puede quedar vacío.

[id_empleado] int NOT NULL, -- Aquí se almacena el identificador del empleado encargado de realizar el servicio. Es obligatorio y funciona como una clave foránea que enlaza con la tabla Empleado, garantizando que solo empleados registrados puedan ser asignados a un servicio

[fecha_servicio] date NOT NULL, -- Aquí se almacena la fecha exacta en la que se realizó el servicio. Debe ingresarse en formato de fecha válido, no puede quedar vacío y no puede ser posterior a la fecha actual.

[id_repuesto] int NULL, -- Aquí se almacena el identificador del repuesto utilizado en el servicio, en caso de que se requiera. Es una clave foránea que referencia la tabla Repuesto. Este campo puede quedar vacío si no se utilizó ningún repuesto.

[cantidad_repuesto] int NULL, -- Aquí se almacena la cantidad de unidades de repuesto utilizadas en el servicio. Debe ser un número entero mayor o igual a cero. Es opcional, pero si existe un id_repuesto, este campo debe contener un valor mayor a cero.

[costo_total] decimal(12,2) NOT NULL, -- Aquí se almacena el costo total del servicio realizado, incluyendo repuestos y mano de obra. Debe ser un número decimal mayor o igual a cero y no puede quedar vacío.

[estado_servicio] varchar(20) NOT NULL, -- Aquí se almacena el estado actual del servicio. Solo se permiten valores predefinidos como pendiente o finalizado. Es obligatorio y no puede quedar vacío.

[observaciones] varchar(20) NULL, -- Aquí se almacenan comentarios adicionales relacionados con el servicio realizado (ejemplo: detalles del diagnóstico, recomendaciones al cliente). Es un campo opcional y puede quedar vacío.

[id_cita_servicio] int NOT NULL, -- FK que relaciona el servicio realizado con la cita que lo originó, es obligatoria y proviene de la tabla Tbl_CitaServicio

[id_respuesto] int NULL,

[id_cita] int NOT NULL -- Funciona como FK, relacionando la tabla ServicioRealizado con la cita que lo originó, ubicada en la Tbl_CitaServicio
)
GO

CREATE TABLE [Tbl_Vehiculo]

(

[id_vehiculo] int NOT NULL, -- Aquí se almacena el identificador numérico único de cada vehículo registrado en la base de datos. No puede repetirse, debe ser un número entero positivo y no puede quedar vacío. Este campo sirve como clave primaria para identificar cada registro de vehículo.

[numero_chasis] varchar(50) NOT NULL, -- Aquí se guarda el número de chasis único del vehículo. Es un valor de texto (hasta 50 caracteres) que identifica de manera exclusiva al vehículo. No puede quedar vacío ni repetirse en otro vehículo.

[marca] varchar(50) NOT NULL, -- Aquí se guarda la marca del vehículo (por ejemplo, Toyota, Honda, Nissan). Es un texto obligatorio y no puede iniciar con caracteres especiales ni estar vacío.

[modelo] varchar(50) NOT NULL, -- Aquí se almacena el modelo del vehículo (por ejemplo, Corolla, Civic, Frontier). Es un texto obligatorio y no puede iniciar con caracteres especiales ni quedar vacío.

[anio] varchar(50) NOT NULL, -- Aquí se guarda el año de fabricación del vehículo. Debe ser un número entero comprendido entre 1900 y el año actual. No puede quedar vacío.

[color] varchar(30) NOT NULL, -- Aquí se almacena el color del vehículo (por ejemplo, "rojo", "negro metálico"). Es un texto obligatorio y no puede quedar vacío.

[tipo_combustible] varchar(20) NOT NULL, -- Aquí se guarda el tipo de combustible que utiliza el vehículo. Solo se permiten valores predefinidos como "gasolina", "diésel", "eléctrico", "híbrido". Es obligatorio.

[tipo_transmision] varchar(20) NOT NULL, -- Aquí se almacena el tipo de transmisión del vehículo. Solo se permiten valores predefinidos como "automático", "mecánico" o "semiautomático". Es obligatorio.

[precio] decimal(12,2) NOT NULL, -- Aquí se guarda el precio de venta del vehículo. Es un número decimal (con dos decimales) mayor o igual a cero. No puede quedar vacío.

[estado_vehiculo] varchar(20) NOT NULL, -- Aquí se almacena el estado actual del vehículo dentro del inventario. Solo se permiten valores predefinidos como "disponible", "vendido" o "reservado". Es obligatorio.

[fecha_ingreso] date NOT NULL, -- Aquí se guarda la fecha en que el vehículo ingresó al inventario de la empresa. No puede ser mayor a la fecha actual ni quedar vacío.

[kilometraje] int NOT NULL -- Aquí se almacena el kilometraje actual del vehículo. Es un número entero mayor o igual a 0 y menor o igual a 1,000,000. No puede quedar vacío.

)
GO

CREATE TABLE [Tbl_VehiculoExterno]

(

[id_vehiculo_externo] int NOT NULL, -- Aquí se almacena el identificador numérico único de cada vehículo externo registrado en la base de datos. Este campo sirve como clave primaria y permite diferenciar de manera inequívoca a cada vehículo externo. Debe ser un número entero positivo mayor que cero y no puede quedar vacío.

[placa] varchar(50) NOT NULL, -- Aquí se almacena el número de placa oficial del vehículo externo. Este valor identifica de forma única al vehículo y se utiliza como referencia principal para búsquedas y validaciones. No puede quedar vacío, no debe contener caracteres especiales fuera del formato de placa, y debe ser único para evitar duplicados en la base de datos.

[marca] varchar(50) NOT NULL, -- Aquí se almacena la marca comercial del vehículo externo, como por ejemplo Toyota, Nissan, Honda, etc. Debe ingresarse siempre con caracteres alfabéticos, puede incluir espacios, pero no puede iniciar con un carácter especial ni con un número. Este campo es obligatorio.

[modelo] varchar(50) NOT NULL, -- Aquí se almacena el modelo específico del vehículo externo, como Corolla, Civic, Hilux, entre otros. Debe ser un valor de texto obligatorio, no puede quedar vacío y debe permitir la combinación de letras y números, aunque no puede iniciar con un carácter especial.

[año] int NOT NULL, -- Aquí se almacena el año de fabricación del vehículo externo. Debe ser un número entero de cuatro dígitos comprendido entre 1900 y el año actual, para garantizar la validez del dato. No puede quedar vacío.

[color] varchar(50) NOT NULL, -- Aquí se almacena el color principal del vehículo externo (ejemplo: negro, rojo metálico, blanco perla). Es un valor obligatorio, no puede quedar vacío y debe contener únicamente letras y, opcionalmente, combinaciones como "gris claro" o "azul oscuro".

[propietario] varchar(100) NOT NULL -- Aquí se almacena el nombre completo del propietario del vehículo externo. Debe contener únicamente caracteres alfabéticos, puede incluir espacios y guiones, pero no puede iniciar con un carácter especial ni con un número. Es obligatorio y no puede quedar vacío.

)
GO

CREATE TABLE [Tbl_Venta]

(
 [id_venta] int NOT NULL, -- Aquí se almacena el identificador numérico único de cada venta registrada en la base de datos. No puede repetirse, debe ser un número entero positivo y no puede quedar vacío. Este campo sirve como clave primaria para identificar cada registro de venta.

 [id_vehiculo] int NOT NULL, -- Aquí se almacena el identificador del vehículo vendido. Es un dato obligatorio y corresponde a una clave foránea que enlaza con la tabla Vehículo. Debe existir previamente en la tabla Vehículo.

 [id_cliente] int NOT NULL, -- Aquí se almacena el identificador del cliente que realizó la compra. Es un dato obligatorio y corresponde a una clave foránea que enlaza con la tabla Cliente. Debe existir previamente en la tabla Cliente.

 [id_empleado] int NOT NULL, -- Aquí se almacena el identificador del empleado que realizó la venta. Es un dato obligatorio y corresponde a una clave foránea que enlaza con la tabla Empleado. Debe existir previamente en la tabla Empleado.

 [fecha_venta] date NOT NULL, -- Aquí se almacena la fecha en que se realizó la venta. Es obligatoria, no puede quedar vacía y no puede ser mayor a la fecha actual.

 [precio_venta] decimal(12,2) NOT NULL, -- Aquí se almacena el precio final de venta del vehículo. Debe ser un número decimal mayor o igual a cero y no puede quedar vacío.

 [forma_pago] varchar(20) NOT NULL, -- Aquí se almacena la forma de pago acordada con el cliente. Solo se permiten valores predefinidos como contado o crédito. Es obligatoria y no puede quedar vacía.

 [estado_pago] varchar(20) NOT NULL, -- Aquí se almacena el estado actual del pago de la venta. Solo se permiten valores predefinidos como pagado o pendiente. Es obligatorio y no puede quedar vacío.

 [observaciones] varchar(255) NULL -- Aquí se almacena cualquier comentario adicional relacionado con la venta. Este campo es opcional y puede quedar vacío.

)
GO

/* Create Primary Keys, Indexes, Uniques, Checks */

ALTER TABLE [Tbl_CitaServicio]
 ADD CONSTRAINT [PK_Tbl_CitaServicio]
 PRIMARY KEY CLUSTERED ([id_cita] ASC)
GO

ALTER TABLE [Tbl_CitaServicio]
 ADD CONSTRAINT [UQ_Tbl_CitaServicio_IdCita] UNIQUE NONCLUSTERED ([id_cita]
ASC)

GO

```
ALTER TABLE [Tbl_CitaServicio]
ADD CONSTRAINT [CK_Tbl_CitaServicio_Fecha] CHECK (fecha_cita >=
CAST(GETDATE() AS DATE))
GO
```

```
ALTER TABLE [Tbl_CitaServicio]
ADD CONSTRAINT [CK_Tbl_CitaServicio_Estado] CHECK (estado IN
('Pendiente','Confirmada','Cancelada','Atendida'))
GO
```

```
ALTER TABLE [Tbl_CitaServicio]
ADD CONSTRAINT [CK_Tbl_CitaServicio_Vehiculo] CHECK ((id_vehiculo IS NOT NULL
OR id_vehiculo_externo IS NOT NULL))
GO
```

```
CREATE NONCLUSTERED INDEX [IXFK_Tbl_CitaServicio_Tbl_Cliente]
ON [Tbl_CitaServicio] ([id_cliente] ASC)
GO
```

```
CREATE NONCLUSTERED INDEX [IXFK_Tbl_CitaServicio_Tbl_Servicio]
ON [Tbl_CitaServicio] ([id_servicio] ASC)
GO
```

```
CREATE NONCLUSTERED INDEX [IXFK_Tbl_CitaServicio_Tbl_Vehiculo]
ON [Tbl_CitaServicio] ([id_vehiculo] ASC)
GO
```

```
CREATE NONCLUSTERED INDEX [IXFK_Tbl_CitaServicio_Tbl_VehiculoExterno]
ON [Tbl_CitaServicio] ([id_vehiculo_externo] ASC)
GO
```

```
ALTER TABLE [Tbl_Cliente]
ADD CONSTRAINT [PK_Cliente]
PRIMARY KEY CLUSTERED ([id_cliente] ASC)
GO
```

```
ALTER TABLE [Tbl_Cliente]
ADD CONSTRAINT [UQ_Cliente_Email] UNIQUE NONCLUSTERED ([email] ASC)
GO
```

```
ALTER TABLE [Tbl_Cliente]
ADD CONSTRAINT [UQ_Cliente_IdCliente] UNIQUE NONCLUSTERED ([id_cliente] ASC)
GO
```

```
ALTER TABLE [Tbl_Cliente]
ADD CONSTRAINT [CK_Cliente_IdCliente] CHECK (id_cliente > 0)
```

GO

```
ALTER TABLE [Tbl_Cliente]
ADD CONSTRAINT [CK_Cliente_Email_Format] CHECK (email IS NULL OR
(CHARINDEX('@', email) > 1 AND CHARINDEX('.', email, CHARINDEX('@', email)) >
CHARINDEX('@', email) + 1 AND email NOT LIKE '% %'))
```

GO

```
ALTER TABLE [Tbl_Cliente]
ADD CONSTRAINT [CK_Cliente_Nombre_IniciaLetra] CHECK (nombre IS NOT NULL AND
nombre LIKE '[A-Za-z]%')
```

GO

```
ALTER TABLE [Tbl_Cliente]
ADD CONSTRAINT [CK_Cliente_Apellido_IniciaLetra] CHECK (apellido IS NOT NULL AND
apellido LIKE '[A-Za-z]%')
```

GO

```
ALTER TABLE [Tbl_Cliente]
ADD CONSTRAINT [CK_Cliente_Telefono_Format] CHECK (telefono IS NULL OR
(PATINDEX('%[^0-9 +()-]%', telefono) = 0 AND (telefono LIKE '[0-9]%' OR telefono LIKE
'+%') AND LEN(telefono) <= 20))
```

GO

```
ALTER TABLE [Tbl_Cliente]
ADD CONSTRAINT [CK_Cliente_CodigoPostal_Formato] CHECK (codigo_postal IS NULL
OR (PATINDEX('%[^A-Za-z0-9 -]%', codigo_postal) = 0 AND LEN(codigo_postal) <= 10))
```

GO

```
ALTER TABLE [Tbl_Cliente]
ADD CONSTRAINT [CK_Cliente_FechaRegistro] CHECK (fecha_registro <=
CAST(GETDATE() AS DATE))
```

GO

```
ALTER TABLE [Tbl_Cliente]
ADD CONSTRAINT [CK_Cliente_TipoCliente] CHECK (tipo_cliente IN
('particular','empresa'))
```

GO

```
ALTER TABLE [Tbl_DetalleOrdenCompra]
ADD CONSTRAINT [PK_Tbl_DetalleOrdenCompra]
PRIMARY KEY CLUSTERED ([id_detalle_orden_compra] ASC)
```

GO

```
ALTER TABLE [Tbl_DetalleOrdenCompra]
ADD CONSTRAINT [UQ_DetalleOrdenCompra_IdDetalleOrdenCompra] UNIQUE
NONCLUSTERED ([id_detalle_orden_compra] ASC)
```

GO

```
ALTER TABLE [Tbl_DetalleOrdenCompra]
ADD CONSTRAINT [CK_DetalleOrdenCompra_IdDetalle] CHECK
(id_detalle_orden_compra > 0)
GO
```

```
ALTER TABLE [Tbl_DetalleOrdenCompra]
ADD CONSTRAINT [CK_DetalleOrdenCompra_Cantidad] CHECK (cantidad > 0)
GO
```

```
ALTER TABLE [Tbl_DetalleOrdenCompra]
ADD CONSTRAINT [CK_DetalleOrdenCompra_PrecioUnitario] CHECK (precio_unitario >=
0)
GO
```

```
ALTER TABLE [Tbl_DetalleOrdenCompra]
ADD CONSTRAINT [CK_DetalleOrdenCompra_Subtotal] CHECK (subtotal >= 0 AND
subtotal = cantidad * precio_unitario)
GO
```

```
CREATE NONCLUSTERED INDEX [IXFK_Tbl_DetalleOrdenCompra_Tbl_OrdenCompra]
ON [Tbl_DetalleOrdenCompra] ([id_orden_compra] ASC)
GO
```

```
CREATE NONCLUSTERED INDEX [IXFK_Tbl_DetalleOrdenCompra_Tbl_Repuesto]
ON [Tbl_DetalleOrdenCompra] ([id_respuesto] ASC)
GO
```

```
ALTER TABLE [Tbl_Empleado]
ADD CONSTRAINT [PK_Empleado]
PRIMARY KEY CLUSTERED ([id_empleado] ASC)
GO
```

```
ALTER TABLE [Tbl_Empleado]
ADD CONSTRAINT [UQ_Empleado_Email] UNIQUE NONCLUSTERED ([email] ASC)
GO
```

```
ALTER TABLE [Tbl_Empleado]
ADD CONSTRAINT [UQ_Empleado_IdEmpleado] UNIQUE NONCLUSTERED
([id_empleado] ASC)
GO
```

```
ALTER TABLE [Tbl_Empleado]
ADD CONSTRAINT [CK_Empleado_IdEmpleado] CHECK (id_empleado > 0)
GO
```

```
ALTER TABLE [Tbl_Empleado]
```

```

ADD CONSTRAINT [CK_Empleado_Email_Format] CHECK (email IS NULL OR
(CHARINDEX('@', email) > 1 AND CHARINDEX('.', email, CHARINDEX('@', email)) >
CHARINDEX('@', email) + 1 AND email NOT LIKE '% %'))
GO

```

```

ALTER TABLE [Tbl_Empleado]
ADD CONSTRAINT [CK_Empleado_Nombre_IniciaLetra] CHECK (nombre IS NOT NULL
AND nombre LIKE '[A-Za-z]%')
GO

```

```

ALTER TABLE [Tbl_Empleado]
ADD CONSTRAINT [CK_Empleado_Apellido_IniciaLetra] CHECK (apellido IS NOT NULL
AND apellido LIKE '[A-Za-z]%')
GO

```

```

ALTER TABLE [Tbl_Empleado]
ADD CONSTRAINT [CK_Empleado_Cargo_Valores] CHECK (cargo IN
('vendedor','mecánico','gerente','administrativo'))
GO

```

```

ALTER TABLE [Tbl_Empleado]
ADD CONSTRAINT [CK_Empleado_Telefono_Format] CHECK (telefono IS NULL OR
(PATINDEX('%[^0-9 +()/-]%', telefono) = 0 AND (telefono LIKE '[0-9]%' OR telefono LIKE
'+%') AND LEN(telefono) <= 20))
GO

```

```

ALTER TABLE [Tbl_Empleado]
ADD CONSTRAINT [CK_Empleado_FechaContratacion] CHECK (fecha_contratacion <=
CAST(GETDATE() AS DATE))
GO

```

```

ALTER TABLE [Tbl_Empleado]
ADD CONSTRAINT [CK_Empleado_Salario] CHECK (salario >= 0)
GO

```

```

ALTER TABLE [Tbl_Empleado]
ADD CONSTRAINT [CK_Empleado_EstadoValores] CHECK (estado_empleado IN
('activo','inactivo'))
GO

```

```

ALTER TABLE [Tbl_Factura]
ADD CONSTRAINT [PK_Tbl_Factura]
PRIMARY KEY CLUSTERED ([id_factura] ASC)
GO

```

```

ALTER TABLE [Tbl_Factura]
ADD CONSTRAINT [UQ_Factura_IdFactura] UNIQUE NONCLUSTERED ([id_factura]
ASC)

```

GO

```
ALTER TABLE [Tbl_Factura]
ADD CONSTRAINT [UQ_Factura_NumeroRecibo] UNIQUE NONCLUSTERED
([numero_recibo] ASC)
GO
```

```
ALTER TABLE [Tbl_Factura]
ADD CONSTRAINT [CK_Factura_IdFactura] CHECK (id_factura > 0)
GO
```

```
ALTER TABLE [Tbl_Factura]
ADD CONSTRAINT [CK_Factura_FechaEmision] CHECK (fecha_emision <=
CAST(GETDATE() AS DATE))
GO
```

```
ALTER TABLE [Tbl_Factura]
ADD CONSTRAINT [CK_Factura_MontoTotal] CHECK (monto_total >= 0)
GO
```

```
ALTER TABLE [Tbl_Factura]
ADD CONSTRAINT [CK_Factura_EstadoPago] CHECK (estado_pago IN
('pendiente','pagado'))
GO
```

```
ALTER TABLE [Tbl_Factura]
ADD CONSTRAINT [CK_Factura_MetodoPago] CHECK (metodo_pago IS NULL OR
metodo_pago IN ('efectivo','tarjeta'))
GO
```

```
ALTER TABLE [Tbl_Factura]
ADD CONSTRAINT [CK_Factura_FechaVencimiento] CHECK (fecha_vencimiento >=
fecha_emision)
GO
```

```
ALTER TABLE [Tbl_Factura]
ADD CONSTRAINT [CK_Factura_PagoDatos] CHECK ((
(estado_pago = 'pendiente' AND metodo_pago IS NULL AND numero_recibo IS NULL)
OR
(estado_pago = 'pagado' AND metodo_pago IS NOT NULL AND numero_recibo IS NOT
NULL)
))
GO
```

```
CREATE NONCLUSTERED INDEX [IXFK_Tbl_Factura_Tbl_Empleado]
ON [Tbl_Factura] ([id_empleado] ASC)
GO
```

```
CREATE NONCLUSTERED INDEX [IXFK_Tbl_Factura_Tbl_Venta]
ON [Tbl_Factura] ([id_venta] ASC)
GO
```

```
ALTER TABLE [Tbl_OrdenCompra]
ADD CONSTRAINT [PK_Tbl_OrdenCompra]
PRIMARY KEY CLUSTERED ([id_orden_compra] ASC)
GO
```

```
ALTER TABLE [Tbl_OrdenCompra]
ADD CONSTRAINT [UQ_OrdenCompra_IdOrdenCompra] UNIQUE NONCLUSTERED
([id_orden_compra] ASC)
GO
```

```
ALTER TABLE [Tbl_OrdenCompra]
ADD CONSTRAINT [CK_OrdenCompra_IdOrdenCompra] CHECK (id_orden_compra > 0)
GO
```

```
ALTER TABLE [Tbl_OrdenCompra]
ADD CONSTRAINT [CK_OrdenCompra_FechaOrden] CHECK (fecha_orden <=
CAST(GETDATE() AS DATE))
GO
```

```
ALTER TABLE [Tbl_OrdenCompra]
ADD CONSTRAINT [CK_OrdenCompra_FechaEntrega] CHECK (fecha_entrega IS NULL
OR fecha_entrega >= fecha_orden)
GO
```

```
ALTER TABLE [Tbl_OrdenCompra]
ADD CONSTRAINT [CK_OrdenCompra_EstadoOrden] CHECK (estado_orden IN
('pendiente','recibida'))
GO
```

```
ALTER TABLE [Tbl_OrdenCompra]
ADD CONSTRAINT [CK_OrdenCompra_TotalOrden] CHECK (total_orden >= 0)
GO
```

```
ALTER TABLE [Tbl_OrdenCompra]
ADD CONSTRAINT [CK_OrdenCompra_MetodoPago] CHECK (metodo_pago IN
('efectivo','transferencia'))
GO
```

```
ALTER TABLE [Tbl_OrdenCompra]
ADD CONSTRAINT [CK_OrdenCompra_FormaEnvio] CHECK (forma_envio IN
('terrestre','aéreo','marítimo'))
GO
```

```
CREATE NONCLUSTERED INDEX [IXFK_Tbl_OrdenCompra_Tbl_Empleado]
```



```
ON [Tbl_OrdenCompra] ([id_empleado] ASC)
GO
```

```
CREATE NONCLUSTERED INDEX [IXFK_Tbl_OrdenCompra_Tbl_Proveedor]
ON [Tbl_OrdenCompra] ([id_proveedor] ASC)
GO
```

```
ALTER TABLE [Tbl_Proveedor]
ADD CONSTRAINT [PK_Tbl_Proveedor]
PRIMARY KEY CLUSTERED ([id_proveedor] ASC)
GO
```

```
ALTER TABLE [Tbl_Proveedor]
ADD CONSTRAINT [UQ_Proveedor_Email] UNIQUE NONCLUSTERED ([email] ASC)
GO
```

```
ALTER TABLE [Tbl_Proveedor]
ADD CONSTRAINT [UQ_Proveedor_IdProveedor] UNIQUE NONCLUSTERED
([id_proveedor] ASC)
GO
```

```
ALTER TABLE [Tbl_Proveedor]
ADD CONSTRAINT [CK_Proveedor_IdProveedor] CHECK (id_proveedor > 0)
GO
```

```
ALTER TABLE [Tbl_Proveedor]
ADD CONSTRAINT [CK_Proveedor_Nombre] CHECK (LEN(LTRIM(RTRIM(nombre)))) > 0)
GO
```

```
ALTER TABLE [Tbl_Proveedor]
ADD CONSTRAINT [CK_Proveedor_Telefono] CHECK (telefono IS NULL OR
(LEN(telefono) <= 20 AND telefono NOT LIKE '%[^0-9+() -]%' ))
GO
```

```
ALTER TABLE [Tbl_Proveedor]
ADD CONSTRAINT [CK_Proveedor_EmailFormato] CHECK (email IS NULL OR email
LIKE '%_@_%._%')
GO
```

```
ALTER TABLE [Tbl_Proveedor]
ADD CONSTRAINT [CK_Proveedor_Direccion] CHECK (LEN(LTRIM(RTRIM(direccion)))) >
0)
GO
```

```
ALTER TABLE [Tbl_Proveedor]
ADD CONSTRAINT [CK_Proveedor_Ciudad] CHECK (LEN(LTRIM(RTRIM(ciudad)))) > 0)
GO
```

```
ALTER TABLE [Tbl_Proveedor]
ADD CONSTRAINT [CK_Proveedor_EstadoProveedor] CHECK
(LEN(LTRIM(RTRIM(estado_proveedor))) > 0)
GO
```

```
ALTER TABLE [Tbl_Proveedor]
ADD CONSTRAINT [CK_Proveedor_CodigoPostal] CHECK (codigo_postal IS NULL OR
(LEN(codigo_postal) <= 10 AND codigo_postal NOT LIKE '%[^A-Za-z0-9 -]%' ))
GO
```

```
ALTER TABLE [Tbl_Proveedor]
ADD CONSTRAINT [CK_Proveedor_ContactoPrincipal] CHECK
(LEN(LTRIM(RTRIM(contacto_principal))) > 0)
GO
```

```
ALTER TABLE [Tbl_Proveedor]
ADD CONSTRAINT [CK_Proveedor_FechaRegistro] CHECK (fecha_registro <=
CAST(GETDATE() AS DATE))
GO
```

```
ALTER TABLE [Tbl_Proveedor]
ADD CONSTRAINT [CK_Proveedor_TipoProveedor] CHECK (tipo_proveedor IN
('local','internacional'))
GO
```

```
ALTER TABLE [Tbl_Repuesto]
ADD CONSTRAINT [PK_Tbl_Repuesto]
PRIMARY KEY CLUSTERED ([id_vehiculorepuesto] ASC)
GO
```

```
ALTER TABLE [Tbl_Repuesto]
ADD CONSTRAINT [UQ_Repuesto_Codigo] UNIQUE NONCLUSTERED
([codigo_repuesto] ASC)
GO
```

```
ALTER TABLE [Tbl_Repuesto]
ADD CONSTRAINT [UQ_Repuesto_IdRepuesto] UNIQUE NONCLUSTERED
([id_vehiculorepuesto] ASC)
GO
```

```
ALTER TABLE [Tbl_Repuesto]
ADD CONSTRAINT [CK_Repuesto_IdRepuesto] CHECK (id_vehiculorepuesto > 0)
GO
```

```
ALTER TABLE [Tbl_Repuesto]
ADD CONSTRAINT [CK_Repuesto_Nombre_IniciaLetra] CHECK (nombre IS NOT NULL
AND nombre LIKE '[A-Za-z]%' )
GO
```

```
ALTER TABLE [Tbl_Repuesto]
ADD CONSTRAINT [CK_Repuesto_Categoria] CHECK (categoria IN
('motor','suspensión','eléctrico','carrocería'))
GO
```

```
ALTER TABLE [Tbl_Repuesto]
ADD CONSTRAINT [CK_Repuesto_PrecioUnitario] CHECK (precio_unitario >= 0)
GO
```

```
ALTER TABLE [Tbl_Repuesto]
ADD CONSTRAINT [CK_Repuesto_Stock] CHECK (stock >= 0)
GO
```

```
ALTER TABLE [Tbl_Repuesto]
ADD CONSTRAINT [CK_Repuesto_FechaIngreso] CHECK (fecha_ingreso IS NOT NULL
AND fecha_ingreso <= CAST(GETDATE() AS DATE))
GO
```

```
ALTER TABLE [Tbl_Repuesto]
ADD CONSTRAINT [CK_Repuesto_Estado] CHECK (estado_repuesto IN
('disponible','agotado'))
GO
```

```
CREATE NONCLUSTERED INDEX [IXFK_Tbl_Repuesto_Tbl_Proveedor]
ON [Tbl_Repuesto] ([id_proveedor] ASC)
GO
```

```
ALTER TABLE [Tbl_Servicio]
ADD CONSTRAINT [PK_Tbl_Servicio]
PRIMARY KEY CLUSTERED ([id_servicio] ASC)
GO
```

```
ALTER TABLE [Tbl_Servicio]
ADD CONSTRAINT [UQ_Servicio_IdServicio] UNIQUE NONCLUSTERED ([id_servicio]
ASC)
GO
```

```
ALTER TABLE [Tbl_Servicio]
ADD CONSTRAINT [CK_Servicio_IdServicio] CHECK (id_servicio > 0)
GO
```

```
ALTER TABLE [Tbl_Servicio]
ADD CONSTRAINT [CK_Servicio_Nombre_IniciaLetra] CHECK (nombre IS NOT NULL
AND nombre LIKE '[A-Za-z]%)')
GO
```

```
ALTER TABLE [Tbl_Servicio]
```

```
ADD CONSTRAINT [CK_Servicio_Costo] CHECK (costo >= 0)
GO
```

```
ALTER TABLE [Tbl_Servicio]
ADD CONSTRAINT [CK_Servicio_Duracion] CHECK (duracion_horas > 0)
GO
```

```
ALTER TABLE [Tbl_Servicio]
ADD CONSTRAINT [CK_Servicio_Garantia] CHECK (garantia_meses >= 0)
GO
```

```
ALTER TABLE [Tbl_Servicio]
ADD CONSTRAINT [CK_Servicio_RequiereRepuesto] CHECK (requiere_repuesto IN
('Sí','No'))
GO
```

```
ALTER TABLE [Tbl_Servicio]
ADD CONSTRAINT [CK_Servicio_NivelDificultad] CHECK (nivel_dificultad IN
('básico','medio','avanzado'))
GO
```

```
ALTER TABLE [Tbl_Servicio]
ADD CONSTRAINT [CK_Servicio_FechaCreacion] CHECK (fecha_creacion <=
CAST(GETDATE() AS DATE))
GO
```

```
ALTER TABLE [Tbl_ServicioRealizado]
ADD CONSTRAINT [PK_Tbl_ServicioRealizado]
PRIMARY KEY CLUSTERED ([id_servicio_realizado] ASC)
GO
```

```
ALTER TABLE [Tbl_ServicioRealizado]
ADD CONSTRAINT [UQ_ServReal_IdServicioRealizado] UNIQUE NONCLUSTERED
([id_servicio_realizado] ASC)
GO
```

```
ALTER TABLE [Tbl_ServicioRealizado]
ADD CONSTRAINT [CK_ServReal_Id] CHECK (id_servicio_realizado > 0)
GO
```

```
ALTER TABLE [Tbl_ServicioRealizado]
ADD CONSTRAINT [CK_ServReal_Fecha] CHECK (fecha_servicio <= CAST(GETDATE()
AS DATE))
GO
```

```
ALTER TABLE [Tbl_ServicioRealizado]
ADD CONSTRAINT [CK_ServReal_Cantidad] CHECK (cantidad_repuesto IS NULL OR
cantidad_repuesto >= 0)
```

GO

```
ALTER TABLE [Tbl_ServicioRealizado]
ADD CONSTRAINT [CK_ServReal_Costo] CHECK (costo_total >= 0)
GO
```

```
ALTER TABLE [Tbl_ServicioRealizado]
ADD CONSTRAINT [CK_ServReal_Estado] CHECK (estado_servicio IN
('pendiente','finalizado'))
GO
```

```
CREATE NONCLUSTERED INDEX [IXFK_Tbl_ServicioRealizado_Tbl_CitaServicio]
ON [Tbl_ServicioRealizado] ([id_cita] ASC)
GO
```

```
CREATE NONCLUSTERED INDEX [IXFK_Tbl_ServicioRealizado_Tbl_Empleado]
ON [Tbl_ServicioRealizado] ([id_empleado] ASC)
GO
```

```
CREATE NONCLUSTERED INDEX [IXFK_Tbl_ServicioRealizado_Tbl_Repuesto]
ON [Tbl_ServicioRealizado] ([id_respuesto] ASC)
GO
```

```
ALTER TABLE [Tbl_Vehiculo]
ADD CONSTRAINT [PK_Vehiculo]
PRIMARY KEY CLUSTERED ([id_vehiculo] ASC)
GO
```

```
ALTER TABLE [Tbl_Vehiculo]
ADD CONSTRAINT [UQ_Vehiculo_id] UNIQUE NONCLUSTERED ([id_vehiculo] ASC)
GO
```

```
ALTER TABLE [Tbl_Vehiculo]
ADD CONSTRAINT [UQ_Vehiculo_NumeroChasis] UNIQUE NONCLUSTERED
([numero_chasis] ASC)
GO
```

```
ALTER TABLE [Tbl_Vehiculo]
ADD CONSTRAINT [CK_Vehiculo_IdVehiculo] CHECK (id_vehiculo > 0)
GO
```

```
ALTER TABLE [Tbl_Vehiculo]
ADD CONSTRAINT [CK_Vehiculo_Anio] CHECK (anio >= 1900 AND anio <=
YEAR(GETDATE()))
GO
```

```
ALTER TABLE [Tbl_Vehiculo]
ADD CONSTRAINT [CK_Vehiculo_Precio] CHECK (precio >= 0)
```

GO

```
ALTER TABLE [Tbl_Vehiculo]
ADD CONSTRAINT [CK_Vehiculo_Kilometraje] CHECK (kilometraje >= 0 AND kilometraje
<= 1000000)
GO
```

```
ALTER TABLE [Tbl_Vehiculo]
ADD CONSTRAINT [CK_Vehiculo_TipoCombustible] CHECK (tipo_combustible IN
('gasolina','diesel','eléctrico','híbrido'))
GO
```

```
ALTER TABLE [Tbl_Vehiculo]
ADD CONSTRAINT [CK_Vehiculo_TipoTransmision] CHECK (tipo_transmision IN
('automático','mecánico','semiautomático'))
GO
```

```
ALTER TABLE [Tbl_Vehiculo]
ADD CONSTRAINT [CK_Vehiculo_Estado] CHECK (estado_vehiculo IN
('disponible','vendido','reservado'))
GO
```

```
ALTER TABLE [Tbl_Vehiculo]
ADD CONSTRAINT [CK_Vehiculo_FechaIngreso] CHECK (fecha_ingreso <=
CAST(GETDATE() AS DATE))
GO
```

```
ALTER TABLE [Tbl_VehiculoExterno]
ADD CONSTRAINT [PK_Tbl_VehiculoExterno]
PRIMARY KEY CLUSTERED ([id_vehiculo_externo] ASC)
GO
```

```
ALTER TABLE [Tbl_VehiculoExterno]
ADD CONSTRAINT [UQ_VehiculoExterno_Placa] UNIQUE NONCLUSTERED ([placa]
ASC)
GO
```

```
ALTER TABLE [Tbl_VehiculoExterno]
ADD CONSTRAINT [CK_VehiculoExterno_Id] CHECK (id_vehiculo_externo > 0)
GO
```

```
ALTER TABLE [Tbl_VehiculoExterno]
ADD CONSTRAINT [CK_VehiculoExterno_Placa] CHECK (placa NOT LIKE '%[^A-Z0-9]%' )
GO
```

```
ALTER TABLE [Tbl_VehiculoExterno]
ADD CONSTRAINT [CK_VehiculoExterno_Año] CHECK (año >= 1900 AND año <=
YEAR(GETDATE()))
```

GO

```
ALTER TABLE [Tbl_VehiculoExterno]
  ADD CONSTRAINT [CK_VehiculoExterno_Propietario] CHECK (propietario NOT LIKE
    '['^A-Za-z]%' )
GO
```

```
ALTER TABLE [Tbl_Venta]
  ADD CONSTRAINT [PK_Venta]
    PRIMARY KEY CLUSTERED ([id_venta] ASC)
GO
```

```
ALTER TABLE [Tbl_Venta]
  ADD CONSTRAINT [CK_Venta_IdVenta] CHECK (id_venta > 0)
GO
```

```
ALTER TABLE [Tbl_Venta]
  ADD CONSTRAINT [CK_Venta_FechaVenta] CHECK (fecha_venta <= CAST(GETDATE()
    AS DATE))
GO
```

```
ALTER TABLE [Tbl_Venta]
  ADD CONSTRAINT [CK_Venta_PrecioVenta] CHECK (precio_venta >= 0)
GO
```

```
ALTER TABLE [Tbl_Venta]
  ADD CONSTRAINT [CK_Venta_FormaPago] CHECK (forma_pago IN ('contado','crédito'))
GO
```

```
ALTER TABLE [Tbl_Venta]
  ADD CONSTRAINT [CK_Venta_EstadoPago] CHECK (estado_pago IN
    ('pagado','pendiente'))
GO
```

```
CREATE NONCLUSTERED INDEX [IXFK_Tbl_Venta_Tbl_Cliente]
  ON [Tbl_Venta] ([id_cliente] ASC)
GO
```

```
CREATE NONCLUSTERED INDEX [IXFK_Tbl_Venta_Tbl_Empleado]
  ON [Tbl_Venta] ([id_empleado] ASC)
GO
```

```
CREATE NONCLUSTERED INDEX [IXFK_Tbl_Venta_Tbl_Vehiculo]
  ON [Tbl_Venta] ([id_vehiculo] ASC)
GO
```

/* Create Foreign Key Constraints */

```
ALTER TABLE [Tbl_CitaServicio] ADD CONSTRAINT [FK_Tbl_CitaServicio_Tbl_Cliente]
    FOREIGN KEY ([id_cliente]) REFERENCES [Tbl_Cliente] ([id_cliente]) ON DELETE
    No Action ON UPDATE No Action
GO
```

```
ALTER TABLE [Tbl_CitaServicio] ADD CONSTRAINT [FK_Tbl_CitaServicio_Tbl_Servicio]
    FOREIGN KEY ([id_servicio]) REFERENCES [Tbl_Servicio] ([id_servicio]) ON
    DELETE No Action ON UPDATE No Action
GO
```

```
ALTER TABLE [Tbl_CitaServicio] ADD CONSTRAINT [FK_Tbl_CitaServicio_Tbl_Vehiculo]
    FOREIGN KEY ([id_vehiculo]) REFERENCES [Tbl_Vehiculo] ([id_vehiculo]) ON
    DELETE No Action ON UPDATE No Action
GO
```

```
ALTER TABLE [Tbl_CitaServicio] ADD CONSTRAINT
[FK_Tbl_CitaServicio_Tbl_VehiculoExterno]
    FOREIGN KEY ([id_vehiculo_externo]) REFERENCES [Tbl_VehiculoExterno]
    ([id_vehiculo_externo]) ON DELETE No Action ON UPDATE No Action
GO
```

```
ALTER TABLE [Tbl_DetalleOrdenCompra] ADD CONSTRAINT
[FK_Tbl_DetalleOrdenCompra_Tbl_OrdenCompra]
    FOREIGN KEY ([id_orden_compra]) REFERENCES [Tbl_OrdenCompra]
    ([id_orden_compra]) ON DELETE No Action ON UPDATE No Action
GO
```

```
ALTER TABLE [Tbl_DetalleOrdenCompra] ADD CONSTRAINT
[FK_Tbl_DetalleOrdenCompra_Tbl_Repuesto]
    FOREIGN KEY ([id_respuesto]) REFERENCES [Tbl_Repuesto]
    ([id_vehiculorepuesto]) ON DELETE No Action ON UPDATE No Action
GO
```

```
ALTER TABLE [Tbl_Factura] ADD CONSTRAINT [FK_Tbl_Factura_Tbl_Empleado]
    FOREIGN KEY ([id_empleado]) REFERENCES [Tbl_Empleado] ([id_empleado]) ON
    DELETE No Action ON UPDATE No Action
GO
```

```
ALTER TABLE [Tbl_Factura] ADD CONSTRAINT [FK_Tbl_Factura_Tbl_Venta]
    FOREIGN KEY ([id_venta]) REFERENCES [Tbl_Venta] ([id_venta]) ON DELETE No
    Action ON UPDATE No Action
GO
```

```
ALTER TABLE [Tbl_OrdenCompra] ADD CONSTRAINT
[FK_Tbl_OrdenCompra_Tbl_Empleado]
    FOREIGN KEY ([id_empleado]) REFERENCES [Tbl_Empleado] ([id_empleado]) ON
    DELETE No Action ON UPDATE No Action
GO
```



```

ALTER TABLE [Tbl_OrdenCompra] ADD CONSTRAINT
[FK_Tbl_OrdenCompra_Tbl_Proveedor]
    FOREIGN KEY ([id_proveedor]) REFERENCES [Tbl_Proveedor] ([id_proveedor])
ON DELETE No Action ON UPDATE No Action
GO

```

```

ALTER TABLE [Tbl_Repuesto] ADD CONSTRAINT [FK_Tbl_Repuesto_Tbl_Proveedor]
    FOREIGN KEY ([id_proveedor]) REFERENCES [Tbl_Proveedor] ([id_proveedor])
ON DELETE No Action ON UPDATE No Action
GO

```

```

ALTER TABLE [Tbl_ServicioRealizado] ADD CONSTRAINT
[FK_Tbl_ServicioRealizado_Tbl_CitaServicio]
    FOREIGN KEY ([id_cita]) REFERENCES [Tbl_CitaServicio] ([id_cita]) ON DELETE
No Action ON UPDATE No Action
GO

```

```

ALTER TABLE [Tbl_ServicioRealizado] ADD CONSTRAINT
[FK_Tbl_ServicioRealizado_Tbl_Empleado]
    FOREIGN KEY ([id_empleado]) REFERENCES [Tbl_Empleado] ([id_empleado]) ON
DELETE No Action ON UPDATE No Action
GO

```

```

ALTER TABLE [Tbl_ServicioRealizado] ADD CONSTRAINT
[FK_Tbl_ServicioRealizado_Tbl_Repuesto]
    FOREIGN KEY ([id_respuesto]) REFERENCES [Tbl_Repuesto]
([id_vehiculorepuesto]) ON DELETE No Action ON UPDATE No Action
GO

```

```

ALTER TABLE [Tbl_Venta] ADD CONSTRAINT [FK_Tbl_Venta_Tbl_Cliente]
    FOREIGN KEY ([id_cliente]) REFERENCES [Tbl_Cliente] ([id_cliente]) ON DELETE
No Action ON UPDATE No Action
GO

```

```

ALTER TABLE [Tbl_Venta] ADD CONSTRAINT [FK_Tbl_Venta_Tbl_Empleado]
    FOREIGN KEY ([id_empleado]) REFERENCES [Tbl_Empleado] ([id_empleado]) ON
DELETE No Action ON UPDATE No Action
GO

```

```

ALTER TABLE [Tbl_Venta] ADD CONSTRAINT [FK_Tbl_Venta_Tbl_Vehiculo]
    FOREIGN KEY ([id_vehiculo]) REFERENCES [Tbl_Vehiculo] ([id_vehiculo]) ON
DELETE No Action ON UPDATE No Action
GO

```

```

/* Create Table Comments */

```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'table',
'Tbl_CitaServicio', NULL, NULL))
```

```
begin
```

```
EXEC sys.sp_updateextendedproperty 'MS_Description', 'La tabla Tbl_CitaServicio
almacena la información de las citas programadas por los clientes de NicaMotores para
recibir servicios de mantenimiento o reparación en sus vehículos. Permite llevar un control
de quién agenda la cita, para qué vehículo, qué servicio solicita, la fecha y hora asignada, y
el estado de la cita (pendiente, confirmada, cancelada o atendida). Este registro es
fundamental para la gestión de la agenda del taller, evitando duplicidades, mejorando la
organización y garantizando un adecuado seguimiento de los servicios solicitados por los
clientes.', 'SCHEMA', 'dbo', 'table', 'Tbl_CitaServicio'
```

```
end
```

```
else
```

```
begin
```

```
EXEC sys.sp_addextendedproperty 'MS_Description', 'La tabla Tbl_CitaServicio almacena
la información de las citas programadas por los clientes de NicaMotores para recibir
servicios de mantenimiento o reparación en sus vehículos. Permite llevar un control de
quién agenda la cita, para qué vehículo, qué servicio solicita, la fecha y hora asignada, y el
estado de la cita (pendiente, confirmada, cancelada o atendida). Este registro es
fundamental para la gestión de la agenda del taller, evitando duplicidades, mejorando la
organización y garantizando un adecuado seguimiento de los servicios solicitados por los
clientes.', 'SCHEMA', 'dbo', 'table', 'Tbl_CitaServicio'
```

```
end
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el identificador
numérico único de cada cita registrada en la base de datos. No puede repetirse, debe ser
un número entero positivo y no puede quedar vacío. Este campo sirve como clave primaria
para identificar de manera inequívoca cada cita de servicio agendada por un cliente.',
'Schema', [dbo], 'table', [Tbl_CitaServicio], 'column', [id_cita]
```

GO

```
EXEC sp_addextendedproperty 'MS_Description', 'Este campo almacena el identificador del
cliente que solicita la cita de servicio. Es un número entero que establece una relación con
la tabla Tbl_Cliente, garantizando que toda cita esté asociada a un cliente registrado en el
sistema. No puede quedar vacío, ya que toda cita debe corresponder a un cliente
específico.', 'Schema', [dbo], 'table', [Tbl_CitaServicio], 'column', [id_cliente]
```

GO

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el identificador del
vehículo que pertenece a NicaMotores y que será atendido en la cita. Es una clave foránea
hacia la tabla Tbl_Vehiculo. Este campo es opcional porque la cita puede ser para un
vehículo externo.', 'Schema', [dbo], 'table', [Tbl_CitaServicio], 'column', [id_vehiculo]
```

GO

```
EXEC sp_addextendedproperty 'MS_Description', 'Este atributo registra el identificador del
vehículo externo (no comprado en NicaMotores) que será atendido en la cita. Se relaciona
```

con la tabla Tbl_VehiculoExterno. Es opcional, ya que la cita también puede ser para un vehículo interno.', 'Schema', [dbo], 'table', [Tbl_CitaServicio], 'column', [id_vehiculo_externo]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Este campo guarda el identificador del servicio solicitado en la cita. Se relaciona con la tabla Tbl_Servicio y permite registrar si se trata de un mantenimiento, reparación u otro tipo de servicio. No puede quedar vacío porque toda cita debe estar vinculada a un servicio específico.', 'Schema', [dbo], 'table', [Tbl_CitaServicio], 'column', [id_servicio]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena la fecha agendada para la cita. Debe ser una fecha igual o posterior a la fecha actual (no puede estar en el pasado). No puede quedar vacía ya que toda cita debe tener definida una fecha de realización.', 'Schema', [dbo], 'table', [Tbl_CitaServicio], 'column', [fecha_cita]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Este atributo guarda la hora específica de la cita, lo que permite organizar el calendario de servicios en el taller. No puede estar vacío porque es necesario definir un horario para atender cada cita. En futuras versiones se deberá validar que no se programen más de dos citas en el mismo horario y que no existan traslapes de tiempo de acuerdo con la duración del servicio. Esto podrá implementarse mediante triggers o reglas de negocio en la aplicación.', 'Schema', [dbo], 'table', [Tbl_CitaServicio], 'column', [hora_cita]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Este campo almacena el estado actual de la cita, el cual puede tomar valores como Pendiente, Confirmada, Cancelada o Atendida. Permite dar seguimiento al ciclo de vida de la cita, desde que es agendada hasta que se atiende o se cancela. No puede estar vacío.', 'Schema', [dbo], 'table', [Tbl_CitaServicio], 'column', [estado]
GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_CitaServicio', 'CONSTRAINT', 'FK_Tbl_CitaServicio_Tbl_Cliente'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Garantiza que cada cita
esté asociada a un cliente válido dentro de la base de datos.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_CitaServicio', 'CONSTRAINT', 'FK_Tbl_CitaServicio_Tbl_Cliente'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Garantiza que cada cita
esté asociada a un cliente válido dentro de la base de datos.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_CitaServicio', 'CONSTRAINT', 'FK_Tbl_CitaServicio_Tbl_Cliente'
end
GO
```

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_CitaServicio', 'CONSTRAINT', 'FK_Tbl_CitaServicio_Tbl_Servicio'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Asocia la cita con el
servicio solicitado por el cliente.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_CitaServicio',
'CONSTRAINT', 'FK_Tbl_CitaServicio_Tbl_Servicio'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Asocia la cita con el servicio
solicitado por el cliente.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_CitaServicio', 'CONSTRAINT',
'FK_Tbl_CitaServicio_Tbl_Servicio'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_CitaServicio', 'CONSTRAINT', 'FK_Tbl_CitaServicio_Tbl_Vehiculo'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Relaciona la cita con un
vehículo vendido por NicaMotores. Este campo es opcional.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_CitaServicio', 'CONSTRAINT', 'FK_Tbl_CitaServicio_Tbl_Vehiculo'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Relaciona la cita con un
vehículo vendido por NicaMotores. Este campo es opcional.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_CitaServicio', 'CONSTRAINT', 'FK_Tbl_CitaServicio_Tbl_Vehiculo'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_CitaServicio', 'CONSTRAINT', 'FK_Tbl_CitaServicio_Tbl_VehiculoExterno'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Relaciona la cita con un
vehículo externo registrado en la base de datos. Este campo es opcional.', 'SCHEMA', 'dbo',
'TABLE', 'Tbl_CitaServicio', 'CONSTRAINT', 'FK_Tbl_CitaServicio_Tbl_VehiculoExterno'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Relaciona la cita con un
vehículo externo registrado en la base de datos. Este campo es opcional.', 'SCHEMA', 'dbo',
'TABLE', 'Tbl_CitaServicio', 'CONSTRAINT', 'FK_Tbl_CitaServicio_Tbl_VehiculoExterno'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_CitaServicio', 'INDEX', 'IXFK_Tbl_CitaServicio_Tbl_Cliente'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Índice asociado a la
restricción FK_Tbl_CitaServicio_Tbl_Cliente. Optimiza las búsquedas de citas programadas
por cliente, permitiendo acceder de manera rápida al historial de citas y agilizar reportes que
relacionen clientes con los servicios solicitados.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_CitaServicio', 'INDEX', 'IXFK_Tbl_CitaServicio_Tbl_Cliente'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Índice asociado a la
restricción FK_Tbl_CitaServicio_Tbl_Cliente. Optimiza las búsquedas de citas programadas
por cliente, permitiendo acceder de manera rápida al historial de citas y agilizar reportes que
relacionen clientes con los servicios solicitados.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_CitaServicio', 'INDEX', 'IXFK_Tbl_CitaServicio_Tbl_Cliente'
end

GO

```

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_CitaServicio', 'INDEX', 'IXFK_Tbl_CitaServicio_Tbl_Servicio'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Índice creado para la
restricción FK_Tbl_CitaServicio_Tbl_Servicio. Permite optimizar consultas relacionadas con
la agenda de servicios, verificando la disponibilidad y frecuencia de citas por tipo de
servicio, y contribuye a gestionar la carga de trabajo en el taller.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_CitaServicio', 'INDEX', 'IXFK_Tbl_CitaServicio_Tbl_Servicio'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Índice creado para la
restricción FK_Tbl_CitaServicio_Tbl_Servicio. Permite optimizar consultas relacionadas con
la agenda de servicios, verificando la disponibilidad y frecuencia de citas por tipo de
servicio, y contribuye a gestionar la carga de trabajo en el taller.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_CitaServicio', 'INDEX', 'IXFK_Tbl_CitaServicio_Tbl_Servicio'
end

GO

```

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_CitaServicio', 'INDEX', 'IXFK_Tbl_CitaServicio_Tbl_Vehiculo'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Índice asociado a la
restricción FK_Tbl_CitaServicio_Tbl_Vehiculo. Facilita las consultas de citas vinculadas a
vehículos vendidos por la empresa, permitiendo generar reportes de servicios preventivos o

```

```

mantenimientos programados por vehículo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_CitaServicio',
'INDEX', 'IXFK_Tbl_CitaServicio_Tbl_Vehiculo'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Índice asociado a la
restricción FK_Tbl_CitaServicio_Tbl_Vehiculo. Facilita las consultas de citas vinculadas a
vehículos vendidos por la empresa, permitiendo generar reportes de servicios preventivos o
mantenimientos programados por vehículo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_CitaServicio',
'INDEX', 'IXFK_Tbl_CitaServicio_Tbl_Vehiculo'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_CitaServicio', 'INDEX', 'IXFK_Tbl_CitaServicio_Tbl_VehiculoExterno'))
begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Índice creado para la
restricción FK_Tbl_CitaServicio_Tbl_VehiculoExterno. Mejora el rendimiento en la búsqueda
de citas de vehículos externos, garantizando trazabilidad y acceso rápido a la información
de clientes que no adquirieron vehículos directamente en NicaMotores.', 'SCHEMA', 'dbo',
'TABLE', 'Tbl_CitaServicio', 'INDEX', 'IXFK_Tbl_CitaServicio_Tbl_VehiculoExterno'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Índice creado para la
restricción FK_Tbl_CitaServicio_Tbl_VehiculoExterno. Mejora el rendimiento en la búsqueda
de citas de vehículos externos, garantizando trazabilidad y acceso rápido a la información
de clientes que no adquirieron vehículos directamente en NicaMotores.', 'SCHEMA', 'dbo',
'TABLE', 'Tbl_CitaServicio', 'INDEX', 'IXFK_Tbl_CitaServicio_Tbl_VehiculoExterno'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_CitaServicio', 'CONSTRAINT', 'PK_Tbl_CitaServicio'))
begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Asegura la unicidad de
cada cita de servicio, identificando de manera exclusiva cada registro.', 'SCHEMA', 'dbo',
'TABLE', 'Tbl_CitaServicio', 'CONSTRAINT', 'PK_Tbl_CitaServicio'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Asegura la unicidad de cada
cita de servicio, identificando de manera exclusiva cada registro.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_CitaServicio', 'CONSTRAINT', 'PK_Tbl_CitaServicio'
end

```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_CitaServicio', 'CONSTRAINT', 'CK_Tbl_CitaServicio_Fecha'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Garantiza que la cita no se
pueda agendar en una fecha pasada.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_CitaServicio',
'CONSTRAINT', 'CK_Tbl_CitaServicio_Fecha'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Garantiza que la cita no se
pueda agendar en una fecha pasada.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_CitaServicio',
'CONSTRAINT', 'CK_Tbl_CitaServicio_Fecha'
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_CitaServicio', 'CONSTRAINT', 'CK_Tbl_CitaServicio_Estado'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Restringe los valores
posibles del estado de la cita a opciones válidas.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_CitaServicio', 'CONSTRAINT', 'CK_Tbl_CitaServicio_Estado'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Restringe los valores
posibles del estado de la cita a opciones válidas.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_CitaServicio', 'CONSTRAINT', 'CK_Tbl_CitaServicio_Estado'
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_CitaServicio', 'CONSTRAINT', 'CK_Tbl_CitaServicio_Vehiculo'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Asegura que toda cita esté
asociada al menos a un vehículo (interno o externo), no pudiendo quedar ambos campos
vacíos.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_CitaServicio', 'CONSTRAINT',
'CK_Tbl_CitaServicio_Vehiculo'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Asegura que toda cita esté
asociada al menos a un vehículo (interno o externo), no pudiendo quedar ambos campos
vacíos.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_CitaServicio', 'CONSTRAINT',
'CK_Tbl_CitaServicio_Vehiculo'
```

end

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_CitaServicio', 'CONSTRAINT', 'UQ_Tbl_CitaServicio_IdCita'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Asegura que el id de la cita
sea único', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_CitaServicio', 'CONSTRAINT',
'UQ_Tbl_CitaServicio_IdCita'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Asegura que el id de la cita
sea único', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_CitaServicio', 'CONSTRAINT',
'UQ_Tbl_CitaServicio_IdCita'
end
```

GO

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'table',
'Tbl_Cliente', NULL, NULL))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Contiene la información de cada
cliente de la empresa, tanto particulares como empresas. Guarda datos de contacto,
dirección, ciudad, estado y tipo de cliente, además de la fecha de registro. Permite dar
seguimiento a compras y servicios para ofrecer atención personalizada.', 'SCHEMA', 'dbo',
'table', 'Tbl_Cliente'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Contiene la información de cada
cliente de la empresa, tanto particulares como empresas. Guarda datos de contacto,
dirección, ciudad, estado y tipo de cliente, además de la fecha de registro. Permite dar
seguimiento a compras y servicios para ofrecer atención personalizada.', 'SCHEMA', 'dbo',
'table', 'Tbl_Cliente'
end
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el identificador
numérico único de cada cliente registrado en la base de datos. No puede repetirse, debe ser
un número entero positivo y no puede quedar vacío. Este campo sirve como clave primaria
para identificar cada registro de cliente.', 'Schema', [dbo], 'table', [Tbl_Cliente], 'column',
[id_cliente]
GO
```


EXEC sp_addextendedproperty 'MS_Description', 'Aquí se guarda el correo electrónico principal del cliente. Debe ser único, no puede repetirse entre clientes y debe tener un formato válido de correo electrónico (usuario@dominio). No puede iniciar ni terminar con espacios en blanco.', 'Schema', [dbo], 'table', [Tbl_Cliente], 'column', [email]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el nombre completo del cliente. Debe contener caracteres alfabéticos, no puede quedar vacío ni comenzar con un carácter especial. Puede incluir espacios o guiones después de la primera letra.', 'Schema', [dbo], 'table', [Tbl_Cliente], 'column', [nombre]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el/los apellidos del cliente. Debe contener caracteres alfabéticos, no puede quedar vacío ni comenzar con un carácter especial. Puede incluir espacios o guiones después de la primera letra.', 'Schema', [dbo], 'table', [Tbl_Cliente], 'column', [apellido]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el número telefónico de contacto del cliente. Puede ser nulo. Si se ingresa, solo puede contener dígitos, espacios, signos +, - o paréntesis, y debe iniciar con dígito o +.', 'Schema', [dbo], 'table', [Tbl_Cliente], 'column', [telefono]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena la dirección física del cliente (calle, número, referencia). Es obligatoria, no puede quedar vacía y tiene un máximo de 255 caracteres.', 'Schema', [dbo], 'table', [Tbl_Cliente], 'column', [direccion]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena la ciudad de residencia del cliente. Es obligatoria y no puede quedar vacía ni iniciar con caracteres especiales.', 'Schema', [dbo], 'table', [Tbl_Cliente], 'column', [ciudad]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el estado, provincia o departamento donde reside el cliente. Es obligatorio y no puede quedar vacío.', 'Schema', [dbo], 'table', [Tbl_Cliente], 'column', [estado_cliente]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el código postal del cliente. Es opcional; si se ingresa, debe contener solo caracteres alfanuméricos, espacios o guiones y tener máximo 10 caracteres.', 'Schema', [dbo], 'table', [Tbl_Cliente], 'column', [codigo_postal]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena la fecha en que se registró el cliente en el sistema. Es obligatoria, no puede quedar vacía y no puede ser mayor a la fecha actual.', 'Schema', [dbo], 'table', [Tbl_Cliente], 'column', [fecha_registro]

GO

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el tipo de cliente según su naturaleza. Solo se permiten valores predefinidos como particular o empresa. Es obligatorio y no puede quedar vacío.', 'Schema', [dbo], 'table', [Tbl_Cliente], 'column', [tipo_cliente]
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Cliente', 'CONSTRAINT', 'PK_Cliente'))
```

```
begin
```

```
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Aquí se almacena el identificador numérico único de cada cliente registrado en la base de datos. No puede repetirse, debe ser un número entero positivo y no puede quedar vacío. Este campo sirve como clave primaria para identificar de manera inequívoca a cada cliente, ya sea persona o empresa.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Cliente', 'CONSTRAINT', 'PK_Cliente'
```

```
end
```

```
else
```

```
begin
```

```
EXEC sys.sp_addextendedproperty 'MS_Description', 'Aquí se almacena el identificador numérico único de cada cliente registrado en la base de datos. No puede repetirse, debe ser un número entero positivo y no puede quedar vacío. Este campo sirve como clave primaria para identificar de manera inequívoca a cada cliente, ya sea persona o empresa.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Cliente', 'CONSTRAINT', 'PK_Cliente'
```

```
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Cliente', 'CONSTRAINT', 'CK_Cliente_IdCliente'))
```

```
begin
```

```
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Asegura que el identificador sea mayor que 0.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Cliente', 'CONSTRAINT', 'CK_Cliente_IdCliente'
```

```
end
```

```
else
```

```
begin
```

```
EXEC sys.sp_addextendedproperty 'MS_Description', 'Asegura que el identificador sea mayor que 0.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Cliente', 'CONSTRAINT', 'CK_Cliente_IdCliente'
```

```
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Cliente', 'CONSTRAINT', 'UQ_Cliente_Email'))
```

```
begin
```

```

EXEC sys.sp_updateextendedproperty 'MS_Description', 'Garantiza que ningún
cliente registrado comparta la misma dirección de correo electrónico.', 'SCHEMA', 'dbo',
'TABLE', 'Tbl_Cliente', 'CONSTRAINT', 'UQ_Cliente_Email'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Garantiza que ningún
cliente registrado comparta la misma dirección de correo electrónico.', 'SCHEMA', 'dbo',
'TABLE', 'Tbl_Cliente', 'CONSTRAINT', 'UQ_Cliente_Email'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Cliente', 'CONSTRAINT', 'CK_Cliente_Email_Format'))
begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Valida que el correo
electrónico tenga formato correcto, contenga @ y un dominio válido, y no incluya espacios.',
'SCHEMA', 'dbo', 'TABLE', 'Tbl_Cliente', 'CONSTRAINT', 'CK_Cliente_Email_Format'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Valida que el correo
electrónico tenga formato correcto, contenga @ y un dominio válido, y no incluya espacios.',
'SCHEMA', 'dbo', 'TABLE', 'Tbl_Cliente', 'CONSTRAINT', 'CK_Cliente_Email_Format'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Cliente', 'CONSTRAINT', 'CK_Cliente_Nombre_IniciaLetra'))
begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Obliga a que el nombre
empiece con una letra.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Cliente', 'CONSTRAINT',
'CK_Cliente_Nombre_IniciaLetra'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Obliga a que el nombre
empiece con una letra.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Cliente', 'CONSTRAINT',
'CK_Cliente_Nombre_IniciaLetra'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Cliente', 'CONSTRAINT', 'CK_Cliente_Apellido_IniciaLetra'))
begin

```

```

EXEC sys.sp_updateextendedproperty 'MS_Description', 'Obliga a que el apellido
empiece con una letra.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Cliente', 'CONSTRAINT',
'CK_Cliente_Apellido_IniciaLetra'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Obliga a que el apellido
empiece con una letra.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Cliente', 'CONSTRAINT',
'CK_Cliente_Apellido_IniciaLetra'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Cliente', 'CONSTRAINT', 'CK_Cliente_Telefono_Format'))
begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Asegura que el teléfono
solo tenga caracteres numéricos y símbolos válidos (+, -, (), /) y longitud máxima de 20.',
'SCHEMA', 'dbo', 'TABLE', 'Tbl_Cliente', 'CONSTRAINT', 'CK_Cliente_Telefono_Format'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Asegura que el teléfono
solo tenga caracteres numéricos y símbolos válidos (+, -, (), /) y longitud máxima de 20.',
'SCHEMA', 'dbo', 'TABLE', 'Tbl_Cliente', 'CONSTRAINT', 'CK_Cliente_Telefono_Format'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Cliente', 'CONSTRAINT', 'CK_Cliente_CodigoPostal_Formato'))
begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Restringe el código postal
a caracteres alfanuméricos y longitud máxima de 10.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Cliente', 'CONSTRAINT', 'CK_Cliente_CodigoPostal_Formato'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Restringe el código postal
a caracteres alfanuméricos y longitud máxima de 10.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Cliente',
'CONSTRAINT', 'CK_Cliente_CodigoPostal_Formato'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Cliente', 'CONSTRAINT', 'CK_Cliente_FechaRegistro'))
begin

```

```

EXEC sys.sp_updateextendedproperty 'MS_Description', 'Garantiza que la fecha de
registro no sea posterior a la fecha actual.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Cliente',
'CONSTRAINT', 'CK_Cliente_FechaRegistro'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Garantiza que la fecha de
registro no sea posterior a la fecha actual.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Cliente',
'CONSTRAINT', 'CK_Cliente_FechaRegistro'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Cliente', 'CONSTRAINT', 'CK_Cliente_TipoCliente'))
begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Restringe el tipo de cliente
a "particular" o "empresa".', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Cliente', 'CONSTRAINT',
'CK_Cliente_TipoCliente'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Restringe el tipo de cliente a
"particular" o "empresa".', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Cliente', 'CONSTRAINT',
'CK_Cliente_TipoCliente'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Cliente', 'CONSTRAINT', 'UQ_Cliente_IdCliente'))
begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Evita duplicados en el
identificador único de clientes.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Cliente', 'CONSTRAINT',
'UQ_Cliente_IdCliente'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Evita duplicados en el
identificador único de clientes.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Cliente', 'CONSTRAINT',
'UQ_Cliente_IdCliente'
end

```

GO

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'table',
'Tbl_DetalleOrdenCompra', NULL, NULL))
```

```
begin
```

```
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Especifica los repuestos
solicitados en cada orden de compra. Indica cantidad, precio unitario y subtotal de cada
ítem. Se vincula tanto a la orden de compra como al repuesto para llevar un control
detallado del pedido.', 'SCHEMA', 'dbo', 'table', 'Tbl_DetalleOrdenCompra'
```

```
end
```

```
else
```

```
begin
```

```
EXEC sys.sp_addextendedproperty 'MS_Description', 'Especifica los repuestos solicitados
en cada orden de compra. Indica cantidad, precio unitario y subtotal de cada ítem. Se
vincula tanto a la orden de compra como al repuesto para llevar un control detallado del
pedido.', 'SCHEMA', 'dbo', 'table', 'Tbl_DetalleOrdenCompra'
```

```
end
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Representa el identificador único de cada
línea de detalle de una orden de compra. Es un número entero positivo, obligatorio y único.
Permite diferenciar cada producto o repuesto dentro de una orden de compra.', 'Schema',
[dbo], 'table', [Tbl_DetalleOrdenCompra], 'column', [id_detalle_orden_compra]
```

```
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se guarda el identificador de la
orden de compra a la que pertenece este detalle. Es un dato obligatorio que no puede
quedar vacío y funciona como clave foránea para asegurar que cada línea de detalle esté
vinculada a una orden de compra existente en la base de datos.', 'Schema', [dbo], 'table',
[Tbl_DetalleOrdenCompra], 'column', [id_orden_compra]
```

```
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el identificador del
repuesto solicitado en la orden de compra. Es obligatorio, no puede quedar vacío y funciona
como clave foránea para garantizar que solo se registren repuestos existentes en la base de
datos.', 'Schema', [dbo], 'table', [Tbl_DetalleOrdenCompra], 'column', [id_repuesto]
```

```
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se guarda la cantidad del repuesto
solicitado en esta línea de detalle. Es un campo obligatorio, no puede quedar vacío y debe
ser un número entero mayor que cero, ya que representa la cantidad a comprar.', 'Schema',
[dbo], 'table', [Tbl_DetalleOrdenCompra], 'column', [cantidad]
```

```
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el precio unitario
acordado para cada unidad del repuesto en esta línea de detalle. Es obligatorio, no puede
quedar vacío y debe ser un número decimal mayor o igual a cero.', 'Schema', [dbo], 'table',
[Tbl_DetalleOrdenCompra], 'column', [precio_unitario]
```

```
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el importe total de la
línea de detalle calculado como la multiplicación entre la cantidad solicitada y el precio
unitario de ese repuesto; no puede quedar vacío y debe ser mayor o igual a cero. ',
'Schema', [dbo], 'table', [Tbl_DetalleOrdenCompra], 'column', [subtotal]
GO
```

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_DetalleOrdenCompra', 'CONSTRAINT',
'FK_Tbl_DetalleOrdenCompra_Tbl_OrdenCompra'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Vincula cada detalle de
orden de compra con su orden de compra principal. Garantiza que ningún detalle exista sin
estar asociado a una orden válida.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_DetalleOrdenCompra',
'CONSTRAINT', 'FK_Tbl_DetalleOrdenCompra_Tbl_OrdenCompra'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Vincula cada detalle de
orden de compra con su orden de compra principal. Garantiza que ningún detalle exista sin
estar asociado a una orden válida.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_DetalleOrdenCompra',
'CONSTRAINT', 'FK_Tbl_DetalleOrdenCompra_Tbl_OrdenCompra'
end
```

```
GO
```

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_DetalleOrdenCompra', 'CONSTRAINT', 'FK_Tbl_DetalleOrdenCompra_Tbl_Repuesto'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Relaciona el detalle de
orden con el repuesto solicitado. Asegura que únicamente se puedan registrar repuestos
existentes en la tabla de repuestos.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_DetalleOrdenCompra',
'CONSTRAINT', 'FK_Tbl_DetalleOrdenCompra_Tbl_Repuesto'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Relaciona el detalle de
orden con el repuesto solicitado. Asegura que únicamente se puedan registrar repuestos
existentes en la tabla de repuestos.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_DetalleOrdenCompra',
'CONSTRAINT', 'FK_Tbl_DetalleOrdenCompra_Tbl_Repuesto'
end
```

```
GO
```

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_DetalleOrdenCompra', 'INDEX', 'IXFK_Tbl_DetalleOrdenCompra_Tbl_OrdenCompra'))
begin
```

```
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Mejora el rendimiento de
las consultas que unen los detalles con sus órdenes de compra.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_DetalleOrdenCompra', 'INDEX', 'IXFK_Tbl_DetalleOrdenCompra_Tbl_OrdenCompra'
```

```
end
```

```
else
```

```
begin
```

```
EXEC sys.sp_addextendedproperty 'MS_Description', 'Mejora el rendimiento de las
consultas que unen los detalles con sus órdenes de compra.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_DetalleOrdenCompra', 'INDEX', 'IXFK_Tbl_DetalleOrdenCompra_Tbl_OrdenCompra'
```

```
end
```

```
GO
```

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_DetalleOrdenCompra', 'INDEX', 'IXFK_Tbl_DetalleOrdenCompra_Tbl_Repuesto'))
```

```
begin
```

```
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Facilita búsquedas y
uniones entre los repuestos y el detalle de órdenes de compra.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_DetalleOrdenCompra', 'INDEX', 'IXFK_Tbl_DetalleOrdenCompra_Tbl_Repuesto'
```

```
end
```

```
else
```

```
begin
```

```
EXEC sys.sp_addextendedproperty 'MS_Description', 'Facilita búsquedas y
uniones entre los repuestos y el detalle de órdenes de compra.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_DetalleOrdenCompra', 'INDEX', 'IXFK_Tbl_DetalleOrdenCompra_Tbl_Repuesto'
```

```
end
```

```
GO
```

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_DetalleOrdenCompra', 'CONSTRAINT', 'PK_Tbl_DetalleOrdenCompra'))
```

```
begin
```

```
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Representa el identificador
único de cada línea de detalle de una orden de compra. Es un número entero positivo,
obligatorio y único. Permite diferenciar cada producto o repuesto dentro de una orden de
compra.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_DetalleOrdenCompra', 'CONSTRAINT',
'PK_Tbl_DetalleOrdenCompra'
```

```
end
```

```
else
```

```
begin
```

```
EXEC sys.sp_addextendedproperty 'MS_Description', 'Representa el identificador
único de cada línea de detalle de una orden de compra. Es un número entero positivo,
obligatorio y único. Permite diferenciar cada producto o repuesto dentro de una orden de
compra.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_DetalleOrdenCompra', 'CONSTRAINT',
```

```
'PK_Tbl_DetalleOrdenCompra'
```

```
end
```

```
GO
```



```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_DetalleOrdenCompra', 'CONSTRAINT', 'CK_DetalleOrdenCompra_IdDetalle'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Verifica que el ID sea
mayor que 0.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_DetalleOrdenCompra', 'CONSTRAINT',
'CK_DetalleOrdenCompra_IdDetalle'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Verifica que el ID sea mayor
que 0.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_DetalleOrdenCompra', 'CONSTRAINT',
'CK_DetalleOrdenCompra_IdDetalle'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_DetalleOrdenCompra', 'CONSTRAINT', 'CK_DetalleOrdenCompra_Cantidad'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Obliga a que la cantidad
pedida sea positiva.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_DetalleOrdenCompra', 'CONSTRAINT',
'CK_DetalleOrdenCompra_Cantidad'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Obliga a que la cantidad
pedida sea positiva.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_DetalleOrdenCompra', 'CONSTRAINT',
'CK_DetalleOrdenCompra_Cantidad'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_DetalleOrdenCompra', 'CONSTRAINT', 'CK_DetalleOrdenCompra_PrecioUnitario'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Asegura que el precio
unitario no sea negativo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_DetalleOrdenCompra',
'CONSTRAINT', 'CK_DetalleOrdenCompra_PrecioUnitario'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Asegura que el precio
unitario no sea negativo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_DetalleOrdenCompra',
'CONSTRAINT', 'CK_DetalleOrdenCompra_PrecioUnitario'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_DetalleOrdenCompra', 'CONSTRAINT',
'UQ_DetalleOrdenCompra_IdDetalleOrdenCompra'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Garantiza que cada detalle
de orden sea único.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_DetalleOrdenCompra', 'CONSTRAINT',
'UQ_DetalleOrdenCompra_IdDetalleOrdenCompra'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Garantiza que cada detalle
de orden sea único.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_DetalleOrdenCompra', 'CONSTRAINT',
'UQ_DetalleOrdenCompra_IdDetalleOrdenCompra'
end

GO

```

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_DetalleOrdenCompra', 'CONSTRAINT', 'CK_DetalleOrdenCompra_Subtotal'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Comprueba que el subtotal
sea coherente con cantidad × precio unitario y no sea negativo.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_DetalleOrdenCompra', 'CONSTRAINT', 'CK_DetalleOrdenCompra_Subtotal'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Comprueba que el subtotal
sea coherente con cantidad × precio unitario y no sea negativo.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_DetalleOrdenCompra', 'CONSTRAINT', 'CK_DetalleOrdenCompra_Subtotal'
end

GO

```

```

GO
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'table',
'Tbl_Empleado', NULL, NULL))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Almacena los datos del personal
de la casa automotriz: nombre, apellido, cargo (vendedor, mecánico, etc.), salario, dirección,
fecha de contratación y estado laboral. Sirve para controlar quién realiza ventas, servicios,
facturas u órdenes.', 'SCHEMA', 'dbo', 'table', 'Tbl_Empleado'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Almacena los datos del personal de
la casa automotriz: nombre, apellido, cargo (vendedor, mecánico, etc.), salario, dirección,

```

fecha de contratación y estado laboral. Sirve para controlar quién realiza ventas, servicios, facturas u órdenes.', 'SCHEMA', 'dbo', 'table', 'Tbl_Empleado'
end

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el identificador numérico único de cada empleado registrado en la base de datos. No puede repetirse, debe ser un número entero positivo y no puede quedar vacío. Este campo sirve como clave primaria para identificar cada registro de empleado.', 'Schema', [dbo], 'table', [Tbl_Empleado], 'column', [id_empleado]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se guarda el correo electrónico del empleado. Debe ser único, no puede repetirse entre empleados y debe tener un formato válido de correo electrónico (usuario@dominio). No puede iniciar ni terminar con espacios en blanco.', 'Schema', [dbo], 'table', [Tbl_Empleado], 'column', [email]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el nombre del empleado. Debe contener caracteres alfabéticos, no puede quedar vacío ni comenzar con un carácter especial. Puede incluir espacios o guiones después de la primera letra.', 'Schema', [dbo], 'table', [Tbl_Empleado], 'column', [nombre]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el/los apellidos del empleado. Debe contener caracteres alfabéticos, no puede quedar vacío ni comenzar con un carácter especial. Puede incluir espacios o guiones después de la primera letra.', 'Schema', [dbo], 'table', [Tbl_Empleado], 'column', [apellido]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el cargo o puesto que ocupa el empleado dentro de la empresa. Solo se permiten valores predefinidos como vendedor, mecánico, etc. Es obligatorio y no puede quedar vacío.', 'Schema', [dbo], 'table', [Tbl_Empleado], 'column', [cargo]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el número telefónico de contacto del empleado. Puede ser nulo. Si se ingresa, solo puede contener dígitos, espacios, signos +, - o paréntesis, y debe iniciar con dígito o +.', 'Schema', [dbo], 'table', [Tbl_Empleado], 'column', [telefono]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena la fecha en que fue contratado el empleado. Es obligatoria, no puede quedar vacía y no puede ser mayor a la fecha actual.', 'Schema', [dbo], 'table', [Tbl_Empleado], 'column', [fecha_contratacion]
GO

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el salario mensual del
empleado. Debe ser un número decimal mayor o igual a cero y no puede quedar vacío.',
'Schema', [dbo], 'table', [Tbl_Empleado], 'column', [salario]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena la dirección física del
empleado. Es obligatoria, no puede quedar vacía y tiene un máximo de 255 caracteres.',
'Schema', [dbo], 'table', [Tbl_Empleado], 'column', [direccion]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el estado laboral
actual del empleado. Solo se permiten valores predefinidos como activo o inactivo. Es
obligatorio y no puede quedar vacío.', 'Schema', [dbo], 'table', [Tbl_Empleado], 'column',
[estado_empleado]
GO
```

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Empleado', 'CONSTRAINT', 'PK_Empleado'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Representa el identificador
único de cada empleado de la empresa. Es un valor numérico entero, no puede estar vacío
ni repetirse. Permite diferenciar a cada empleado de forma precisa para tareas de gestión,
control de nómina y asignación de responsabilidades.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Empleado', 'CONSTRAINT', 'PK_Empleado'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Representa el identificador
único de cada empleado de la empresa. Es un valor numérico entero, no puede estar vacío
ni repetirse. Permite diferenciar a cada empleado de forma precisa para tareas de gestión,
control de nómina y asignación de responsabilidades.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Empleado', 'CONSTRAINT', 'PK_Empleado'
end

GO
```

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Empleado', 'CONSTRAINT', 'CK_Empleado_IdEmpleado'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Asegura que el ID sea
positivo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Empleado', 'CONSTRAINT',
'CK_Empleado_IdEmpleado'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Asegura que el ID sea
positivo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Empleado', 'CONSTRAINT',
'CK_Empleado_IdEmpleado'
```

end

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Empleado', 'CONSTRAINT', 'UQ_Empleado_Email'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'No permite correos
electrónicos duplicados en empleados.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Empleado',
'CONSTRAINT', 'UQ_Empleado_Email'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'No permite correos
electrónicos duplicados en empleados.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Empleado',
'CONSTRAINT', 'UQ_Empleado_Email'
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Empleado', 'CONSTRAINT', 'CK_Empleado_Email_Format'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Valida que el correo tenga
formato correcto y sin espacios.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Empleado',
'CONSTRAINT', 'CK_Empleado_Email_Format'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Valida que el correo tenga
formato correcto y sin espacios.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Empleado',
'CONSTRAINT', 'CK_Empleado_Email_Format'
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Empleado', 'CONSTRAINT', 'CK_Empleado_Nombre_IniciaLetra'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Obliga a que el nombre
inicie con letra.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Empleado', 'CONSTRAINT',
'CK_Empleado_Nombre_IniciaLetra'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Obliga a que el nombre
inicie con letra.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Empleado', 'CONSTRAINT',
'CK_Empleado_Nombre_IniciaLetra'
```

end

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Empleado', 'CONSTRAINT', 'CK_Empleado_Apellido_IniciaLetra'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Obliga a que el apellido
inicie con letra.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Empleado', 'CONSTRAINT',
'CK_Empleado_Apellido_IniciaLetra'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Obliga a que el apellido
inicie con letra.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Empleado', 'CONSTRAINT',
'CK_Empleado_Apellido_IniciaLetra'
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Empleado', 'CONSTRAINT', 'CK_Empleado_Cargo_Valores'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Restringe los cargos a
“vendedor”, “mecánico”, “gerente” o “administrativo”.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Empleado', 'CONSTRAINT', 'CK_Empleado_Cargo_Valores'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Restringe los cargos a
“vendedor”, “mecánico”, “gerente” o “administrativo”.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Empleado', 'CONSTRAINT', 'CK_Empleado_Cargo_Valores'
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Empleado', 'CONSTRAINT', 'CK_Empleado_Telefono_Format'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Valida que el teléfono
tenga solo caracteres permitidos y longitud = 20.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Empleado', 'CONSTRAINT', 'CK_Empleado_Telefono_Format'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Valida que el teléfono tenga
solo caracteres permitidos y longitud = 20.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Empleado',
'CONSTRAINT', 'CK_Empleado_Telefono_Format'
```

end

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Empleado', 'CONSTRAINT', 'CK_Empleado_FechaContratacion'))
```

```
begin
```

```
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Garantiza que la fecha de
contratación no sea futura.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Empleado', 'CONSTRAINT',
'CK_Empleado_FechaContratacion'
```

```
end
```

```
else
```

```
begin
```

```
EXEC sys.sp_addextendedproperty 'MS_Description', 'Garantiza que la fecha de
contratación no sea futura.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Empleado', 'CONSTRAINT',
'CK_Empleado_FechaContratacion'
```

```
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Empleado', 'CONSTRAINT', 'CK_Empleado_Salario'))
```

```
begin
```

```
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Asegura que el salario no
sea negativo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Empleado', 'CONSTRAINT',
'CK_Empleado_Salario'
```

```
end
```

```
else
```

```
begin
```

```
EXEC sys.sp_addextendedproperty 'MS_Description', 'Asegura que el salario no
sea negativo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Empleado', 'CONSTRAINT',
'CK_Empleado_Salario'
```

```
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Empleado', 'CONSTRAINT', 'CK_Empleado_EstadoValores'))
```

```
begin
```

```
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Restringe estado del
empleado a "activo" o "inactivo".', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Empleado',
'CONSTRAINT', 'CK_Empleado_EstadoValores'
```

```
end
```

```
else
```

```
begin
```

```
EXEC sys.sp_addextendedproperty 'MS_Description', 'Restringe estado del
empleado a "activo" o "inactivo".', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Empleado',
'CONSTRAINT', 'CK_Empleado_EstadoValores'
```

end

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Empleado', 'CONSTRAINT', 'UQ_Empleado_IdEmpleado'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Evita IDs de empleados
repetidos.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Empleado', 'CONSTRAINT',
'UQ_Empleado_IdEmpleado'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Evita IDs de empleados
repetidos.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Empleado', 'CONSTRAINT',
'UQ_Empleado_IdEmpleado'
end
```

GO

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'table',
'Tbl_Factura', NULL, NULL))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Contiene los documentos de
facturación emitidos por ventas. Se asocia a la venta y al empleado que la generó. Incluye
fecha de emisión, monto total, estado y método de pago, número de recibo, observaciones y
fecha de vencimiento para controlar cobros.', 'SCHEMA', 'dbo', 'table', 'Tbl_Factura'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Contiene los documentos de
facturación emitidos por ventas. Se asocia a la venta y al empleado que la generó. Incluye
fecha de emisión, monto total, estado y método de pago, número de recibo, observaciones y
fecha de vencimiento para controlar cobros.', 'SCHEMA', 'dbo', 'table', 'Tbl_Factura'
end
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Identificador único de cada factura
emitida por la empresa. Es un número entero positivo, obligatorio y exclusivo. Garantiza la
integridad en el control de facturación y permite asociar la factura con su respectiva venta y
empleado.', 'Schema', [dbo], 'table', [Tbl_Factura], 'column', [id_factura]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se guarda el identificador de la
venta asociada a la factura. Es obligatorio, no puede quedar vacío y funciona como clave
foránea para asegurar que toda factura esté vinculada a una venta existente en la base de
datos.', 'Schema', [dbo], 'table', [Tbl_Factura], 'column', [id_venta]
```


GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena la fecha en la que se emitió la factura. Es obligatoria, no puede quedar vacía y no puede ser posterior a la fecha actual para garantizar que refleje un evento real.', 'Schema', [dbo], 'table', [Tbl_Factura], 'column', [fecha_emision]

GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se guarda el monto total facturado. Es obligatorio, no puede quedar vacío y debe ser mayor o igual a cero, ya que representa la suma de los conceptos incluidos en la factura.', 'Schema', [dbo], 'table', [Tbl_Factura], 'column', [monto_total]

GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el estado actual del pago de la factura. Es obligatorio y solo puede tomar uno de los valores predefinidos ("pagado", "pendiente"), permitiendo controlar el estado financiero.', 'Schema', [dbo], 'table', [Tbl_Factura], 'column', [estado_pago]

GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se guarda el método de pago utilizado para la factura. Es obligatorio únicamente cuando el estado de pago es "pagado"; si la factura está pendiente este campo puede quedar vacío. Solo puede tomar valores predefinidos como "efectivo" o "tarjeta".', 'Schema', [dbo], 'table', [Tbl_Factura], 'column', [metodo_pago]

GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el número único de recibo asociado a la factura. Es obligatorio únicamente cuando el estado de pago es "pagado"; si la factura está pendiente este campo puede quedar vacío. No puede repetirse para facturas pagadas.', 'Schema', [dbo], 'table', [Tbl_Factura], 'column', [numero_recibo]

GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena la fecha límite para el pago de la factura. Es obligatoria, no puede quedar vacía y debe ser igual o posterior a la fecha de emisión para asegurar coherencia en los plazos.', 'Schema', [dbo], 'table', [Tbl_Factura], 'column', [fecha_vencimiento]

GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se guarda el identificador del empleado que generó la factura. Es obligatorio, no puede quedar vacío y funciona como clave foránea para asegurar que cada factura esté asociada a un empleado registrado en el sistema.', 'Schema', [dbo], 'table', [Tbl_Factura], 'column', [id_empleado]

GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se pueden almacenar comentarios adicionales relacionados con la factura. Es un campo opcional y puede quedar vacío.', 'Schema', [dbo], 'table', [Tbl_Factura], 'column', [observaciones]

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Factura', 'CONSTRAINT', 'FK_Tbl_Factura_Tbl_Empleado'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Indica qué empleado
emitió la factura. Garantiza trazabilidad de quién registró la operación.', 'SCHEMA', 'dbo',
'TABLE', 'Tbl_Factura', 'CONSTRAINT', 'FK_Tbl_Factura_Tbl_Empleado'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Indica qué empleado emitió
la factura. Garantiza trazabilidad de quién registró la operación.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Factura', 'CONSTRAINT', 'FK_Tbl_Factura_Tbl_Empleado'
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Factura', 'CONSTRAINT', 'FK_Tbl_Factura_Tbl_Venta'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Relaciona la factura con la
venta a la que corresponde. Impide que exista una factura sin estar vinculada a una venta
válida.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Factura', 'CONSTRAINT',
'FK_Tbl_Factura_Tbl_Venta'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Relaciona la factura con la
venta a la que corresponde. Impide que exista una factura sin estar vinculada a una venta
válida.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Factura', 'CONSTRAINT',
'FK_Tbl_Factura_Tbl_Venta'
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Factura', 'INDEX', 'IXFK_Tbl_Factura_Tbl_Empleado'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Optimiza consultas que
relacionan facturas con el empleado responsable.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Factura',
'INDEX', 'IXFK_Tbl_Factura_Tbl_Empleado'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Optimiza consultas que
relacionan facturas con el empleado responsable.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Factura',
'INDEX', 'IXFK_Tbl_Factura_Tbl_Empleado'
```

end

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Factura', 'INDEX', 'IXFK_Tbl_Factura_Tbl_Venta'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Permite búsquedas
rápidas y joins entre facturas y sus ventas asociadas.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Factura', 'INDEX', 'IXFK_Tbl_Factura_Tbl_Venta'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Permite búsquedas rápidas
y joins entre facturas y sus ventas asociadas.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Factura',
'INDEX', 'IXFK_Tbl_Factura_Tbl_Venta'
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Factura', 'CONSTRAINT', 'PK_Tbl_Factura'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Código único de cada
factura emitida por la empresa. Es un número entero positivo, obligatorio y exclusivo.
Garantiza la integridad en el control de facturación y permite asociar la factura con su
respectiva venta y empleado.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Factura', 'CONSTRAINT',
'PK_Tbl_Factura'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Código único de cada
factura emitida por la empresa. Es un número entero positivo, obligatorio y exclusivo.
Garantiza la integridad en el control de facturación y permite asociar la factura con su
respectiva venta y empleado.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Factura', 'CONSTRAINT',
'PK_Tbl_Factura'
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Factura', 'CONSTRAINT', 'UQ_Factura_IdFactura'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Evita IDs de factura
duplicados.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Factura', 'CONSTRAINT',
'UQ_Factura_IdFactura'
end
else
```

```

begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Evita IDs de factura
duplicados.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Factura', 'CONSTRAINT',
'UQ_Factura_IdFactura'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Factura', 'CONSTRAINT', 'CK_Factura_IdFactura'))

```

```

begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Verifica que el ID de
factura sea positivo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Factura', 'CONSTRAINT',
'CK_Factura_IdFactura'
end

```

else

```

begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Verifica que el ID de factura
sea positivo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Factura', 'CONSTRAINT',
'CK_Factura_IdFactura'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Factura', 'CONSTRAINT', 'CK_Factura_FechaEmision'))

```

```

begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Asegura que la fecha de
emisión no supere la fecha actual.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Factura',
'CONSTRAINT', 'CK_Factura_FechaEmision'
end

```

else

```

begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Asegura que la fecha de
emisión no supere la fecha actual.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Factura',
'CONSTRAINT', 'CK_Factura_FechaEmision'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Factura', 'CONSTRAINT', 'CK_Factura_MontoTotal'))

```

```

begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'No permite montos
negativos.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Factura', 'CONSTRAINT',
'CK_Factura_MontoTotal'
end

```

else

```

begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'No permite montos
negativos.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Factura', 'CONSTRAINT',
'CK_Factura_MontoTotal'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Factura', 'CONSTRAINT', 'CK_Factura_EstadoPago'))

```

```

    begin
        EXEC sys.sp_updateextendedproperty 'MS_Description', 'Solo admite estados
“pendiente” o “pagado”.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Factura', 'CONSTRAINT',
'CK_Factura_EstadoPago'
    end

```

end

else

```

    begin
        EXEC sys.sp_addextendedproperty 'MS_Description', 'Solo admite estados
“pendiente” o “pagado”.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Factura', 'CONSTRAINT',
'CK_Factura_EstadoPago'
    end

```

end

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Factura', 'CONSTRAINT', 'CK_Factura_MetodoPago'))

```

```

    begin
        EXEC sys.sp_updateextendedproperty 'MS_Description', 'Valida que el método sea
nulo (si pendiente) o válido (“efectivo”, “tarjeta”).', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Factura',
'CONSTRAINT', 'CK_Factura_MetodoPago'
    end

```

end

else

```

    begin
        EXEC sys.sp_addextendedproperty 'MS_Description', 'Valida que el método sea
nulo (si pendiente) o válido (“efectivo”, “tarjeta”).', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Factura',
'CONSTRAINT', 'CK_Factura_MetodoPago'
    end

```

end

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Factura', 'CONSTRAINT', 'UQ_Factura_NumeroRecibo'))

```

```

    begin
        EXEC sys.sp_updateextendedproperty 'MS_Description', 'Garantiza que cada recibo
sea único.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Factura', 'CONSTRAINT',
'UQ_Factura_NumeroRecibo'
    end

```

end

else

```

begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Garantiza que cada recibo
sea único.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Factura', 'CONSTRAINT',
'UQ_Factura_NumeroRecibo'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Factura', 'CONSTRAINT', 'CK_Factura_FechaVencimiento'))

```

```

begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Requiere que la fecha de
vencimiento sea posterior o igual a la de emisión.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Factura',
'CONSTRAINT', 'CK_Factura_FechaVencimiento'
end

```

else

```

begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Requiere que la fecha de
vencimiento sea posterior o igual a la de emisión.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Factura',
'CONSTRAINT', 'CK_Factura_FechaVencimiento'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Factura', 'CONSTRAINT', 'CK_Factura_PagoDatos'))

```

```

begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Verifica consistencia: si
está pendiente, no puede tener método/recibo; si está pagada, sí debe tenerlos.',
'SCHEMA', 'dbo', 'TABLE', 'Tbl_Factura', 'CONSTRAINT', 'CK_Factura_PagoDatos'
end

```

else

```

begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Verifica consistencia: si está
pendiente, no puede tener método/recibo; si está pagada, sí debe tenerlos.', 'SCHEMA',
'dbo', 'TABLE', 'Tbl_Factura', 'CONSTRAINT', 'CK_Factura_PagoDatos'
end

```

GO

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'table',
'Tbl_OrdenCompra', NULL, NULL))

```

begin

```

    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Registra las órdenes de compra
que la empresa realiza a proveedores. Contiene la fecha, estado (pendiente o recibida),
monto total, método de pago, forma de envío, observaciones y el empleado que la gestionó.

```

```
Se vincula al proveedor para controlar abastecimiento.', 'SCHEMA', 'dbo', 'table',
'Tbl_OrdenCompra'
end
else
begin
```

```
EXEC sys.sp_addextendedproperty 'MS_Description', 'Registra las órdenes de compra que
la empresa realiza a proveedores. Contiene la fecha, estado (pendiente o recibida), monto
total, método de pago, forma de envío, observaciones y el empleado que la gestionó. Se
vincula al proveedor para controlar abastecimiento.', 'SCHEMA', 'dbo', 'table',
'Tbl_OrdenCompra'
end
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Identificador único de cada orden de
compra realizada a proveedores. No se puede repetir, debe ser entero positivo y no nulo.
Permite rastrear las órdenes, asociarlas con proveedores y empleados responsables.',
'Schema', [dbo], 'table', [Tbl_OrdenCompra], 'column', [id_orden_compra]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se guarda el identificador del
proveedor al que se realiza la orden de compra. Es un dato obligatorio que no puede quedar
vacío y funciona como clave foránea para garantizar que toda orden esté vinculada a un
proveedor existente en la base de datos.', 'Schema', [dbo], 'table', [Tbl_OrdenCompra],
'column', [id_proveedor]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena la fecha en la que se
generó la orden de compra. Es un campo obligatorio que no puede quedar vacío ni ser
posterior a la fecha actual, ya que debe reflejar el momento real en que se hizo el pedido.',
'Schema', [dbo], 'table', [Tbl_OrdenCompra], 'column', [fecha_orden]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se guarda la fecha estimada o real
de entrega de la orden de compra. Es un campo opcional que puede quedar vacío mientras
no se tenga confirmación, pero si se ingresa debe ser igual o posterior a la fecha de orden
para mantener coherencia temporal.', 'Schema', [dbo], 'table', [Tbl_OrdenCompra], 'column',
[fecha_entrega]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el estado actual de la
orden de compra. Es un campo obligatorio que solo puede tomar los valores permitidos
("pendiente" o "recibida"), para garantizar un control uniforme del avance de las órdenes.',
'Schema', [dbo], 'table', [Tbl_OrdenCompra], 'column', [estado_orden]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se guarda el monto total de la orden
de compra. Es obligatorio, no puede quedar vacío y debe ser mayor o igual a cero, ya que
representa el total comprometido para esa orden.', 'Schema', [dbo], 'table',
[Tbl_OrdenCompra], 'column', [total_orden]
```

GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se pueden registrar comentarios adicionales relacionados con la orden de compra. Es un campo opcional y puede quedar vacío.', 'Schema', [dbo], 'table', [Tbl_OrdenCompra], 'column', [observaciones]

GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se guarda el identificador del empleado que gestionó la orden de compra. Es un campo obligatorio que no puede quedar vacío y funciona como clave foránea para asegurar que cada orden esté asociada a un empleado registrado en el sistema.', 'Schema', [dbo], 'table', [Tbl_OrdenCompra], 'column', [id_empleado]

GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el método de pago acordado para la orden de compra. Es un dato obligatorio y solo puede tomar los valores permitidos ("efectivo" o "transferencia"), para clasificar y auditar correctamente los pagos.', 'Schema', [dbo], 'table', [Tbl_OrdenCompra], 'column', [metodo_pago]

GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se guarda la forma de envío de la orden de compra. Es un dato obligatorio que solo puede tomar los valores permitidos ("terrestre", "aéreo" o "marítimo"), para reflejar el medio logístico utilizado.', 'Schema', [dbo], 'table', [Tbl_OrdenCompra], 'column', [forma_envio]

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_OrdenCompra', 'CONSTRAINT', 'FK_Tbl_OrdenCompra_Tbl_Empleado'))
```

```
begin
```

```
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Asocia la orden de compra
al empleado responsable de gestionarla. Permite seguimiento interno de quién realizó la
orden.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_OrdenCompra', 'CONSTRAINT',
'FK_Tbl_OrdenCompra_Tbl_Empleado'
```

```
end
```

```
else
```

```
begin
```

```
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Asocia la orden de compra
al empleado responsable de gestionarla. Permite seguimiento interno de quién realizó la
orden.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_OrdenCompra', 'CONSTRAINT',
'FK_Tbl_OrdenCompra_Tbl_Empleado'
```

```
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_OrdenCompra', 'CONSTRAINT', 'FK_Tbl_OrdenCompra_Tbl_Proveedor'))
```

```
begin
```



```

EXEC sys.sp_updateextendedproperty 'MS_Description', 'Vincula la orden con el
proveedor correspondiente. Garantiza que las órdenes siempre estén dirigidas a un
proveedor registrado.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_OrdenCompra', 'CONSTRAINT',
'FK_Tbl_OrdenCompra_Tbl_Proveedor'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Vincula la orden con el
proveedor correspondiente. Garantiza que las órdenes siempre estén dirigidas a un
proveedor registrado.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_OrdenCompra', 'CONSTRAINT',
'FK_Tbl_OrdenCompra_Tbl_Proveedor'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_OrdenCompra', 'INDEX', 'IXFK_Tbl_OrdenCompra_Tbl_Empleado'))
begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Acelera consultas que
muestran órdenes de compra realizadas por un empleado específico.', 'SCHEMA', 'dbo',
'TABLE', 'Tbl_OrdenCompra', 'INDEX', 'IXFK_Tbl_OrdenCompra_Tbl_Empleado'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Acelera consultas que
muestran órdenes de compra realizadas por un empleado específico.', 'SCHEMA', 'dbo',
'TABLE', 'Tbl_OrdenCompra', 'INDEX', 'IXFK_Tbl_OrdenCompra_Tbl_Empleado'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_OrdenCompra', 'INDEX', 'IXFK_Tbl_OrdenCompra_Tbl_Proveedor'))
begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Mejora el rendimiento de
consultas que relacionan órdenes con proveedores.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_OrdenCompra', 'INDEX', 'IXFK_Tbl_OrdenCompra_Tbl_Proveedor'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Mejora el rendimiento de
consultas que relacionan órdenes con proveedores.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_OrdenCompra', 'INDEX', 'IXFK_Tbl_OrdenCompra_Tbl_Proveedor'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_OrdenCompra', 'CONSTRAINT', 'PK_Tbl_OrdenCompra'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Identificador único de cada
orden de compra realizada a proveedores. No se puede repetir, debe ser entero positivo y
no nulo. Permite rastrear las órdenes, asociarlas con proveedores y empleados
responsables.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_OrdenCompra', 'CONSTRAINT',
'PK_Tbl_OrdenCompra'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Identificador único de cada
orden de compra realizada a proveedores. No se puede repetir, debe ser entero positivo y
no nulo. Permite rastrear las órdenes, asociarlas con proveedores y empleados
responsables.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_OrdenCompra', 'CONSTRAINT',
'PK_Tbl_OrdenCompra'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_OrdenCompra', 'CONSTRAINT', 'CK_OrdenCompra_IdOrdenCompra'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Asegura que el ID sea
positivo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_OrdenCompra', 'CONSTRAINT',
'CK_OrdenCompra_IdOrdenCompra'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Asegura que el ID sea
positivo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_OrdenCompra', 'CONSTRAINT',
'CK_OrdenCompra_IdOrdenCompra'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_OrdenCompra', 'CONSTRAINT', 'CK_OrdenCompra_FechaOrden'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'La fecha de orden no
puede ser futura.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_OrdenCompra', 'CONSTRAINT',
'CK_OrdenCompra_FechaOrden'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'La fecha de orden no puede
ser futura.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_OrdenCompra', 'CONSTRAINT',
'CK_OrdenCompra_FechaOrden'

```

end

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_OrdenCompra', 'CONSTRAINT', 'CK_OrdenCompra_FechaEntrega'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'La entrega debe ser nula o
posterior a la fecha de orden.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_OrdenCompra',
'CONSTRAINT', 'CK_OrdenCompra_FechaEntrega'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'La entrega debe ser nula o
posterior a la fecha de orden.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_OrdenCompra',
'CONSTRAINT', 'CK_OrdenCompra_FechaEntrega'
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_OrdenCompra', 'CONSTRAINT', 'CK_OrdenCompra_EstadoOrden'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Restringe a “pendiente” o
“recibida”.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_OrdenCompra', 'CONSTRAINT',
'CK_OrdenCompra_EstadoOrden'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Restringe a “pendiente” o
“recibida”.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_OrdenCompra', 'CONSTRAINT',
'CK_OrdenCompra_EstadoOrden'
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_OrdenCompra', 'CONSTRAINT', 'CK_OrdenCompra_TotalOrden'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'El total no puede ser
negativo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_OrdenCompra', 'CONSTRAINT',
'CK_OrdenCompra_TotalOrden'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'El total no puede ser
negativo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_OrdenCompra', 'CONSTRAINT',
'CK_OrdenCompra_TotalOrden'
```

end

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_OrdenCompra', 'CONSTRAINT', 'CK_OrdenCompra_MetodoPago'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Restringido a "efectivo" o
"transferencia".', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_OrdenCompra', 'CONSTRAINT',
'CK_OrdenCompra_MetodoPago'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Restringido a "efectivo" o
"transferencia".', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_OrdenCompra', 'CONSTRAINT',
'CK_OrdenCompra_MetodoPago'
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_OrdenCompra', 'CONSTRAINT', 'CK_OrdenCompra_FormaEnvio'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Solo admite "terrestre",
"aéreo" o "marítimo".', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_OrdenCompra', 'CONSTRAINT',
'CK_OrdenCompra_FormaEnvio'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Solo admite "terrestre",
"aéreo" o "marítimo".', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_OrdenCompra', 'CONSTRAINT',
'CK_OrdenCompra_FormaEnvio'
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_OrdenCompra', 'CONSTRAINT', 'UQ_OrdenCompra_IdOrdenCompra'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Evita duplicados en IDs de
orden.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_OrdenCompra', 'CONSTRAINT',
'UQ_OrdenCompra_IdOrdenCompra'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Evita duplicados en IDs de
orden.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_OrdenCompra', 'CONSTRAINT',
'UQ_OrdenCompra_IdOrdenCompra'
```

end

GO

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'table',
'Tbl_Proveedor', NULL, NULL))
```

```
begin
```

```
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Proveedor almacena la
información de las empresas o personas que suministran repuestos, materiales o servicios a
la casa automotriz. Esta tabla permite identificar de forma única a cada proveedor, disponer
de sus datos de contacto, ubicación y clasificación (local / internacional), y servir como
referencia en las compras, órdenes y repuestos. ', 'SCHEMA', 'dbo', 'table', 'Tbl_Proveedor'
end
```

```
else
```

```
begin
```

```
EXEC sys.sp_addextendedproperty 'MS_Description', 'Proveedor almacena la información
de las empresas o personas que suministran repuestos, materiales o servicios a la casa
automotriz. Esta tabla permite identificar de forma única a cada proveedor, disponer de sus
datos de contacto, ubicación y clasificación (local / internacional), y servir como referencia
en las compras, órdenes y repuestos. ', 'SCHEMA', 'dbo', 'table', 'Tbl_Proveedor'
end
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el identificador
numérico único de cada proveedor registrado en la base de datos. No puede repetirse, debe
ser un número entero positivo mayor que cero y no puede quedar vacío. Este campo sirve
como clave primaria para identificar de forma inequívoca cada proveedor.', 'Schema', [dbo],
'table', [Tbl_Proveedor], 'column', [id_proveedor]
```

GO

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se guarda el nombre oficial o razón
social del proveedor. Es un dato obligatorio que no puede quedar vacío ni contener
únicamente espacios, ya que se utiliza en reportes, búsquedas y facturación para reconocer
de forma clara a cada proveedor.', 'Schema', [dbo], 'table', [Tbl_Proveedor], 'column',
[nombre]
```

GO

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el número telefónico
de contacto del proveedor. Este campo es opcional, pero si se ingresa debe respetar el
formato habitual de números telefónicos, permitiendo dígitos, guiones, paréntesis y el signo
más. Sirve como un medio directo de comunicación con el proveedor.', 'Schema', [dbo],
'table', [Tbl_Proveedor], 'column', [telefono]
```

GO

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se guarda la dirección de correo
electrónico principal del proveedor. Es un campo opcional, pero cuando se proporciona
debe tener un formato válido de correo electrónico y no puede repetirse, ya que se utilizará
```

como un medio de contacto formal y para el envío de notificaciones electrónicas.', 'Schema', [dbo], 'table', [Tbl_Proveedor], 'column', [email]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena la dirección física completa del proveedor, como calle, número de casa o local, barrio o colonia. Es un dato obligatorio que no puede quedar vacío, ya que resulta indispensable para la localización geográfica, envíos de mercancía y registros oficiales.', 'Schema', [dbo], 'table', [Tbl_Proveedor], 'column', [direccion]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se guarda el nombre de la ciudad donde se ubica el proveedor. Este campo es obligatorio, no puede quedar vacío y permite clasificar a los proveedores según su ubicación geográfica para fines de reportes o segmentación.', 'Schema', [dbo], 'table', [Tbl_Proveedor], 'column', [ciudad]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el nombre del estado, provincia o departamento al que pertenece la ciudad donde se ubica el proveedor. Este campo es obligatorio y debe registrarse siempre, ya que facilita la organización geográfica y la generación de informes por región.', 'Schema', [dbo], 'table', [Tbl_Proveedor], 'column', [estado_proveedor]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el código postal correspondiente a la dirección del proveedor. Este campo es opcional, pero cuando se proporciona debe cumplir con la longitud establecida y contener únicamente caracteres válidos como letras, números y guiones, ya que se utiliza para fines de envío y normalización de direcciones.', 'Schema', [dbo], 'table', [Tbl_Proveedor], 'column', [codigo_postal]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el nombre completo de la persona de contacto principal dentro del proveedor. Es un dato obligatorio que no puede quedar vacío, pues resulta esencial para las gestiones comerciales, consultas y negociaciones directas.', 'Schema', [dbo], 'table', [Tbl_Proveedor], 'column', [contacto_principal]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena la fecha en la que el proveedor fue ingresado en la base de datos. Es un campo obligatorio que no puede quedar vacío ni registrar una fecha posterior a la actual, ya que se utiliza con fines de auditoría y control histórico.', 'Schema', [dbo], 'table', [Tbl_Proveedor], 'column', [fecha_registro]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena la clasificación del proveedor según su origen, diferenciando si es de tipo local o internacional. Este dato es obligatorio y únicamente puede tomar uno de estos dos valores, lo que permite aplicar

políticas de compra, impuestos y logística según corresponda.', 'Schema', [dbo], 'table',
[Tbl_Proveedor], 'column', [tipo_proveedor]
GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Proveedor', 'CONSTRAINT', 'PK_Tbl_Proveedor'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Aquí se almacena el
identificador numérico único de cada proveedor registrado en la base de datos. No puede
repetirse, debe ser un número entero positivo mayor que cero y no puede quedar vacío.
Este campo sirve como clave primaria para identificar de forma inequívoca cada proveedor.',
'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT', 'PK_Tbl_Proveedor'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Aquí se almacena el
identificador numérico único de cada proveedor registrado en la base de datos. No puede
repetirse, debe ser un número entero positivo mayor que cero y no puede quedar vacío.
Este campo sirve como clave primaria para identificar de forma inequívoca cada proveedor.',
'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT', 'PK_Tbl_Proveedor'
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Proveedor', 'CONSTRAINT', 'CK_Proveedor_IdProveedor'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Asegura que sea positivo.',
'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT', 'CK_Proveedor_IdProveedor'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Asegura que sea positivo.',
'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT', 'CK_Proveedor_IdProveedor'
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Proveedor', 'CONSTRAINT', 'CK_Proveedor_Nombre'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'El nombre no puede estar
vacío.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT',
'CK_Proveedor_Nombre'
end
else
begin
```

```

EXEC sys.sp_addextendedproperty 'MS_Description', 'El nombre no puede estar
vacío.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT',
'CK_Proveedor_Nombre'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Proveedor', 'CONSTRAINT', 'CK_Proveedor_Telefono'))
begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'alida formato de teléfono y
longitud = 20.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT',
'CK_Proveedor_Telefono'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'alida formato de teléfono y
longitud = 20.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT',
'CK_Proveedor_Telefono'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Proveedor', 'CONSTRAINT', 'CK_Proveedor_EmailFormato'))
begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Valida estructura mínima
de correo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT',
'CK_Proveedor_EmailFormato'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Valida estructura mínima de
correo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT',
'CK_Proveedor_EmailFormato'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Proveedor', 'CONSTRAINT', 'UQ_Proveedor_Email'))
begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Evita duplicación de
correos en proveedores.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT',
'UQ_Proveedor_Email'
end
else
begin

```



```

EXEC sys.sp_addextendedproperty 'MS_Description', 'Evita duplicación de correos
en proveedores.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT',
'UQ_Proveedor_Email'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Proveedor', 'CONSTRAINT', 'CK_Proveedor_Direccion'))

```

```

begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'La dirección no puede ser
vacía.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT',
'CK_Proveedor_Direccion'
end

```

```

else
begin

```

```

EXEC sys.sp_addextendedproperty 'MS_Description', 'La dirección no puede ser
vacía.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT',
'CK_Proveedor_Direccion'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Proveedor', 'CONSTRAINT', 'CK_Proveedor_Ciudad'))

```

```

begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'La ciudad no puede ser
vacía.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT', 'CK_Proveedor_Ciudad'
end

```

```

else
begin

```

```

EXEC sys.sp_addextendedproperty 'MS_Description', 'La ciudad no puede ser
vacía.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT', 'CK_Proveedor_Ciudad'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Proveedor', 'CONSTRAINT', 'CK_Proveedor_EstadoProveedor'))

```

```

begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'El estado no puede ser
vacío.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT',
'CK_Proveedor_EstadoProveedor'
end

```

```

else
begin

```

```

EXEC sys.sp_addextendedproperty 'MS_Description', 'El estado no puede ser
vacío.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT',
'CK_Proveedor_EstadoProveedor'
end

```

```
GO
```

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Proveedor', 'CONSTRAINT', 'CK_Proveedor_CodigoPostal'))
begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Restringe longitud y
caracteres del código postal.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT',
'CK_Proveedor_CodigoPostal'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Restringe longitud y
caracteres del código postal.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT',
'CK_Proveedor_CodigoPostal'
end

```

```
GO
```

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Proveedor', 'CONSTRAINT', 'CK_Proveedor_ContactoPrincipal'))
begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'El contacto principal no
puede ser vacío.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT',
'CK_Proveedor_ContactoPrincipal'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'El contacto principal no
puede ser vacío.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT',
'CK_Proveedor_ContactoPrincipal'
end

```

```
GO
```

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Proveedor', 'CONSTRAINT', 'CK_Proveedor_FechaRegistro'))
begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'No permite fecha futura.',
'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT', 'CK_Proveedor_FechaRegistro'
end
else
begin

```

```
EXEC sys.sp_addextendedproperty 'MS_Description', 'No permite fecha futura.',
'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT', 'CK_Proveedor_FechaRegistro'
end
```

```
GO
```

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Proveedor', 'CONSTRAINT', 'CK_Proveedor_TipoProveedor'))
```

```
begin
```

```
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Solo admite "local" o
"internacional".', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT',
'CK_Proveedor_TipoProveedor'
```

```
end
```

```
else
```

```
begin
```

```
EXEC sys.sp_addextendedproperty 'MS_Description', 'Solo admite "local" o
"internacional".', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT',
'CK_Proveedor_TipoProveedor'
```

```
end
```

```
GO
```

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Proveedor', 'CONSTRAINT', 'UQ_Proveedor_IdProveedor'))
```

```
begin
```

```
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Evita duplicados en IDs de
proveedor.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT',
'UQ_Proveedor_IdProveedor'
```

```
end
```

```
else
```

```
begin
```

```
EXEC sys.sp_addextendedproperty 'MS_Description', 'Evita duplicados en IDs de
proveedor.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Proveedor', 'CONSTRAINT',
'UQ_Proveedor_IdProveedor'
```

```
end
```

```
GO
```

```
GO
```

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'table',
'Tbl_Repuesto', NULL, NULL))
```

```
begin
```

```
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Gestiona el inventario de
repuestos. Contiene código único, nombre, descripción, categoría, precio unitario, stock,
ubicación en almacén, estado (disponible o agotado) y proveedor. Permite conocer la
disponibilidad para servicios y órdenes de compra.', 'SCHEMA', 'dbo', 'table', 'Tbl_Repuesto'
```

```
end
```

```
else
begin
```

```
EXEC sys.sp_addextendedproperty 'MS_Description', 'Gestiona el inventario de repuestos.
Contiene código único, nombre, descripción, categoría, precio unitario, stock, ubicación en
almacén, estado (disponible o agotado) y proveedor. Permite conocer la disponibilidad para
servicios y órdenes de compra.', 'SCHEMA', 'dbo', 'table', 'Tbl_Repuesto'
end
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el identificador
numérico único de cada repuesto registrado en la base de datos. No puede repetirse, debe
ser un número entero positivo y no puede quedar vacío. Este campo sirve como clave
primaria para identificar cada repuesto.', 'Schema', [dbo], 'table', [Tbl_Repuesto], 'column',
[id_vehiculorepuesto]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el código único
asignado a cada repuesto. No puede repetirse entre repuestos, debe ser obligatorio y sirve
como referencia de inventario.', 'Schema', [dbo], 'table', [Tbl_Repuesto], 'column',
[codigo_repuesto]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el nombre del
repuesto. Debe contener solo caracteres alfabéticos y espacios, no puede quedar vacío ni
comenzar con un carácter especial.', 'Schema', [dbo], 'table', [Tbl_Repuesto], 'column',
[nombre]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena una breve descripción
del repuesto. Es opcional, puede quedar vacío, pero si se ingresa debe tener un máximo de
255 caracteres.', 'Schema', [dbo], 'table', [Tbl_Repuesto], 'column', [descripcion]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el precio de venta por
unidad del repuesto. Debe ser un número decimal mayor o igual a cero y no puede quedar
vacío.', 'Schema', [dbo], 'table', [Tbl_Repuesto], 'column', [precio_unitario]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena la cantidad disponible
del repuesto en inventario. Debe ser un número entero mayor o igual a cero y no puede
quedar vacío.', 'Schema', [dbo], 'table', [Tbl_Repuesto], 'column', [stock]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el identificador del
proveedor asociado al repuesto. Es un dato obligatorio y corresponde a una clave foránea
que enlaza con la tabla Proveedor. Debe existir previamente en la tabla Proveedor.',
'Schema', [dbo], 'table', [Tbl_Repuesto], 'column', [id_proveedor]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena la fecha en que el
repuesto ingresó al almacén o al inventario de la empresa. Debe ingresarse
obligatoriamente (NO puede quedar vacío), debe ser una fecha válida y no puede ser una
fecha futura (es decir, = la fecha actual).', 'Schema', [dbo], 'table', [Tbl_Repuesto], 'column',
[fecha_ingreso]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena la ubicación física del
repuesto dentro del almacén. Es opcional, puede quedar vacío, y se usará para identificar el
estante o pasillo correspondiente.', 'Schema', [dbo], 'table', [Tbl_Repuesto], 'column',
[ubicacion_almacen]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena la categoría a la que
pertenece el repuesto (ejemplo: motor, suspensión, eléctrico). Es obligatoria y debe
contener únicamente valores predefinidos.', 'Schema', [dbo], 'table', [Tbl_Repuesto],
'column', [categoria]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el estado actual del
repuesto en el inventario. Solo se permiten valores predefinidos como disponible o agotado.
Es obligatorio y no puede quedar vacío.', 'Schema', [dbo], 'table', [Tbl_Repuesto], 'column',
[estado_repuesto]
GO
```

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Repuesto', 'CONSTRAINT', 'FK_Tbl_Repuesto_Tbl_Proveedor'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Relaciona cada repuesto
con su proveedor principal. Asegura que todo repuesto registrado provenga de un
proveedor existente.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Repuesto', 'CONSTRAINT',
'FK_Tbl_Repuesto_Tbl_Proveedor'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Relaciona cada repuesto
con su proveedor principal. Asegura que todo repuesto registrado provenga de un
proveedor existente.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Repuesto', 'CONSTRAINT',
'FK_Tbl_Repuesto_Tbl_Proveedor'
end
GO
```

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Repuesto', 'INDEX', 'IXFK_Tbl_Repuesto_Tbl_Proveedor'))
begin
```

```

EXEC sys.sp_updateextendedproperty 'MS_Description', 'Optimiza consultas que
listan repuestos según el proveedor que los suministra.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Repuesto', 'INDEX', 'IXFK_Tbl_Repuesto_Tbl_Proveedor'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Optimiza consultas que
listan repuestos según el proveedor que los suministra.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Repuesto', 'INDEX', 'IXFK_Tbl_Repuesto_Tbl_Proveedor'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Repuesto', 'CONSTRAINT', 'PK_Tbl_Repuesto'))
begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Identificador único de cada
repuesto almacenado en el inventario. Es un valor numérico entero, positivo y exclusivo. Su
función es facilitar la gestión de repuestos y garantizar la trazabilidad en órdenes de
compra, ventas o servicios.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Repuesto', 'CONSTRAINT',
'PK_Tbl_Repuesto'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Identificador único de cada
repuesto almacenado en el inventario. Es un valor numérico entero, positivo y exclusivo. Su
función es facilitar la gestión de repuestos y garantizar la trazabilidad en órdenes de
compra, ventas o servicios.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Repuesto', 'CONSTRAINT',
'PK_Tbl_Repuesto'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Repuesto', 'CONSTRAINT', 'CK_Repuesto_IdRepuesto'))
begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Verifica que el ID sea
positivo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Repuesto', 'CONSTRAINT',
'CK_Repuesto_IdRepuesto'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Verifica que el ID sea
positivo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Repuesto', 'CONSTRAINT',
'CK_Repuesto_IdRepuesto'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Repuesto', 'CONSTRAINT', 'UQ_Repuesto_Codigo'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Evita códigos de repuesto
repetidos.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Repuesto', 'CONSTRAINT',
'UQ_Repuesto_Codigo'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Evita códigos de repuesto
repetidos.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Repuesto', 'CONSTRAINT',
'UQ_Repuesto_Codigo'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Repuesto', 'CONSTRAINT', 'CK_Repuesto_Nombre_IniciaLetra'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'El nombre debe comenzar
con letra.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Repuesto', 'CONSTRAINT',
'CK_Repuesto_Nombre_IniciaLetra'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'El nombre debe comenzar
con letra.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Repuesto', 'CONSTRAINT',
'CK_Repuesto_Nombre_IniciaLetra'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Repuesto', 'CONSTRAINT', 'CK_Repuesto_Categoria'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Limita categoría a “motor”,
“suspensión”, “eléctrico”, “carrocería”.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Repuesto',
'CONSTRAINT', 'CK_Repuesto_Categoria'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Limita categoría a “motor”,
“suspensión”, “eléctrico”, “carrocería”.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Repuesto',
'CONSTRAINT', 'CK_Repuesto_Categoria'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Repuesto', 'CONSTRAINT', 'CK_Repuesto_PrecioUnitario'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'No permite precios
negativos.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Repuesto', 'CONSTRAINT',
'CK_Repuesto_PrecioUnitario'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'No permite precios
negativos.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Repuesto', 'CONSTRAINT',
'CK_Repuesto_PrecioUnitario'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Repuesto', 'CONSTRAINT', 'CK_Repuesto_Stock'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Asegura stock no
negativo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Repuesto', 'CONSTRAINT', 'CK_Repuesto_Stock'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Asegura stock no negativo.',
'SCHEMA', 'dbo', 'TABLE', 'Tbl_Repuesto', 'CONSTRAINT', 'CK_Repuesto_Stock'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Repuesto', 'CONSTRAINT', 'CK_Repuesto_FechaIngreso'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Fecha de ingreso debe ser
actual o pasada.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Repuesto', 'CONSTRAINT',
'CK_Repuesto_FechaIngreso'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Fecha de ingreso debe ser
actual o pasada.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Repuesto', 'CONSTRAINT',
'CK_Repuesto_FechaIngreso'
end

```

GO


```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Repuesto', 'CONSTRAINT', 'CK_Repuesto_Estado'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Solo admite "disponible" o
"agotado".', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Repuesto', 'CONSTRAINT',
'CK_Repuesto_Estado'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Solo admite "disponible" o
"agotado".', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Repuesto', 'CONSTRAINT',
'CK_Repuesto_Estado'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Repuesto', 'CONSTRAINT', 'UQ_Repuesto_IdRepuesto'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Garantiza unicidad del ID
de repuesto.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Repuesto', 'CONSTRAINT',
'UQ_Repuesto_IdRepuesto'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Garantiza unicidad del ID de
repuesto.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Repuesto', 'CONSTRAINT',
'UQ_Repuesto_IdRepuesto'
end

```

GO

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'table',
'Tbl_Servicio', NULL, NULL))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Define los tipos de servicios que
ofrece la empresa (mantenimiento o reparación). Incluye descripción, costo, duración,
garantía, si requiere repuestos, nivel de dificultad, fecha de creación y observaciones. Es el
catálogo de servicios que luego se aplican a los vehículos.', 'SCHEMA', 'dbo', 'table',
'Tbl_Servicio'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Define los tipos de servicios que
ofrece la empresa (mantenimiento o reparación). Incluye descripción, costo, duración,
garantía, si requiere repuestos, nivel de dificultad, fecha de creación y observaciones. Es el

```

```
catálogo de servicios que luego se aplican a los vehículos.', 'SCHEMA', 'dbo', 'table',
'Tbl_Servicio'
end
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el identificador
numérico único de cada servicio registrado en la base de datos. No puede repetirse, debe
ser un número entero positivo y no puede quedar vacío. Este campo sirve como clave
primaria para identificar cada registro de servicio.', 'Schema', [dbo], 'table', [Tbl_Servicio],
'column', [id_servicio]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el nombre del servicio
ofrecido por la empresa (por ejemplo: "Cambio de aceite", "Alineación y balanceo"). Debe
contener caracteres alfabéticos, no puede quedar vacío ni iniciar con un carácter especial.',
'Schema', [dbo], 'table', [Tbl_Servicio], 'column', [nombre]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena una descripción breve
y clara del servicio. Es opcional, puede quedar vacío, pero si se ingresa debe tener un
máximo de 255 caracteres.', 'Schema', [dbo], 'table', [Tbl_Servicio], 'column', [descripcion]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el costo base del
servicio. Debe ser un número decimal mayor o igual a cero y no puede quedar vacío.',
'Schema', [dbo], 'table', [Tbl_Servicio], 'column', [costo]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena la duración
aproximada del servicio expresada en minutos. Debe ser un número entero mayor que cero
y no puede quedar vacío.', 'Schema', [dbo], 'table', [Tbl_Servicio], 'column', [duracion_horas]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el tiempo de garantía
que ofrece el servicio (en meses). Debe ser un número entero mayor o igual a cero. Según
el tipo de servicio.', 'Schema', [dbo], 'table', [Tbl_Servicio], 'column', [garantia_meses]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se indica si el servicio requiere
repuestos para ser realizado. Solo se permiten valores predefinidos: Sí o No. Es
obligatorio.', 'Schema', [dbo], 'table', [Tbl_Servicio], 'column', [requiere_repuesto]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el nivel de dificultad
del servicio. Solo se permiten valores predefinidos como básico, medio o avanzado. Es
obligatorio.', 'Schema', [dbo], 'table', [Tbl_Servicio], 'column', [nivel_dificultad]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena la fecha en que el
servicio fue creado en el sistema. Es obligatoria, no puede quedar vacía y no puede ser
mayor a la fecha actual.', 'Schema', [dbo], 'table', [Tbl_Servicio], 'column', [fecha_creacion]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena cualquier comentario
adicional sobre el servicio. Este campo es opcional y puede quedar vacío.', 'Schema', [dbo],
'table', [Tbl_Servicio], 'column', [observaciones]
GO
```

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Servicio', 'CONSTRAINT', 'PK_Tbl_Servicio'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Representa el código
único de cada servicio ofrecido por la empresa. Es un número entero positivo, obligatorio y
no repetido. Permite catalogar los diferentes tipos de servicios de mantenimiento o
reparación.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Servicio', 'CONSTRAINT', 'PK_Tbl_Servicio'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Representa el código único
de cada servicio ofrecido por la empresa. Es un número entero positivo, obligatorio y no
repetido. Permite catalogar los diferentes tipos de servicios de mantenimiento o reparación.',
'SCHEMA', 'dbo', 'TABLE', 'Tbl_Servicio', 'CONSTRAINT', 'PK_Tbl_Servicio'
end
```

```
GO
```

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Servicio', 'CONSTRAINT', 'CK_Servicio_IdServicio'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Asegura que el ID sea
positivo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Servicio', 'CONSTRAINT', 'CK_Servicio_IdServicio'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Asegura que el ID sea
positivo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Servicio', 'CONSTRAINT', 'CK_Servicio_IdServicio'
end
```

```
GO
```

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Servicio', 'CONSTRAINT', 'UQ_Servicio_IdServicio'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Evita IDs de servicio
duplicados.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Servicio', 'CONSTRAINT',
'UQ_Servicio_IdServicio'
```

```

    end
else
    begin
        EXEC sys.sp_addextendedproperty 'MS_Description', 'Evita IDs de servicio
duplicados.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Servicio', 'CONSTRAINT',
'UQ_Servicio_IdServicio'
    end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Servicio', 'CONSTRAINT', 'CK_Servicio_Nombre_IniciaLetra'))
    begin
        EXEC sys.sp_updateextendedproperty 'MS_Description', 'El nombre debe comenzar
con letra.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Servicio', 'CONSTRAINT',
'CK_Servicio_Nombre_IniciaLetra'
    end
else
    begin
        EXEC sys.sp_addextendedproperty 'MS_Description', 'El nombre debe comenzar
con letra.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Servicio', 'CONSTRAINT',
'CK_Servicio_Nombre_IniciaLetra'
    end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Servicio', 'CONSTRAINT', 'CK_Servicio_Costo'))
    begin
        EXEC sys.sp_updateextendedproperty 'MS_Description', 'No permite costos
negativos.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Servicio', 'CONSTRAINT', 'CK_Servicio_Costo'
    end
else
    begin
        EXEC sys.sp_addextendedproperty 'MS_Description', 'No permite costos
negativos.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Servicio', 'CONSTRAINT', 'CK_Servicio_Costo'
    end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Servicio', 'CONSTRAINT', 'CK_Servicio_Duracion'))
    begin
        EXEC sys.sp_updateextendedproperty 'MS_Description', 'Obliga a que la duración
sea mayor que 0.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Servicio', 'CONSTRAINT',
'CK_Servicio_Duracion'
    end
else

```

```

begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Obliga a que la duración
sea mayor que 0.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Servicio', 'CONSTRAINT',
'CK_Servicio_Duracion'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Servicio', 'CONSTRAINT', 'CK_Servicio_Garantia'))

```

```

    begin
        EXEC sys.sp_updateextendedproperty 'MS_Description', 'Asegura que los meses de
garantía no sean negativos.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Servicio', 'CONSTRAINT',
'CK_Servicio_Garantia'
    end

```

else

```

    begin
        EXEC sys.sp_addextendedproperty 'MS_Description', 'Asegura que los meses de
garantía no sean negativos.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Servicio', 'CONSTRAINT',
'CK_Servicio_Garantia'
    end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Servicio', 'CONSTRAINT', 'CK_Servicio_RequiereRepuesto'))

```

```

    begin
        EXEC sys.sp_updateextendedproperty 'MS_Description', 'Solo admite valores “Sí” o
“No”.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Servicio', 'CONSTRAINT',
'CK_Servicio_RequiereRepuesto'
    end

```

else

```

    begin
        EXEC sys.sp_addextendedproperty 'MS_Description', 'Solo admite valores “Sí” o
“No”.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Servicio', 'CONSTRAINT',
'CK_Servicio_RequiereRepuesto'
    end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Servicio', 'CONSTRAINT', 'CK_Servicio_NivelDificultad'))

```

```

    begin
        EXEC sys.sp_updateextendedproperty 'MS_Description', 'Limita a “básico”, “medio”,
“avanzado”.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Servicio', 'CONSTRAINT',
'CK_Servicio_NivelDificultad'
    end

```

end

else

```

begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Limita a "básico", "medio",
"avanzado".', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Servicio', 'CONSTRAINT',
'CK_Servicio_NivelDificultad'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Servicio', 'CONSTRAINT', 'CK_Servicio_FechaCreacion'))

```

```

begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'No puede superar la fecha
actual.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Servicio', 'CONSTRAINT',
'CK_Servicio_FechaCreacion'
end

```

```

else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'No puede superar la fecha
actual.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Servicio', 'CONSTRAINT',
'CK_Servicio_FechaCreacion'
end

```

GO

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'table',
'Tbl_ServicioRealizado', NULL, NULL))

```

```

begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Aquí se registran todos los
servicios efectivamente realizados a vehículos. Se vincula con la tabla Servicio (qué tipo de
servicio), con Empleado (quién lo realizó), y permite relacionarse tanto con un vehículo
propio de inventario (Vehiculo) como con un vehículo externo (VehiculoExterno). De esta
forma, cualquier servicio queda ligado a un vehículo concreto, sin obligar a que este
pertenezca al inventario de ventas.', 'SCHEMA', 'dbo', 'table', 'Tbl_ServicioRealizado'
end

```

```

else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Aquí se registran todos los servicios
efectivamente realizados a vehículos. Se vincula con la tabla Servicio (qué tipo de servicio),
con Empleado (quién lo realizó), y permite relacionarse tanto con un vehículo propio de
inventario (Vehiculo) como con un vehículo externo (VehiculoExterno). De esta forma,
cualquier servicio queda ligado a un vehículo concreto, sin obligar a que este pertenezca al
inventario de ventas.', 'SCHEMA', 'dbo', 'table', 'Tbl_ServicioRealizado'
end

```

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el identificador numérico único de cada servicio realizado en el taller. Es la clave primaria de la tabla y

permite distinguir de manera única cada registro de servicio. Debe ser un número entero positivo mayor que cero y no puede quedar vacío.', 'Schema', [dbo], 'table', [Tbl_ServicioRealizado], 'column', [id_servicio_realizado]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el identificador del empleado encargado de realizar el servicio. Es obligatorio y funciona como una clave foránea que enlaza con la tabla Empleado, garantizando que solo empleados registrados puedan ser asignados a un servicio', 'Schema', [dbo], 'table', [Tbl_ServicioRealizado], 'column', [id_empleado]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena la fecha exacta en la que se realizó el servicio. Debe ingresarse en formato de fecha válido, no puede quedar vacío y no puede ser posterior a la fecha actual.', 'Schema', [dbo], 'table', [Tbl_ServicioRealizado], 'column', [fecha_servicio]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el identificador del repuesto utilizado en el servicio, en caso de que se requiera. Es una clave foránea que referencia la tabla Repuesto. Este campo puede quedar vacío si no se utilizó ningún repuesto.', 'Schema', [dbo], 'table', [Tbl_ServicioRealizado], 'column', [id_repuesto]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena la cantidad de unidades de repuesto utilizadas en el servicio. Debe ser un número entero mayor o igual a cero. Es opcional, pero si existe un id_repuesto, este campo debe contener un valor mayor a cero.', 'Schema', [dbo], 'table', [Tbl_ServicioRealizado], 'column', [cantidad_repuesto]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el costo total del servicio realizado, incluyendo repuestos y mano de obra. Debe ser un número decimal mayor o igual a cero y no puede quedar vacío.', 'Schema', [dbo], 'table', [Tbl_ServicioRealizado], 'column', [costo_total]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el estado actual del servicio. Solo se permiten valores predefinidos como pendiente o finalizado. Es obligatorio y no puede quedar vacío.', 'Schema', [dbo], 'table', [Tbl_ServicioRealizado], 'column', [estado_servicio]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacenan comentarios adicionales relacionados con el servicio realizado (ejemplo: detalles del diagnóstico, recomendaciones al cliente). Es un campo opcional y puede quedar vacío.', 'Schema', [dbo], 'table', [Tbl_ServicioRealizado], 'column', [observaciones]
GO

```
EXEC sp_addextendedproperty 'MS_Description', 'FK que relaciona el servicio realizado con la cita que lo originó, es obligatoria y proviene de la tabla Tbl_CitaServicio', 'Schema', [dbo], 'table', [Tbl_ServicioRealizado], 'column', [id_cita_servicio]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Funciona como FK, relacionando la tabla ServicioRealizado con la cita que lo originó, ubicada en la Tbl_CitaServicio', 'Schema', [dbo], 'table', [Tbl_ServicioRealizado], 'column', [id_cita]
GO
```

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_ServicioRealizado', 'CONSTRAINT', 'FK_Tbl_ServicioRealizado_Tbl_CitaServicio'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'La clave foránea garantiza la trazabilidad entre la cita programada y la ejecución real del servicio. Cada registro de servicio realizado debe estar vinculado a una cita previamente registrada en Tbl_CitaServicio, asegurando coherencia entre la agenda y la atención efectuada. Una cita puede no llegar a generar un servicio (por cancelación o inasistencia), pero un servicio realizado siempre proviene de una cita existente.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_ServicioRealizado', 'CONSTRAINT', 'FK_Tbl_ServicioRealizado_Tbl_CitaServicio'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'La clave foránea garantiza la trazabilidad entre la cita programada y la ejecución real del servicio. Cada registro de servicio realizado debe estar vinculado a una cita previamente registrada en Tbl_CitaServicio, asegurando coherencia entre la agenda y la atención efectuada. Una cita puede no llegar a generar un servicio (por cancelación o inasistencia), pero un servicio realizado siempre proviene de una cita existente.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_ServicioRealizado', 'CONSTRAINT', 'FK_Tbl_ServicioRealizado_Tbl_CitaServicio'
end
```

```
GO
```

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_ServicioRealizado', 'CONSTRAINT', 'FK_Tbl_ServicioRealizado_Tbl_Empleado'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Identifica al empleado encargado de realizar el servicio. Garantiza control sobre la responsabilidad de ejecución.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_ServicioRealizado', 'CONSTRAINT', 'FK_Tbl_ServicioRealizado_Tbl_Empleado'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Identifica al empleado encargado de realizar el servicio. Garantiza control sobre la responsabilidad de ejecución.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_ServicioRealizado', 'CONSTRAINT', 'FK_Tbl_ServicioRealizado_Tbl_Empleado'
```


end

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_ServicioRealizado', 'CONSTRAINT', 'FK_Tbl_ServicioRealizado_Tbl_Repuesto'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Conecta el servicio
realizado con los repuestos utilizados. Permite registrar correctamente qué piezas se
consumieron en la operación.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_ServicioRealizado',
'CONSTRAINT', 'FK_Tbl_ServicioRealizado_Tbl_Repuesto'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Conecta el servicio
realizado con los repuestos utilizados. Permite registrar correctamente qué piezas se
consumieron en la operación.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_ServicioRealizado',
'CONSTRAINT', 'FK_Tbl_ServicioRealizado_Tbl_Repuesto'
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_ServicioRealizado', 'INDEX', 'IXFK_Tbl_ServicioRealizado_Tbl_CitaServicio'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Se utiliza para optimizar la
búsqueda y validación de registros de Tbl_ServicioRealizado en función de la cita asociada
en Tbl_CitaServicio. Facilita consultas frecuentes como “obtener todos los servicios
realizados a partir de una cita específica” y mejora el rendimiento de las operaciones entre
ambas tablas.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_ServicioRealizado', 'INDEX',
'IXFK_Tbl_ServicioRealizado_Tbl_CitaServicio'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Se utiliza para optimizar la
búsqueda y validación de registros de Tbl_ServicioRealizado en función de la cita asociada
en Tbl_CitaServicio. Facilita consultas frecuentes como “obtener todos los servicios
realizados a partir de una cita específica” y mejora el rendimiento de las operaciones entre
ambas tablas.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_ServicioRealizado', 'INDEX',
'IXFK_Tbl_ServicioRealizado_Tbl_CitaServicio'
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_ServicioRealizado', 'INDEX', 'IXFK_Tbl_ServicioRealizado_Tbl_Empleado'))
begin
```

```
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Facilita consultas de
servicios realizados por empleado.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_ServicioRealizado',
'INDEX', 'IXFK_Tbl_ServicioRealizado_Tbl_Empleado'
```

```
end
```

```
else
```

```
begin
```

```
EXEC sys.sp_addextendedproperty 'MS_Description', 'Facilita consultas de
servicios realizados por empleado.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_ServicioRealizado',
'INDEX', 'IXFK_Tbl_ServicioRealizado_Tbl_Empleado'
```

```
end
```

```
GO
```

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_ServicioRealizado', 'INDEX', 'IXFK_Tbl_ServicioRealizado_Tbl_Repuesto'))
```

```
begin
```

```
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Acelera búsquedas de
servicios en los que se usaron repuestos.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_ServicioRealizado', 'INDEX', 'IXFK_Tbl_ServicioRealizado_Tbl_Repuesto'
```

```
end
```

```
else
```

```
begin
```

```
EXEC sys.sp_addextendedproperty 'MS_Description', 'Acelera búsquedas de
servicios en los que se usaron repuestos.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_ServicioRealizado', 'INDEX', 'IXFK_Tbl_ServicioRealizado_Tbl_Repuesto'
```

```
end
```

```
GO
```

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_ServicioRealizado', 'CONSTRAINT', 'PK_Tbl_ServicioRealizado'))
```

```
begin
```

```
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Identificador único de cada
registro de servicio efectivamente realizado a un vehículo (interno o externo). Es un número
entero positivo, no repetido y obligatorio. Sirve como referencia principal para detallar los
servicios ejecutados.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_ServicioRealizado', 'CONSTRAINT',
'PK_Tbl_ServicioRealizado'
```

```
end
```

```
else
```

```
begin
```

```
EXEC sys.sp_addextendedproperty 'MS_Description', 'Identificador único de cada
registro de servicio efectivamente realizado a un vehículo (interno o externo). Es un número
entero positivo, no repetido y obligatorio. Sirve como referencia principal para detallar los
servicios ejecutados.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_ServicioRealizado', 'CONSTRAINT',
'PK_Tbl_ServicioRealizado'
```

```
end
```

```
GO
```

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_ServicioRealizado', 'CONSTRAINT', 'CK_ServReal_Id'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Asegura que el ID sea
positivo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_ServicioRealizado', 'CONSTRAINT',
'CK_ServReal_Id'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Asegura que el ID sea
positivo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_ServicioRealizado', 'CONSTRAINT',
'CK_ServReal_Id'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_ServicioRealizado', 'CONSTRAINT', 'CK_ServReal_Fecha'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'No permite fechas futuras
para el servicio.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_ServicioRealizado', 'CONSTRAINT',
'CK_ServReal_Fecha'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'No permite fechas futuras
para el servicio.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_ServicioRealizado', 'CONSTRAINT',
'CK_ServReal_Fecha'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_ServicioRealizado', 'CONSTRAINT', 'CK_ServReal_Cantidad'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'La cantidad de repuestos
debe ser = 0 o nula.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_ServicioRealizado', 'CONSTRAINT',
'CK_ServReal_Cantidad'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'La cantidad de repuestos
debe ser = 0 o nula.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_ServicioRealizado', 'CONSTRAINT',
'CK_ServReal_Cantidad'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_ServicioRealizado', 'CONSTRAINT', 'CK_ServReal_Costo'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'El costo total no puede ser
negativo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_ServicioRealizado', 'CONSTRAINT',
'CK_ServReal_Costo'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'El costo total no puede ser
negativo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_ServicioRealizado', 'CONSTRAINT',
'CK_ServReal_Costo'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_ServicioRealizado', 'CONSTRAINT', 'CK_ServReal_Estado'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Limita estado a
“pendiente” o “finalizado”.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_ServicioRealizado',
'CONSTRAINT', 'CK_ServReal_Estado'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Limita estado a “pendiente”
o “finalizado”.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_ServicioRealizado', 'CONSTRAINT',
'CK_ServReal_Estado'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_ServicioRealizado', 'CONSTRAINT', 'UQ_ServReal_IdServicioRealizado'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Evita duplicados en IDs.',
'SCHEMA', 'dbo', 'TABLE', 'Tbl_ServicioRealizado', 'CONSTRAINT',
'UQ_ServReal_IdServicioRealizado'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Evita duplicados en IDs.',
'SCHEMA', 'dbo', 'TABLE', 'Tbl_ServicioRealizado', 'CONSTRAINT',
'UQ_ServReal_IdServicioRealizado'
end

```

GO

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'table',
'Tbl_Vehiculo', NULL, NULL))
```

```
begin
```

```
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Registra todos los vehículos
nuevos o usados disponibles en el inventario de la empresa. Incluye datos técnicos (marca,
modelo, año, color, tipo de combustible y transmisión, kilometraje), estado del vehículo
(disponible, vendido, reservado), precio y fecha de ingreso. Es la base para las ventas y los
servicios realizados.', 'SCHEMA', 'dbo', 'table', 'Tbl_Vehiculo'
```

```
end
```

```
else
```

```
begin
```

```
EXEC sys.sp_addextendedproperty 'MS_Description', 'Registra todos los vehículos nuevos
o usados disponibles en el inventario de la empresa. Incluye datos técnicos (marca, modelo,
año, color, tipo de combustible y transmisión, kilometraje), estado del vehículo (disponible,
vendido, reservado), precio y fecha de ingreso. Es la base para las ventas y los servicios
realizados.', 'SCHEMA', 'dbo', 'table', 'Tbl_Vehiculo'
```

```
end
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el identificador
numérico único de cada vehículo registrado en la base de datos. No puede repetirse, debe
ser un número entero positivo y no puede quedar vacío. Este campo sirve como clave
primaria para identificar cada registro de vehículo.', 'Schema', [dbo], 'table', [Tbl_Vehiculo],
'column', [id_vehiculo]
```

GO

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se guarda el número de chasis
único del vehículo. Es un valor de texto (hasta 50 caracteres) que identifica de manera
exclusiva al vehículo. No puede quedar vacío ni repetirse en otro vehículo.', 'Schema', [dbo],
'table', [Tbl_Vehiculo], 'column', [numero_chasis]
```

GO

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se guarda la marca del vehículo
(por ejemplo, Toyota, Honda, Nissan). Es un texto obligatorio y no puede iniciar con
caracteres especiales ni estar vacío.', 'Schema', [dbo], 'table', [Tbl_Vehiculo], 'column',
[marca]
```

GO

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el modelo del vehículo
(por ejemplo, Corolla, Civic, Frontier). Es un texto obligatorio y no puede iniciar con
caracteres especiales ni quedar vacío.', 'Schema', [dbo], 'table', [Tbl_Vehiculo], 'column',
[modelo]
```

GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se guarda el año de fabricación del vehículo. Debe ser un número entero comprendido entre 1900 y el año actual. No puede quedar vacío.', 'Schema', [dbo], 'table', [Tbl_Vehiculo], 'column', [anio]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el color del vehículo (por ejemplo, "rojo", "negro metálico"). Es un texto obligatorio y no puede quedar vacío.', 'Schema', [dbo], 'table', [Tbl_Vehiculo], 'column', [color]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se guarda el tipo de combustible que utiliza el vehículo. Solo se permiten valores predefinidos como "gasolina", "diésel", "eléctrico", "híbrido". Es obligatorio.', 'Schema', [dbo], 'table', [Tbl_Vehiculo], 'column', [tipo_combustible]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el tipo de transmisión del vehículo. Solo se permiten valores predefinidos como "automático", "mecánico" o "semiautomático". Es obligatorio.', 'Schema', [dbo], 'table', [Tbl_Vehiculo], 'column', [tipo_transmision]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se guarda el precio de venta del vehículo. Es un número decimal (con dos decimales) mayor o igual a cero. No puede quedar vacío.', 'Schema', [dbo], 'table', [Tbl_Vehiculo], 'column', [precio]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el estado actual del vehículo dentro del inventario. Solo se permiten valores predefinidos como "disponible", "vendido" o "reservado". Es obligatorio.', 'Schema', [dbo], 'table', [Tbl_Vehiculo], 'column', [estado_vehiculo]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se guarda la fecha en que el vehículo ingresó al inventario de la empresa. No puede ser mayor a la fecha actual ni quedar vacío.', 'Schema', [dbo], 'table', [Tbl_Vehiculo], 'column', [fecha_ingreso]
GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el kilometraje actual del vehículo. Es un número entero mayor o igual a 0 y menor o igual a 1,000,000. No puede quedar vacío.', 'Schema', [dbo], 'table', [Tbl_Vehiculo], 'column', [kilometraje]
GO

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Vehiculo', 'CONSTRAINT', 'PK_Vehiculo'))
begin

EXEC sys.sp_updateextendedproperty 'MS_Description', 'Identificador único de cada vehículo que ha sido adquirido y registrado por la empresa. Debe ser un valor numérico

```

positivo y exclusivo, garantizando que no existan duplicados. Es fundamental para
relacionar el vehículo con ventas, clientes y servicios.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Vehiculo', 'CONSTRAINT', 'PK_Vehiculo'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Identificador único de cada
vehículo que ha sido adquirido y registrado por la empresa. Debe ser un valor numérico
positivo y exclusivo, garantizando que no existan duplicados. Es fundamental para
relacionar el vehículo con ventas, clientes y servicios.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Vehiculo', 'CONSTRAINT', 'PK_Vehiculo'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Vehiculo', 'CONSTRAINT', 'UQ_Vehiculo_id'))
begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Garantiza unicidad del ID
de vehículo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Vehiculo', 'CONSTRAINT', 'UQ_Vehiculo_id'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Garantiza unicidad del ID de
vehículo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Vehiculo', 'CONSTRAINT', 'UQ_Vehiculo_id'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Vehiculo', 'CONSTRAINT', 'CK_Vehiculo_IdVehiculo'))
begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Verifica que el ID sea
positivo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Vehiculo', 'CONSTRAINT',
'CK_Vehiculo_IdVehiculo'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Verifica que el ID sea
positivo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Vehiculo', 'CONSTRAINT',
'CK_Vehiculo_IdVehiculo'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Vehiculo', 'CONSTRAINT', 'UQ_Vehiculo_NumeroChasis'))
begin

```

```

        EXEC sys.sp_updateextendedproperty 'MS_Description', 'Asegura que el número de
chasis sea único.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Vehiculo', 'CONSTRAINT',
'UQ_Vehiculo_NumeroChasis'
    end
else
    begin
        EXEC sys.sp_addextendedproperty 'MS_Description', 'Asegura que el número de
chasis sea único.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Vehiculo', 'CONSTRAINT',
'UQ_Vehiculo_NumeroChasis'
    end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Vehiculo', 'CONSTRAINT', 'CK_Vehiculo_Anio'))
    begin
        EXEC sys.sp_updateextendedproperty 'MS_Description', 'Restringe año a valores
entre 1900 y el año actual.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Vehiculo', 'CONSTRAINT',
'CK_Vehiculo_Anio'
    end
else
    begin
        EXEC sys.sp_addextendedproperty 'MS_Description', 'Restringe año a valores
entre 1900 y el año actual.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Vehiculo', 'CONSTRAINT',
'CK_Vehiculo_Anio'
    end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Vehiculo', 'CONSTRAINT', 'CK_Vehiculo_Precio'))
    begin
        EXEC sys.sp_updateextendedproperty 'MS_Description', 'No permite precio
negativo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Vehiculo', 'CONSTRAINT', 'CK_Vehiculo_Precio'
    end
else
    begin
        EXEC sys.sp_addextendedproperty 'MS_Description', 'No permite precio
negativo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Vehiculo', 'CONSTRAINT', 'CK_Vehiculo_Precio'
    end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Vehiculo', 'CONSTRAINT', 'CK_Vehiculo_Kilometraje'))
    begin

```



```

        EXEC sys.sp_updateextendedproperty 'MS_Description', 'Asegura kilometraje dentro
de 0 a 1,000,000.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Vehiculo', 'CONSTRAINT',
'CK_Vehiculo_Kilometraje'
    end
else
    begin
        EXEC sys.sp_addextendedproperty 'MS_Description', 'Asegura kilometraje dentro
de 0 a 1,000,000.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Vehiculo', 'CONSTRAINT',
'CK_Vehiculo_Kilometraje'
    end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Vehiculo', 'CONSTRAINT', 'CK_Vehiculo_TipoCombustible'))
    begin
        EXEC sys.sp_updateextendedproperty 'MS_Description', 'Solo admite "gasolina",
"diesel", "eléctrico", "híbrido".', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Vehiculo', 'CONSTRAINT',
'CK_Vehiculo_TipoCombustible'
    end
else
    begin
        EXEC sys.sp_addextendedproperty 'MS_Description', 'Solo admite "gasolina",
"diesel", "eléctrico", "híbrido".', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Vehiculo', 'CONSTRAINT',
'CK_Vehiculo_TipoCombustible'
    end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Vehiculo', 'CONSTRAINT', 'CK_Vehiculo_TipoTransmision'))
    begin
        EXEC sys.sp_updateextendedproperty 'MS_Description', 'Solo admite "automático",
"mecánico", "semiautomático".', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Vehiculo', 'CONSTRAINT',
'CK_Vehiculo_TipoTransmision'
    end
else
    begin
        EXEC sys.sp_addextendedproperty 'MS_Description', 'Solo admite "automático",
"mecánico", "semiautomático".', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Vehiculo', 'CONSTRAINT',
'CK_Vehiculo_TipoTransmision'
    end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Vehiculo', 'CONSTRAINT', 'CK_Vehiculo_Estado'))
    begin

```

```

EXEC sys.sp_updateextendedproperty 'MS_Description', 'Limita estado a
“disponible”, “vendido”, “reservado”.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Vehiculo',
'CONSTRAINT', 'CK_Vehiculo_Estado'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Limita estado a “disponible”,
“vendido”, “reservado”.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Vehiculo', 'CONSTRAINT',
'CK_Vehiculo_Estado'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Vehiculo', 'CONSTRAINT', 'CK_Vehiculo_FechaIngreso'))
begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Fecha de ingreso no
puede ser futura.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Vehiculo', 'CONSTRAINT',
'CK_Vehiculo_FechaIngreso'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Fecha de ingreso no puede
ser futura.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Vehiculo', 'CONSTRAINT',
'CK_Vehiculo_FechaIngreso'
end

```

GO

```

GO
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'table',
'Tbl_VehiculoExterno', NULL, NULL))
begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Aquí se registran los vehículos
que no forman parte del inventario de ventas, pero que ingresan al taller para recibir
mantenimiento o reparación. Contiene solo los datos básicos necesarios para identificar al
vehículo y dar seguimiento al servicio, sin requerir información de precio, número de chasis
u otros datos exclusivos de los vehículos de inventario.', 'SCHEMA', 'dbo', 'table',
'Tbl_VehiculoExterno'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Aquí se registran los vehículos que
no forman parte del inventario de ventas, pero que ingresan al taller para recibir
mantenimiento o reparación. Contiene solo los datos básicos necesarios para identificar al
vehículo y dar seguimiento al servicio, sin requerir información de precio, número de chasis

```

```
u otros datos exclusivos de los vehículos de inventario.', 'SCHEMA', 'dbo', 'table',
'Tbl_VehiculoExterno'
end
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el identificador
numérico único de cada vehículo externo registrado en la base de datos. Este campo sirve
como clave primaria y permite diferenciar de manera inequívoca a cada vehículo externo.
Debe ser un número entero positivo mayor que cero y no puede quedar vacío.', 'Schema',
[dbo], 'table', [Tbl_VehiculoExterno], 'column', [id_vehiculo_externo]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el número de placa
oficial del vehículo externo. Este valor identifica de forma única al vehículo y se utiliza como
referencia principal para búsquedas y validaciones. No puede quedar vacío, no debe
contener caracteres especiales fuera del formato de placa, y debe ser único para evitar
duplicados en la base de datos.', 'Schema', [dbo], 'table', [Tbl_VehiculoExterno], 'column',
[placa]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena la marca comercial del
vehículo externo, como por ejemplo Toyota, Nissan, Honda, etc. Debe ingresarse siempre
con caracteres alfabéticos, puede incluir espacios, pero no puede iniciar con un carácter
especial ni con un número. Este campo es obligatorio.', 'Schema', [dbo], 'table',
[Tbl_VehiculoExterno], 'column', [marca]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el modelo específico
del vehículo externo, como Corolla, Civic, Hilux, entre otros. Debe ser un valor de texto
obligatorio, no puede quedar vacío y debe permitir la combinación de letras y números,
aunque no puede iniciar con un carácter especial.', 'Schema', [dbo], 'table',
[Tbl_VehiculoExterno], 'column', [modelo]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el año de fabricación
del vehículo externo. Debe ser un número entero de cuatro dígitos comprendido entre 1900
y el año actual, para garantizar la validez del dato. No puede quedar vacío.', 'Schema',
[dbo], 'table', [Tbl_VehiculoExterno], 'column', [año]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el color principal del
vehículo externo (ejemplo: negro, rojo metálico, blanco perla). Es un valor obligatorio, no
puede quedar vacío y debe contener únicamente letras y, opcionalmente, combinaciones
como "gris claro" o "azul oscuro".', 'Schema', [dbo], 'table', [Tbl_VehiculoExterno], 'column',
[color]
GO
```

```
EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el nombre completo
del propietario del vehículo externo. Debe contener únicamente caracteres alfabéticos,
```

puede incluir espacios y guiones, pero no puede iniciar con un carácter especial ni con un número. Es obligatorio y no puede quedar vacío.', 'Schema', [dbo], 'table', [Tbl_VehiculoExterno], 'column', [propietario]
GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_VehiculoExterno', 'CONSTRAINT', 'PK_Tbl_VehiculoExterno'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Clave primaria que
distingue de forma única a cada vehículo externo (no adquirido en la empresa) que recibe
servicios. Es un número entero positivo, obligatorio y no repetido. Permite llevar control de
los servicios realizados a clientes externos.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_VehiculoExterno', 'CONSTRAINT', 'PK_Tbl_VehiculoExterno'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Clave primaria que
distingue de forma única a cada vehículo externo (no adquirido en la empresa) que recibe
servicios. Es un número entero positivo, obligatorio y no repetido. Permite llevar control de
los servicios realizados a clientes externos.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_VehiculoExterno', 'CONSTRAINT', 'PK_Tbl_VehiculoExterno'
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_VehiculoExterno', 'CONSTRAINT', 'CK_VehiculoExterno_Id'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Verifica que el ID sea
positivo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_VehiculoExterno', 'CONSTRAINT',
'CK_VehiculoExterno_Id'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Verifica que el ID sea
positivo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_VehiculoExterno', 'CONSTRAINT',
'CK_VehiculoExterno_Id'
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_VehiculoExterno', 'CONSTRAINT', 'UQ_VehiculoExterno_Placa'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Garantiza unicidad en la
placa.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_VehiculoExterno', 'CONSTRAINT',
'UQ_VehiculoExterno_Placa'
end
```

```

else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Garantiza unicidad en la
placa.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_VehiculoExterno', 'CONSTRAINT',
'UQ_VehiculoExterno_Placa'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_VehiculoExterno', 'CONSTRAINT', 'CK_VehiculoExterno_Placa'))
begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Valida que la placa
contenga solo letras mayúsculas y números.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_VehiculoExterno', 'CONSTRAINT', 'CK_VehiculoExterno_Placa'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Valida que la placa
contenga solo letras mayúsculas y números.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_VehiculoExterno', 'CONSTRAINT', 'CK_VehiculoExterno_Placa'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_VehiculoExterno', 'CONSTRAINT', 'CK_VehiculoExterno_Año'))
begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Restringe año a =1900 y =
año actual.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_VehiculoExterno', 'CONSTRAINT',
'CK_VehiculoExterno_Año'
end
else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Restringe año a =1900 y =
año actual.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_VehiculoExterno', 'CONSTRAINT',
'CK_VehiculoExterno_Año'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_VehiculoExterno', 'CONSTRAINT', 'CK_VehiculoExterno_Propietario'))
begin
EXEC sys.sp_updateextendedproperty 'MS_Description', 'Obliga a que el propietario
inicie con una letra.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_VehiculoExterno', 'CONSTRAINT',
'CK_VehiculoExterno_Propietario'
end

```

```

else
begin
EXEC sys.sp_addextendedproperty 'MS_Description', 'Obliga a que el propietario
inicie con una letra.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_VehiculoExterno', 'CONSTRAINT',
'CK_VehiculoExterno_Propietario'
end

```

GO

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'table',
'Tbl_Venta', NULL, NULL))

```

```

begin

```

```

EXEC sys.sp_updateextendedproperty 'MS_Description', 'Registra cada transacción de
venta de vehículos. Vincula al vehículo vendido, al cliente comprador y al empleado que
realizó la venta. Incluye precio, fecha, forma y estado de pago, así como observaciones de
la operación.', 'SCHEMA', 'dbo', 'table', 'Tbl_Venta'

```

```

end

```

```

else

```

```

begin

```

```

EXEC sys.sp_addextendedproperty 'MS_Description', 'Registra cada transacción de venta
de vehículos. Vincula al vehículo vendido, al cliente comprador y al empleado que realizó la
venta. Incluye precio, fecha, forma y estado de pago, así como observaciones de la
operación.', 'SCHEMA', 'dbo', 'table', 'Tbl_Venta'

```

```

end

```

```

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el identificador
numérico único de cada venta registrada en la base de datos. No puede repetirse, debe ser
un número entero positivo y no puede quedar vacío. Este campo sirve como clave primaria
para identificar cada registro de venta.', 'Schema', [dbo], 'table', [Tbl_Venta], 'column',
[id_venta]

```

GO

```

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el identificador del
vehículo vendido. Es un dato obligatorio y corresponde a una clave foránea que enlaza con
la tabla Vehículo. Debe existir previamente en la tabla Vehículo.', 'Schema', [dbo], 'table',
[Tbl_Venta], 'column', [id_vehiculo]

```

GO

```

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el identificador del
cliente que realizó la compra. Es un dato obligatorio y corresponde a una clave foránea que
enlaza con la tabla Cliente. Debe existir previamente en la tabla Cliente.', 'Schema', [dbo],
'table', [Tbl_Venta], 'column', [id_cliente]

```

GO

```

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el identificador del
empleado que realizó la venta. Es un dato obligatorio y corresponde a una clave foránea

```

que enlaza con la tabla Empleado. Debe existir previamente en la tabla Empleado.',
 'Schema', [dbo], 'table', [Tbl_Venta], 'column', [id_empleado]
 GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena la fecha en que se
 realizó la venta. Es obligatoria, no puede quedar vacía y no puede ser mayor a la fecha
 actual.', 'Schema', [dbo], 'table', [Tbl_Venta], 'column', [fecha_venta]
 GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el precio final de venta
 del vehículo. Debe ser un número decimal mayor o igual a cero y no puede quedar vacío.',
 'Schema', [dbo], 'table', [Tbl_Venta], 'column', [precio_venta]
 GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena la forma de pago
 acordada con el cliente. Solo se permiten valores predefinidos como contado o crédito. Es
 obligatoria y no puede quedar vacía.', 'Schema', [dbo], 'table', [Tbl_Venta], 'column',
 [forma_pago]
 GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena el estado actual del
 pago de la venta. Solo se permiten valores predefinidos como pagado o pendiente. Es
 obligatorio y no puede quedar vacío.', 'Schema', [dbo], 'table', [Tbl_Venta], 'column',
 [estado_pago]
 GO

EXEC sp_addextendedproperty 'MS_Description', 'Aquí se almacena cualquier comentario
 adicional relacionado con la venta. Este campo es opcional y puede quedar vacío.',
 'Schema', [dbo], 'table', [Tbl_Venta], 'column', [observaciones]
 GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Venta', 'CONSTRAINT', 'FK_Tbl_Venta_Tbl_Cliente'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Establece el vínculo entre
la venta y el cliente comprador. Impide registrar ventas sin cliente asociado.', 'SCHEMA',
'dbo', 'TABLE', 'Tbl_Venta', 'CONSTRAINT', 'FK_Tbl_Venta_Tbl_Cliente'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Establece el vínculo entre la
venta y el cliente comprador. Impide registrar ventas sin cliente asociado.', 'SCHEMA', 'dbo',
'TABLE', 'Tbl_Venta', 'CONSTRAINT', 'FK_Tbl_Venta_Tbl_Cliente'
end
GO
```

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Venta', 'CONSTRAINT', 'FK_Tbl_Venta_Tbl_Empleado'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Relaciona la venta con el
empleado que la gestionó. Garantiza trazabilidad del personal que realizó la operación.',
'SCHEMA', 'dbo', 'TABLE', 'Tbl_Venta', 'CONSTRAINT', 'FK_Tbl_Venta_Tbl_Empleado'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Relaciona la venta con el
empleado que la gestionó. Garantiza trazabilidad del personal que realizó la operación.',
'SCHEMA', 'dbo', 'TABLE', 'Tbl_Venta', 'CONSTRAINT', 'FK_Tbl_Venta_Tbl_Empleado'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Venta', 'CONSTRAINT', 'FK_Tbl_Venta_Tbl_Vehiculo'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Vincula la venta con el
vehículo vendido. Asegura que cada venta esté asociada a un vehículo existente en
inventario.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Venta', 'CONSTRAINT',
'FK_Tbl_Venta_Tbl_Vehiculo'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Vincula la venta con el
vehículo vendido. Asegura que cada venta esté asociada a un vehículo existente en
inventario.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Venta', 'CONSTRAINT',
'FK_Tbl_Venta_Tbl_Vehiculo'
end

```

GO

```

if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Venta', 'INDEX', 'IXFK_Tbl_Venta_Tbl_Cliente'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Mejora consultas que
muestran las ventas realizadas a un cliente.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Venta',
'INDEX', 'IXFK_Tbl_Venta_Tbl_Cliente'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Mejora consultas que
muestran las ventas realizadas a un cliente.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Venta',
'INDEX', 'IXFK_Tbl_Venta_Tbl_Cliente'
end

```


GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Venta', 'INDEX', 'IXFK_Tbl_Venta_Tbl_Empleado'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Facilita búsquedas de
ventas gestionadas por un empleado específico.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Venta',
'INDEX', 'IXFK_Tbl_Venta_Tbl_Empleado'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Facilita búsquedas de
ventas gestionadas por un empleado específico.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Venta',
'INDEX', 'IXFK_Tbl_Venta_Tbl_Empleado'
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Venta', 'INDEX', 'IXFK_Tbl_Venta_Tbl_Vehiculo'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Optimiza consultas que
relacionan un vehículo con la venta correspondiente.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Venta', 'INDEX', 'IXFK_Tbl_Venta_Tbl_Vehiculo'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Optimiza consultas que
relacionan un vehículo con la venta correspondiente.', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Venta', 'INDEX', 'IXFK_Tbl_Venta_Tbl_Vehiculo'
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Venta', 'CONSTRAINT', 'PK_Venta'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Identificador único de cada
transacción de venta registrada. No puede repetirse, debe ser numérico positivo y
obligatorio. Permite relacionar la venta con cliente, empleado y vehículo vendido.',
'SCHEMA', 'dbo', 'TABLE', 'Tbl_Venta', 'CONSTRAINT', 'PK_Venta'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Identificador único de cada
transacción de venta registrada. No puede repetirse, debe ser numérico positivo y
obligatorio. Permite relacionar la venta con cliente, empleado y vehículo vendido.',
'SCHEMA', 'dbo', 'TABLE', 'Tbl_Venta', 'CONSTRAINT', 'PK_Venta'
```

end

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Venta', 'CONSTRAINT', 'CK_Venta_IdVenta'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Asegura que el ID sea
positivo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Venta', 'CONSTRAINT', 'CK_Venta_IdVenta'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Asegura que el ID sea
positivo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Venta', 'CONSTRAINT', 'CK_Venta_IdVenta'
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Venta', 'CONSTRAINT', 'CK_Venta_FechaVenta'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'No permite ventas con
fecha futura.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Venta', 'CONSTRAINT',
'CK_Venta_FechaVenta'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'No permite ventas con
fecha futura.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Venta', 'CONSTRAINT',
'CK_Venta_FechaVenta'
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Venta', 'CONSTRAINT', 'CK_Venta_PrecioVenta'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'El precio de venta no
puede ser negativo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Venta', 'CONSTRAINT',
'CK_Venta_PrecioVenta'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'El precio de venta no puede
ser negativo.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Venta', 'CONSTRAINT',
'CK_Venta_PrecioVenta'
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Venta', 'CONSTRAINT', 'CK_Venta_FormaPago'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Limita la forma de pago a
“contado” o “crédito”.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Venta', 'CONSTRAINT',
'CK_Venta_FormaPago'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Limita la forma de pago a
“contado” o “crédito”.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Venta', 'CONSTRAINT',
'CK_Venta_FormaPago'
end
```

GO

```
if exists (select * from ::fn_listextendedproperty ('MS_Description', 'SCHEMA', 'dbo', 'TABLE',
'Tbl_Venta', 'CONSTRAINT', 'CK_Venta_EstadoPago'))
begin
    EXEC sys.sp_updateextendedproperty 'MS_Description', 'Restringe el estado a
“pagado” o “pendiente”.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Venta', 'CONSTRAINT',
'CK_Venta_EstadoPago'
end
else
begin
    EXEC sys.sp_addextendedproperty 'MS_Description', 'Restringe el estado a
“pagado” o “pendiente”.', 'SCHEMA', 'dbo', 'TABLE', 'Tbl_Venta', 'CONSTRAINT',
'CK_Venta_EstadoPago'
end
```

GO





Tbl_Vehiculo


Database table in package 'Tables'








Registra todos los vehículos nuevos o usados disponibles en el inventario de la empresa. Incluye datos técnicos (marca, modelo, año, color, tipo de combustible y transmisión, kilometraje), estado del vehículo (disponible, vendido, reservado), precio y fecha de ingreso. Es la base para las ventas y los servicios realizados.


Tbl_Vehiculo
Version 1.0 Phase 1.0 Proposed
LPT-DELL created on 11/09/2025. Last modified 11/09/2025
DBMS SQL Server 2012

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
 id_vehiculo	int	True	Aquí se almacena el identificador numérico único de cada vehículo registrado en la base de datos. No puede repetirse, debe ser un número entero positivo y no puede quedar vacío. Este campo sirve como clave primaria para identificar cada registro de vehículo.
 numero_chasis	varchar(50)	True	Aquí se guarda el número de chasis único del vehículo. Es un valor de texto (hasta 50 caracteres) que identifica de manera exclusiva al vehículo. No puede quedar vacío ni repetirse en otro vehículo.
 marca	varchar(50)	True	Aquí se guarda la marca del vehículo (por ejemplo, Toyota, Honda, Nissan). Es un texto obligatorio y no puede iniciar con caracteres especiales ni estar vacío.
 modelo	varchar(50)	True	Aquí se almacena el modelo del vehículo (por ejemplo, Corolla, Civic, Frontier). Es un texto obligatorio y no puede iniciar con caracteres especiales ni quedar vacío.
 anio	varchar(50)	True	Aquí se guarda el año de fabricación del vehículo. Debe ser un número entero comprendido entre 1900 y el año actual. No puede quedar vacío.
 color	varchar(30)	True	Aquí se almacena el color del vehículo (por ejemplo, "rojo", "negro metálico"). Es un texto obligatorio y no puede quedar vacío.
 tipo_combustible	varchar(20)	True	Aquí se guarda el tipo de combustible que utiliza el vehículo. Solo se permiten valores predefinidos como "gasolina", "diésel", "eléctrico", "híbrido". Es obligatorio.
 tipo_transmision	varchar(20)	True	Aquí se almacena el tipo de transmisión del vehículo. Solo se permiten valores predefinidos como "automático", "mecánico" o "semiautomático". Es obligatorio.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
 precio	decimal(12,2)	True	Aquí se guarda el precio de venta del vehículo. Es un número decimal (con dos decimales) mayor o igual a cero. No puede quedar vacío.
 estado_vehiculo	varchar(20)	True	Aquí se almacena el estado actual del vehículo dentro del inventario. Solo se permiten valores predefinidos como “disponible”, “vendido” o “reservado”. Es obligatorio.
 fecha_ingreso	date	True	Aquí se guarda la fecha en que el vehículo ingresó al inventario de la empresa. No puede ser mayor a la fecha actual ni quedar vacío.
 kilometraje	int	True	Aquí se almacena el kilometraje actual del vehículo. Es un número entero mayor o igual a 0 y menor o igual a 1,000,000. No puede quedar vacío.

PRIMARY KEY NAME	COLUMNS	COMMENTS
 PK_Vehiculo	id_vehiculo	Identificador único de cada vehículo que ha sido adquirido y registrado por la empresa. Debe ser un valor numérico positivo y exclusivo, garantizando que no existan duplicados. Es fundamental para relacionar el vehículo con ventas, clientes y servicios.

TYPE / NAME	COLUMNS	COMMENTS
 «check» CK_Vehiculo_IdVehiculo	id_vehiculo	Verifica que el ID sea positivo.
 «check» CK_Vehiculo_Anio	anio	Restringe año a valores entre 1900 y el año actual.
 «check» CK_Vehiculo_Precio	precio	No permite precio negativo.
 «check» CK_Vehiculo_Kilometraje	kilometraje	Asegura kilometraje dentro de 0 a 1,000,000.
 «check» CK_Vehiculo_TipoCombustible	tipo_combustible	Solo admite “gasolina”, “diesel”, “eléctrico”, “híbrido”.
 «check» CK_Vehiculo_TipoTransmision	tipo_transmision	Solo admite “automático”, “mecánico”, “semiautomático”.
 «check» CK_Vehiculo_Estado	estado_vehiculo	Limita estado a “disponible”,

		“vendido”, “reservado”.
 «check» CK_Vehiculo_FechaIngreso	fecha_ingreso	Fecha de ingreso no puede ser futura.







Tbl_Cliente


Database table in package 'Tables'




Contiene la información de cada cliente de la empresa, tanto particulares como empresas. Guarda datos de contacto, dirección, ciudad, estado y tipo de cliente, además de la fecha de registro. Permite dar seguimiento a compras y servicios para ofrecer atención personalizada.






Tbl_Cliente
Version 1.0 Phase 1.0 Proposed
LPT-DELL created on 15/09/2025. Last modified 15/09/2025
DBMS SQL Server 2012

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
 id_cliente	int	True	Aquí se almacena el identificador numérico único de cada cliente registrado en la base de datos. No puede repetirse, debe ser un número entero positivo y no puede quedar vacío. Este campo sirve como clave primaria para identificar cada registro de cliente.
 email	varchar(100)	False	Aquí se guarda el correo electrónico principal del cliente. Debe ser único, no puede repetirse entre clientes y debe tener un formato válido de correo electrónico (usuario@dominio). No puede iniciar ni terminar con espacios en blanco.
 nombre	varchar(50)	True	Aquí se almacena el nombre completo del cliente. Debe contener caracteres alfabéticos, no puede quedar vacío ni comenzar con un carácter especial. Puede incluir espacios o guiones después de la primera letra.
 apellido	varchar(50)	True	Aquí se almacena el/los apellidos del cliente. Debe contener caracteres alfabéticos, no puede quedar vacío ni comenzar con un carácter especial. Puede incluir espacios o guiones después de la primera letra.
 telefono	varchar(20)	False	Aquí se almacena el número telefónico de contacto del cliente. Puede ser nulo. Si se ingresa, solo puede contener dígitos, espacios, signos +, - o paréntesis, y debe iniciar con dígito 0 +.
	varchar(255)		Aquí se almacena la dirección física del cliente

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
 direccion		True	(calle, número, referencia). Es obligatoria, no puede quedar vacía y tiene un máximo de 255 caracteres.
 ciudad	varchar(50)	True	Aquí se almacena la ciudad de residencia del cliente. Es obligatoria y no puede quedar vacía ni iniciar con caracteres especiales.
 estado_cliente	varchar(50)	True	Aquí se almacena el estado, provincia o departamento donde reside el cliente. Es obligatorio y no puede quedar vacío.
 codigo_postal	varchar(10)	False	Aquí se almacena el código postal del cliente. Es opcional; si se ingresa, debe contener solo caracteres alfanuméricos, espacios o guiones y tener máximo 10 caracteres.
 fecha_registro	date	True	Aquí se almacena la fecha en que se registró el cliente en el sistema. Es obligatoria, no puede quedar vacía y no puede ser mayor a la fecha actual.
 tipo_cliente	varchar(20)	True	Aquí se almacena el tipo de cliente según su naturaleza. Solo se permiten valores predefinidos como particular o empresa. Es obligatorio y no puede quedar vacío.

PRIMARY KEY NAME	COLUMNS	COMMENTS
 PK_Cliente	id_cliente	Aquí se almacena el identificador numérico único de cada cliente registrado en la base de datos. No puede repetirse, debe ser un número entero positivo y no puede quedar vacío. Este campo sirve como clave primaria para identificar de manera inequívoca a cada cliente, ya sea persona o empresa.

TYPE / NAME	COLUMNS	COMMENTS
 «check» CK_Cliente_IdCliente	id_cliente	Asegura que el identificador sea mayor que 0.
 «check» CK_Cliente_Email_Format	email	Valida que el correo electrónico tenga formato correcto, contenga @ y un dominio válido, y no incluya espacios.
 «check» CK_Cliente_Nombre_IniciaLetra	nombre	Obliga a que el nombre empiece con una letra.




 «check» CK_Cliente_Apellido_IniciaLetra	apellido	Obliga a que el apellido empiece con una letra.
 «check» CK_Cliente_Telefono_Format	telefono	Asegura que el teléfono solo tenga caracteres numéricos y símbolos válidos (+, -, (), /) y longitud máxima de 20.
 «check» CK_Cliente_CodigoPostal_Formato	codigo_postal	Restringe el código postal a caracteres alfanuméricos y longitud máxima de 10.
 «check» CK_Cliente_FechaRegistro	fecha_registro	Garantiza que la fecha de registro no sea posterior a la fecha actual.
 «check» CK_Cliente_TipoCliente	tipo_cliente	Restringe el tipo de cliente a “particular” o “empresa”.








Tbl_Empleado


Database table in package 'Tables'

Almacena los datos del personal de la casa automotriz: nombre, apellido, cargo (vendedor, mecánico, etc.), salario, dirección, fecha de contratación y estado laboral. Sirve para controlar quién realiza ventas, servicios, facturas u órdenes.

Tbl_Empleado
Version 1.0 Phase 1.0 Proposed
LPT-DELL created on 15/09/2025. Last modified 15/09/2025
DBMS SQL Server 2012

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
 id_empleado	int	True	Aquí se almacena el identificador numérico único de cada empleado registrado en la base de datos. No puede repetirse, debe ser un número entero positivo y no puede quedar vacío. Este campo sirve como clave primaria para identificar cada registro de empleado.
 email	varchar(100)	False	Aquí se guarda el correo electrónico del empleado. Debe ser único, no puede repetirse entre empleados y debe tener un formato válido de correo electrónico (usuario@dominio). No puede iniciar ni terminar con espacios en blanco.
 nombre	varchar(50)	True	Aquí se almacena el nombre del empleado. Debe contener caracteres alfabéticos, no puede quedar vacío ni comenzar con un carácter especial. Puede incluir espacios o guiones después de la primera letra.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
 apellido	varchar(50)	True	Aquí se almacena el/los apellidos del empleado. Debe contener caracteres alfabéticos, no puede quedar vacío ni comenzar con un carácter especial. Puede incluir espacios o guiones después de la primera letra.
 cargo	varchar(50)	True	Aquí se almacena el cargo o puesto que ocupa el empleado dentro de la empresa. Solo se permiten valores predefinidos como vendedor, mecánico, etc. Es obligatorio y no puede quedar vacío.
 telefono	varchar(20)	False	Aquí se almacena el número telefónico de contacto del empleado. Puede ser nulo. Si se ingresa, solo puede contener dígitos, espacios, signos +, - o paréntesis, y debe iniciar con dígito 0 +.
 fecha_contratacion	date	True	Aquí se almacena la fecha en que fue contratado el empleado. Es obligatoria, no puede quedar vacía y no puede ser mayor a la fecha actual.
 salario	decimal(12,2)	True	Aquí se almacena el salario mensual del empleado. Debe ser un número decimal mayor o igual a cero y no puede quedar vacío.
 direccion	varchar(255)	True	Aquí se almacena la dirección física del empleado. Es obligatoria, no puede quedar vacía y tiene un máximo de 255 caracteres.
 estado_empleado	varchar(20)	True	Aquí se almacena el estado laboral actual del empleado. Solo se permiten valores predefinidos como activo o inactivo. Es obligatorio y no puede quedar vacío.

PRIMARY KEY NAME	COLUMNS	COMMENTS
 PK_Empleado	id_empleado	Representa el identificador único de cada empleado de la empresa. Es un valor numérico entero, no puede estar vacío ni repetirse. Permite diferenciar a cada empleado de forma precisa para tareas de gestión, control de nómina y asignación de responsabilidades.

TYPE / NAME	COLUMNS	COMMENTS

«check» CK_Empleado_IdEmpleado	id_empleado	Asegura que el ID sea positivo.
«check» CK_Empleado_Email_Format	email	Valida que el correo tenga formato correcto y sin espacios.
«check» CK_Empleado_Nombre_IniciaLetra	nombre	Obliga a que el nombre inicie con letra.
«check» CK_Empleado_Apellido_IniciaLetra	apellido	Obliga a que el apellido inicie con letra.
«check» CK_Empleado_Cargo_Valores	cargo	Restringe los cargos a “vendedor”, “mecánico”, “gerente” o “administrativo”.
«check» CK_Empleado_Telefono_Format	telefono	Valida que el teléfono tenga solo caracteres permitidos y longitud = 20.
«check» CK_Empleado_FechaContratacion	fecha_contratacion	Garantiza que la fecha de contratación no sea futura.
«check» CK_Empleado_Salario	salario	Asegura que el salario no sea negativo.
«check» CK_Empleado_EstadoValores	estado_empleado	Restringe estado del empleado a “activo” o “inactivo”.








Tbl_Venta


Database table in package 'Tables'

Registra cada transacción de venta de vehículos. Vincula al vehículo vendido, al cliente comprador y al empleado que realizó la venta. Incluye precio, fecha, forma y estado de pago, así como observaciones de la operación.

Tbl_Venta
Version 1.0 Phase 1.0 Proposed
LPT-DELL created on 15/09/2025. Last modified 15/09/2025
DBMS SQL Server 2012

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id_venta	int	True	Aquí se almacena el identificador numérico único de cada venta registrada en la base de datos. No puede repetirse, debe ser un número entero positivo y no puede quedar vacío. Este campo sirve como clave primaria para identificar cada registro de venta.
id_vehiculo	int	True	Aquí se almacena el identificador del vehículo vendido.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
			Es un dato obligatorio y corresponde a una clave foránea que enlaza con la tabla Vehículo. Debe existir previamente en la tabla Vehículo.
 id_cliente	int	True	Aquí se almacena el identificador del cliente que realizó la compra. Es un dato obligatorio y corresponde a una clave foránea que enlaza con la tabla Cliente. Debe existir previamente en la tabla Cliente.
 id_empleado	int	True	Aquí se almacena el identificador del empleado que realizó la venta. Es un dato obligatorio y corresponde a una clave foránea que enlaza con la tabla Empleado. Debe existir previamente en la tabla Empleado.
 fecha_venta	date	True	Aquí se almacena la fecha en que se realizó la venta. Es obligatoria, no puede quedar vacía y no puede ser mayor a la fecha actual.
 precio_venta	decimal(12,2)	True	Aquí se almacena el precio final de venta del vehículo. Debe ser un número decimal mayor o igual a cero y no puede quedar vacío.
 forma_pago	varchar(20)	True	Aquí se almacena la forma de pago acordada con el cliente. Solo se permiten valores predefinidos como contado o crédito. Es obligatoria y no puede quedar vacía.
 estado_pago	varchar(20)	True	Aquí se almacena el estado actual del pago de la venta. Solo se permiten valores predefinidos como pagado o pendiente. Es obligatorio y no puede quedar vacío.
 observaciones	varchar(255)	False	Aquí se almacena cualquier comentario adicional relacionado con la venta. Este campo es opcional y puede quedar vacío.

PRIMARY KEY NAME	COLUMNS	COMMENTS
 PK_Venta	id_venta	Identificador único de cada transacción de venta registrada. No puede repetirse, debe ser numérico positivo y obligatorio. Permite relacionar la venta con cliente, empleado y vehículo vendido.

TYPE / NAME	COLUMNS	COMMENTS

«index» IXFK_Tbl_Venta_Tbl_Cliente	id_cliente	Mejora consultas que muestran las ventas realizadas a un cliente.
«index» IXFK_Tbl_Venta_Tbl_Empleado	id_empleado	Facilita búsquedas de ventas gestionadas por un empleado específico.
«index» IXFK_Tbl_Venta_Tbl_Vehiculo	id_vehiculo	Optimiza consultas que relacionan un vehículo con la venta correspondiente.
«check» CK_Venta_IdVenta	id_venta	Asegura que el ID sea positivo.
«check» CK_Venta_FechaVenta	fecha_venta	No permite ventas con fecha futura.
«check» CK_Venta_PrecioVenta	precio_venta	El precio de venta no puede ser negativo.
«check» CK_Venta_FormaPago	forma_pago	Limita la forma de pago a “contado” o “crédito”.
«check» CK_Venta_EstadoPago	estado_pago	Restringe el estado a “pagado” o “pendiente”.

FOREIGN KEY NAME	COLUMNS	REFERENCES
FK_Tbl_Venta_Tbl_Cliente	id_cliente	Tbl_Cliente(id_cliente)
FK_Tbl_Venta_Tbl_Empleado	id_empleado	Tbl_Empleado(id_empleado)
FK_Tbl_Venta_Tbl_Vehiculo	id_vehiculo	Tbl_Vehiculo(id_vehiculo)

Tbl_Repuesto


Database table in package 'Tables'


Gestiona el inventario de repuestos. Contiene código único, nombre, descripción, categoría, precio unitario, stock, ubicación en almacén, estado (disponible o agotado) y proveedor. Permite conocer la disponibilidad para servicios y órdenes de compra.









Tbl_Repuesto
Version 1.0 Phase 1.0 Proposed
LPT-DELL created on 15/09/2025. Last modified 15/09/2025
DBMS SQL Server 2012


COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id_vehiculorepuesto	int	True	Aquí se almacena el identificador numérico

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
			único de cada repuesto registrado en la base de datos. No puede repetirse, debe ser un número entero positivo y no puede quedar vacío. Este campo sirve como clave primaria para identificar cada repuesto.
 codigo_repuesto	varchar(50)	True	Aquí se almacena el código único asignado a cada repuesto. No puede repetirse entre repuestos, debe ser obligatorio y sirve como referencia de inventario.
 nombre	varchar(100)	True	Aquí se almacena el nombre del repuesto. Debe contener solo caracteres alfabéticos y espacios, no puede quedar vacío ni comenzar con un carácter especial.
 descripcion	varchar(255)	False	Aquí se almacena una breve descripción del repuesto. Es opcional, puede quedar vacío, pero si se ingresa debe tener un máximo de 255 caracteres.
 precio_unitario	decimal(10,2)	True	Aquí se almacena el precio de venta por unidad del repuesto. Debe ser un número decimal mayor o igual a cero y no puede quedar vacío.
 stock	int	True	Aquí se almacena la cantidad disponible del repuesto en inventario. Debe ser un número entero mayor o igual a cero y no puede quedar vacío.
 id_proveedor	int	True	Aquí se almacena el identificador del proveedor asociado al repuesto. Es un dato obligatorio y corresponde a una clave foránea que enlaza con la tabla Proveedor. Debe existir previamente en la tabla Proveedor.
 fecha_ingreso	date	True	Aquí se almacena la fecha en que el repuesto ingresó al almacén o al inventario de la empresa. Debe ingresarse obligatoriamente (NO puede quedar vacío), debe ser una fecha válida y no puede ser una fecha futura (es decir, = la fecha actual).
 ubicacion_almacen	varchar(50)	True	Aquí se almacena la ubicación física del repuesto dentro del almacén. Es opcional, puede quedar vacío, y se usará para identificar el estante o pasillo correspondiente.
 categoria	varchar(50)	True	Aquí se almacena la categoría a la que pertenece el repuesto (ejemplo: motor, suspensión, eléctrico). Es obligatoria y debe contener únicamente valores predefinidos.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
 estado_repuesto	varchar(20)	True	Aquí se almacena el estado actual del repuesto en el inventario. Solo se permiten valores predefinidos como disponible o agotado. Es obligatorio y no puede quedar vacío.

PRIMARY KEY NAME	COLUMNS	COMMENTS
 PK_Tbl_Repuesto	id_vehiculorepuesto	Identificador único de cada repuesto almacenado en el inventario. Es un valor numérico entero, positivo y exclusivo. Su función es facilitar la gestión de repuestos y garantizar la trazabilidad en órdenes de compra, ventas o servicios.

TYPE / NAME	COLUMNS	COMMENTS
 «index» IXFK_Tbl_Repuesto_Tbl_Proveedor	id_proveedor	Optimiza consultas que listan repuestos según el proveedor que los suministra.
 «check» CK_Repuesto_IdRepuesto	id_vehiculorepuesto	Verifica que el ID sea positivo.
 «check» CK_Repuesto_Nombre_IniciaLetra	nombre	El nombre debe comenzar con letra.
 «check» CK_Repuesto_Categoria	categoria	Limita categoría a “motor”, “suspensión”, “eléctrico”, “carrocería”.
 «check» CK_Repuesto_PrecioUnitario	precio_unitario	No permite precios negativos.
 «check» CK_Repuesto_Stock	stock	Asegura stock no negativo.
 «check» CK_Repuesto_FechaIngreso	fecha_ingreso	Fecha de ingreso debe ser actual o pasada.
 «check» CK_Repuesto_Estado	estado_repuesto	Solo admite “disponible” o “agotado”.


FOREIGN KEY NAME	COLUMNS	REFERENCES
 FK_Tbl_Repuesto_Tbl_Proveedor	id_proveedor	Tbl_Proveedor(id_proveedor)



Tbl_Servicio


Database table in package 'Tables'









Define los tipos de servicios que ofrece la empresa (mantenimiento o reparación). Incluye descripción, costo, duración, garantía, si requiere repuestos, nivel de dificultad, fecha de creación y observaciones. Es el catálogo de servicios que luego se aplican a los vehículos.

Tbl_Servicio
Version 1.0 Phase 1.0 Proposed
LPT-DELL created on 15/09/2025. Last modified 15/09/2025
DBMS SQL Server 2012

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
 id_servicio	int	True	Aquí se almacena el identificador numérico único de cada servicio registrado en la base de datos. No puede repetirse, debe ser un número entero positivo y no puede quedar vacío. Este campo sirve como clave primaria para identificar cada registro de servicio.
 nombre	varchar(50)	True	Aquí se almacena el nombre del servicio ofrecido por la empresa (por ejemplo: "Cambio de aceite", "Alineación y balanceo"). Debe contener caracteres alfabéticos, no puede quedar vacío ni iniciar con un carácter especial.
 descripcion	varchar(255)	False	Aquí se almacena una descripción breve y clara del servicio. Es opcional, puede quedar vacío, pero si se ingresa debe tener un máximo de 255 caracteres.
 costo	decimal(10,2)	True	Aquí se almacena el costo base del servicio. Debe ser un número decimal mayor o igual a cero y no puede quedar vacío.
 duracion_horas	int	True	Aquí se almacena la duración aproximada del servicio expresada en minutos. Debe ser un número entero mayor que cero y no puede quedar vacío.
 garantia_meses	int	True	Aquí se almacena el tiempo de garantía que ofrece el servicio (en meses). Debe ser un número entero mayor o igual a cero. Según el tipo de servicio.
 requiere_repuesto	varchar(10)	True	Aquí se indica si el servicio requiere repuestos para ser realizado. Solo se permiten valores predefinidos: Sí o No. Es obligatorio.
 nivel_dificultad	varchar(50)	True	Aquí se almacena el nivel de dificultad del servicio. Solo se permiten valores predefinidos como básico, medio o avanzado. Es obligatorio.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
 fecha_creacion	date	True	Aquí se almacena la fecha en que el servicio fue creado en el sistema. Es obligatoria, no puede quedar vacía y no puede ser mayor a la fecha actual.
 observaciones	varchar(255)	False	Aquí se almacena cualquier comentario adicional sobre el servicio. Este campo es opcional y puede quedar vacío.

PRIMARY KEY NAME	COLUMNS	COMMENTS
 PK_Tbl_Servicio	id_servicio	Representa el código único de cada servicio ofrecido por la empresa. Es un número entero positivo, obligatorio y no repetido. Permite catalogar los diferentes tipos de servicios de mantenimiento o reparación.

TYPE / NAME	COLUMNS	COMMENTS
 «check» CK_Servicio_IdServicio	id_servicio	Asegura que el ID sea positivo.
 «check» CK_Servicio_Nombre_IniciaLetra	nombre	El nombre debe comenzar con letra.
 «check» CK_Servicio_Costo	costo	No permite costos negativos.
 «check» CK_Servicio_Duracion	duracion_horas	Obliga a que la duración sea mayor que 0.
 «check» CK_Servicio_Garantia	garantia_meses	Asegura que los meses de garantía no sean negativos.
 «check» CK_Servicio_RequiereRepuesto	requiere_repuesto	Solo admite valores “Sí” o “No”.
 «check» CK_Servicio_NivelDificultad	nivel_dificultad	Limita a “básico”, “medio”, “avanzado”.
 «check» CK_Servicio_FechaCreacion	fecha_creacion	No puede superar la fecha actual.

Tbl_ServicioRealizado




Database table in package 'Tables'


Aquí se registran todos los servicios efectivamente realizados a vehículos. Se vincula con la tabla Servicio (qué tipo de servicio), con Empleado (quién lo realizó), y permite relacionarse tanto con un vehículo propio de inventario (Vehículo) como con un vehículo externo (VehículoExterno).







De esta forma, cualquier servicio queda ligado a un vehículo concreto, sin obligar a que este pertenezca al inventario de ventas.

Tbl_ServicioRealizado
Version 1.0 Phase 1.0 Proposed
LPT-DELL created on 16/09/2025. Last modified 16/09/2025
DBMS SQL Server 2012

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
 id_servicio_realizado	int	True	Aquí se almacena el identificador numérico único de cada servicio realizado en el taller. Es la clave primaria de la tabla y permite distinguir de manera única cada registro de servicio. Debe ser un número entero positivo mayor que cero y no puede quedar vacío.
 id_empleado	int	True	Aquí se almacena el identificador del empleado encargado de realizar el servicio. Es obligatorio y funciona como una clave foránea que enlaza con la tabla Empleado, garantizando que solo empleados registrados puedan ser asignados a un servicio
 fecha_servicio	date	True	Aquí se almacena la fecha exacta en la que se realizó el servicio. Debe ingresarse en formato de fecha válido, no puede quedar vacío y no puede ser posterior a la fecha actual.
 id_repuesto	int	False	Aquí se almacena el identificador del repuesto utilizado en el servicio, en caso de que se requiera. Es una clave foránea que referencia la tabla Repuesto. Este campo puede quedar vacío si no se utilizó ningún repuesto.
 cantidad_repuesto	int	False	Aquí se almacena la cantidad de unidades de repuesto utilizadas en el servicio. Debe ser un número entero mayor o igual a cero. Es opcional, pero si existe un id_repuesto, este campo debe contener un valor mayor a cero.
 costo_total	decimal(12,2)	True	Aquí se almacena el costo total del servicio realizado, incluyendo repuestos y mano de obra. Debe ser un número decimal mayor o igual a cero y no puede quedar vacío.
 estado_servicio	varchar(20)	True	Aquí se almacena el estado actual del servicio. Solo se permiten valores predefinidos como pendiente o finalizado. Es obligatorio y no puede quedar vacío.
 observaciones	varchar(20)	False	Aquí se almacenan comentarios adicionales relacionados con el servicio realizado (ejemplo: detalles del diagnóstico, recomendaciones al

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
			cliente). Es un campo opcional y puede quedar vacío.
 id_cita_servicio	int	True	FK que relaciona el servicio realizado con la cita que lo originó, es obligatoria y proviene de la tabla Tbl_CitaServicio
 id_respuesto	int	False	
 id_cita	int	True	Funciona como FK, relacionando la tabla ServicioRealizado con la cita que lo originó, ubicada en la Tbl_CitaServicio

PRIMARY KEY NAME	COLUMNS	COMMENTS
 PK_Tbl_ServicioRealizado	id_servicio_realizado	Identificador único de cada registro de servicio efectivamente realizado a un vehículo (interno o externo). Es un número entero positivo, no repetido y obligatorio. Sirve como referencia principal para detallar los servicios ejecutados.

TYPE / NAME	COLUMNS	COMMENTS
 «index» IXFK_Tbl_ServicioRealizado_Tbl_CitaServicio	id_cita	Se utiliza para optimizar la búsqueda y validación de registros de Tbl_ServicioRealizado en función de la cita asociada en Tbl_CitaServicio. Facilita consultas frecuentes como “obtener todos los servicios realizados a partir de una cita específica” y mejora el rendimiento de las operaciones entre ambas tablas.
 «index» IXFK_Tbl_ServicioRealizado_Tbl_Empleado	id_empleado	Facilita consultas de servicios realizados por empleado.
 «index» IXFK_Tbl_ServicioRealizado_Tbl_Repuesto	id_respuesto	Acelera búsquedas de servicios en los que se usaron repuestos.
 «check» CK_ServReal_Id	id_servicio_realizado	Asegura que el ID sea positivo.
 «check» CK_ServReal_Fecha	fecha_servicio	No permite fechas futuras para el servicio.
 «check» CK_ServReal_Cantidad	cantidad_repuesto	La cantidad de repuestos debe ser = 0 o nula.

«check» CK_ServReal_Costo	costo_total	El costo total no puede ser negativo.
«check» CK_ServReal_Estado	estado_servicio	Limita estado a “pendiente” o “finalizado”.

FOREIGN KEY NAME	COLUMNS	REFERENCES
FK_Tbl_ServicioRealizado_Tbl_CitaServicio	id_cita	Tbl_CitaServicio(id_cita)
FK_Tbl_ServicioRealizado_Tbl_Empleado	id_empleado	Tbl_Empleado(id_empleado)
FK_Tbl_ServicioRealizado_Tbl_Repuesto	id_respuesto	Tbl_Repuesto(id_vehiculorepuesto)


Tbl_VehiculoExterno


Database table in package 'Tables'




Aquí se registran los vehículos que no forman parte del inventario de ventas, pero que ingresan al taller para recibir mantenimiento o reparación. Contiene solo los datos básicos necesarios para identificar al vehículo y dar seguimiento al servicio, sin requerir información de precio, número de chasis u otros datos exclusivos de los vehículos de inventario.


Tbl_VehiculoExterno
Version 1.0 Phase 1.0 Proposed
LPT-DELL created on 16/09/2025. Last modified 16/09/2025
DBMS SQL Server 2012

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
id_vehiculo_externo	int	True	Aquí se almacena el identificador numérico único de cada vehículo externo registrado en la base de datos. Este campo sirve como clave primaria y permite diferenciar de manera inequívoca a cada vehículo externo. Debe ser un número entero positivo mayor que cero y no puede quedar vacío.
placa	varchar(50)	True	Aquí se almacena el número de placa oficial del vehículo externo. Este valor identifica de forma única al vehículo y se utiliza como referencia principal para búsquedas y validaciones. No puede quedar vacío, no debe contener caracteres especiales fuera del formato de placa, y debe ser único para evitar duplicados en la base de datos.
marca	varchar(50)	True	Aquí se almacena la marca comercial del vehículo externo, como por ejemplo Toyota, Nissan, Honda, etc. Debe ingresarse siempre con caracteres alfabéticos, puede incluir espacios, pero no puede iniciar con un carácter especial ni con un

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
			número. Este campo es obligatorio.
 modelo	varchar(50)	True	Aquí se almacena el modelo específico del vehículo externo, como Corolla, Civic, Hilux, entre otros. Debe ser un valor de texto obligatorio, no puede quedar vacío y debe permitir la combinación de letras y números, aunque no puede iniciar con un carácter especial.
 año	int	True	Aquí se almacena el año de fabricación del vehículo externo. Debe ser un número entero de cuatro dígitos comprendido entre 1900 y el año actual, para garantizar la validez del dato. No puede quedar vacío.
 color	varchar(50)	True	Aquí se almacena el color principal del vehículo externo (ejemplo: negro, rojo metálico, blanco perla). Es un valor obligatorio, no puede quedar vacío y debe contener únicamente letras y, opcionalmente, combinaciones como “gris claro” o “azul oscuro”.
 propietario	varchar(100)	True	Aquí se almacena el nombre completo del propietario del vehículo externo. Debe contener únicamente caracteres alfabéticos, puede incluir espacios y guiones, pero no puede iniciar con un carácter especial ni con un número. Es obligatorio y no puede quedar vacío.

PRIMARY KEY NAME	COLUMNS	COMMENTS
 PK_Tbl_VehiculoExterno	id_vehiculo_externo	Clave primaria que distingue de forma única a cada vehículo externo (no adquirido en la empresa) que recibe servicios. Es un número entero positivo, obligatorio y no repetido. Permite llevar control de los servicios realizados a clientes externos.

TYPE / NAME	COLUMNS	COMMENTS
 «check» CK_VehiculoExterno_Id	id_vehiculo_externo	Verifica que el ID sea positivo.
 «check» CK_VehiculoExterno_Placa	placa	Valida que la placa contenga solo letras mayúsculas y números.
 «check» CK_VehiculoExterno_Año	año	Restringe año a =1900 y = año actual.

 «check» CK_VehiculoExterno_Propietario	propietario	Obliga a que el propietario inicie con una letra.
--	-------------	---

Tbl_Proveedor







Database table in package 'Tables'


Proveedor almacena la información de las empresas o personas que suministran repuestos, materiales o servicios a la casa automotriz.












Esta tabla permite identificar de forma única a cada proveedor, disponer de sus datos de contacto, ubicación y clasificación (local / internacional), y servir como referencia en las compras, órdenes y repuestos.

Tbl_Proveedor
Version 1.0 Phase 1.0 Proposed
LPT-DELL created on 16/09/2025. Last modified 16/09/2025
DBMS SQL Server 2012

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
 id_proveedor	int	True	Aquí se almacena el identificador numérico único de cada proveedor registrado en la base de datos. No puede repetirse, debe ser un número entero positivo mayor que cero y no puede quedar vacío. Este campo sirve como clave primaria para identificar de forma inequívoca cada proveedor.
 nombre	varchar(150)	True	Aquí se guarda el nombre oficial o razón social del proveedor. Es un dato obligatorio que no puede quedar vacío ni contener únicamente espacios, ya que se utiliza en reportes, búsquedas y facturación para reconocer de forma clara a cada proveedor.
 telefono	varchar(20)	False	Aquí se almacena el número telefónico de contacto del proveedor. Este campo es opcional, pero si se ingresa debe respetar el formato habitual de números telefónicos, permitiendo dígitos, guiones, paréntesis y el signo más. Sirve como un medio directo de comunicación con el proveedor.
 email	varchar(100)	False	Aquí se guarda la dirección de correo electrónico principal del proveedor. Es un campo opcional, pero cuando se proporciona debe tener un formato válido de correo electrónico y no puede repetirse, ya que se utilizará como un medio de contacto formal y para el envío de notificaciones electrónicas.
 direccion	varchar(255)	True	Aquí se almacena la dirección física completa del proveedor, como calle, número de casa o local, barrio o colonia. Es un dato obligatorio que no puede quedar vacío, ya que resulta indispensable para la localización geográfica, envíos de mercancía y registros oficiales.

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
 ciudad	varchar(50)	True	Aquí se guarda el nombre de la ciudad donde se ubica el proveedor. Este campo es obligatorio, no puede quedar vacío y permite clasificar a los proveedores según su ubicación geográfica para fines de reportes o segmentación.
 estado_proveedor	varchar(50)	True	Aquí se almacena el nombre del estado, provincia o departamento al que pertenece la ciudad donde se ubica el proveedor. Este campo es obligatorio y debe registrarse siempre, ya que facilita la organización geográfica y la generación de informes por región.
 codigo_postal	varchar(10)	False	Aquí se almacena el código postal correspondiente a la dirección del proveedor. Este campo es opcional, pero cuando se proporciona debe cumplir con la longitud establecida y contener únicamente caracteres válidos como letras, números y guiones, ya que se utiliza para fines de envío y normalización de direcciones.
 contacto_principal	varchar(100)	True	Aquí se almacena el nombre completo de la persona de contacto principal dentro del proveedor. Es un dato obligatorio que no puede quedar vacío, pues resulta esencial para las gestiones comerciales, consultas y negociaciones directas.
 fecha_registro	date	True	Aquí se almacena la fecha en la que el proveedor fue ingresado en la base de datos. Es un campo obligatorio que no puede quedar vacío ni registrar una fecha posterior a la actual, ya que se utiliza con fines de auditoría y control histórico.
 tipo_proveedor	varchar(50)	True	Aquí se almacena la clasificación del proveedor según su origen, diferenciando si es de tipo local o internacional. Este dato es obligatorio y únicamente puede tomar uno de estos dos valores, lo que permite aplicar políticas de compra, impuestos y logística según corresponda.

PRIMARY KEY NAME	COLUMNS	COMMENTS
 PK_Tbl_Proveedor	id_proveedor	Aquí se almacena el identificador numérico único de cada proveedor registrado en la base de datos. No puede repetirse, debe ser un número entero positivo mayor que cero y no puede quedar vacío. Este campo sirve como clave primaria para identificar de forma inequívoca cada proveedor.


TYPE / NAME	COLUMNS	COMMENTS
 «check» CK_Proveedor_IdProveedor	id_proveedor	Asegura que sea positivo.
 «check» CK_Proveedor_Nombre	nombre	El nombre no puede estar vacío.
 «check» CK_Proveedor_Telefono	telefono	alida formato de teléfono y longitud = 20.
 «check» CK_Proveedor_EmailFormato	email	Valida estructura mínima de correo.
 «check» CK_Proveedor_Direccion	direccion	La dirección no puede ser vacía.
 «check» CK_Proveedor_Ciudad	ciudad	La ciudad no puede ser vacía.
 «check» CK_Proveedor_EstadoProveedor	estado_proveedor	El estado no puede ser vacío.
 «check» CK_Proveedor_CodigoPostal	codigo_postal	Restringe longitud y caracteres del código postal.
 «check» CK_Proveedor_ContactoPrincipal	contacto_principal	El contacto principal no puede ser vacío.
 «check» CK_Proveedor_FechaRegistro	fecha_registro	No permite fecha futura.
 «check» CK_Proveedor_TipoProveedor	tipo_proveedor	Solo admite “local” o “internacional”.



Tbl_Factura

Database table in package 'Tables'


Contiene los documentos de facturación emitidos por ventas. Se asocia a la venta y al empleado que la generó. Incluye fecha de emisión, monto total, estado y método de pago, número de recibo, observaciones y fecha de vencimiento para controlar cobros.










Tbl_Factura
Version 1.0 Phase 1.0 Proposed
LPT-DELL created on 16/09/2025. Last modified 16/09/2025
DBMS SQL Server 2012



COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
 id_factura	int	True	Identificador único de cada factura emitida por la empresa. Es un número entero positivo, obligatorio y exclusivo. Garantiza la integridad en el control de facturación y permite asociar la

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
			factura con su respectiva venta y empleado.
 id_venta	int	True	Aquí se guarda el identificador de la venta asociada a la factura. Es obligatorio, no puede quedar vacío y funciona como clave foránea para asegurar que toda factura esté vinculada a una venta existente en la base de datos.
 fecha_emision	date	True	Aquí se almacena la fecha en la que se emitió la factura. Es obligatoria, no puede quedar vacía y no puede ser posterior a la fecha actual para garantizar que refleje un evento real.
 monto_total	decimal(12,2)	True	Aquí se guarda el monto total facturado. Es obligatorio, no puede quedar vacío y debe ser mayor o igual a cero, ya que representa la suma de los conceptos incluidos en la factura.
 estado_pago	varchar(20)	True	Aquí se almacena el estado actual del pago de la factura. Es obligatorio y solo puede tomar uno de los valores predefinidos (“pagado”, “pendiente”), permitiendo controlar el estado financiero.
 metodo_pago	varchar(30)	False	Aquí se guarda el método de pago utilizado para la factura. Es obligatorio únicamente cuando el estado de pago es “pagado”; si la factura está pendiente este campo puede quedar vacío. Solo puede tomar valores predefinidos como “efectivo” o “tarjeta”.
 numero_recibo	varchar(50)	False	Aquí se almacena el número único de recibo asociado a la factura. Es obligatorio únicamente cuando el estado de pago es “pagado”; si la factura está pendiente este campo puede quedar vacío. No puede repetirse para facturas pagadas.
 fecha_vencimiento	date	True	Aquí se almacena la fecha límite para el pago de la factura. Es obligatoria, no puede quedar vacía y debe ser igual o posterior a la fecha de emisión para asegurar coherencia en los plazos.
 id_empleado	int	True	Aquí se guarda el identificador del empleado que generó la factura. Es obligatorio, no puede quedar vacío y funciona como clave foránea para asegurar que cada factura esté asociada a un empleado registrado en el sistema.
 observaciones	varchar(255)	False	Aquí se pueden almacenar comentarios adicionales relacionados con la factura. Es un campo opcional y puede quedar vacío.

PRIMARY KEY NAME	COLUMNS	COMMENTS

 PK_Tbl_Factura	id_factura	Código único de cada factura emitida por la empresa. Es un número entero positivo, obligatorio y exclusivo. Garantiza la integridad en el control de facturación y permite asociar la factura con su respectiva venta y empleado.
--	------------	---

TYPE / NAME	COLUMNS	COMMENTS
 «index» IXFK_Tbl_Factura_Tbl_Empleado	id_empleado	Optimiza consultas que relacionan facturas con el empleado responsable.
 «index» IXFK_Tbl_Factura_Tbl_Venta	id_venta	Permite búsquedas rápidas y joins entre facturas y sus ventas asociadas.
 «check» CK_Factura_IdFactura	id_factura	Verifica que el ID de factura sea positivo.
 «check» CK_Factura_FechaEmision	fecha_emision	Asegura que la fecha de emisión no supere la fecha actual.
 «check» CK_Factura_MontoTotal	monto_total	No permite montos negativos.
 «check» CK_Factura_EstadoPago	estado_pago	Solo admite estados “pendiente” o “pagado”.
 «check» CK_Factura_MetodoPago	metodo_pago	Valida que el método sea nulo (si pendiente) o válido (“efectivo”, “tarjeta”).
 «check» CK_Factura_FechaVencimiento	fecha_vencimiento	Requiere que la fecha de vencimiento sea posterior o igual a la de emisión.
 «check» CK_Factura_PagoDatos	estado_pago, metodo_pago, numero_recibo	Verifica consistencia: si está pendiente, no puede tener método/recibo; si está pagada, si debe tenerlos.

FOREIGN KEY NAME	COLUMNS	REFERENCES
 FK_Tbl_Factura_Tbl_Venta	id_venta	Tbl_Venta(id_venta)
 FK_Tbl_Factura_Tbl_Empleado	id_empleado	Tbl_Empleado(id_empleado)



Tbl_OrdenCompra


Database table in package 'Tables'








Registra las órdenes de compra que la empresa realiza a proveedores. Contiene la fecha, estado (pendiente o recibida), monto total, método de pago, forma de envío, observaciones y el empleado que la gestionó. Se vincula al proveedor para controlar abastecimiento.



Tbl_OrdenCompra
Version 1.0 Phase 1.0 Proposed
LPT-DELL created on 16/09/2025. Last modified 16/09/2025
DBMS SQL Server 2012



COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
 id_orden_compra	int	True	Identificador único de cada orden de compra realizada a proveedores. No se puede repetir, debe ser entero positivo y no nulo. Permite rastrear las órdenes, asociarlas con proveedores y empleados responsables.
 id_proveedor	int	True	Aquí se guarda el identificador del proveedor al que se realiza la orden de compra. Es un dato obligatorio que no puede quedar vacío y funciona como clave foránea para garantizar que toda orden esté vinculada a un proveedor existente en la base de datos.
 fecha_orden	date	True	Aquí se almacena la fecha en la que se generó la orden de compra. Es un campo obligatorio que no puede quedar vacío ni ser posterior a la fecha actual, ya que debe reflejar el momento real en que se hizo el pedido.
 fecha_entrega	date	False	Aquí se guarda la fecha estimada o real de entrega de la orden de compra. Es un campo opcional que puede quedar vacío mientras no se tenga confirmación, pero si se ingresa debe ser igual o posterior a la fecha de orden para mantener coherencia temporal.
 estado_orden	varchar(20)	True	Aquí se almacena el estado actual de la orden de compra. Es un campo obligatorio que solo puede tomar los valores permitidos ("pendiente" o "recibida"), para garantizar un control uniforme del avance de las órdenes.
 total_orden	decimal(12,2)	True	Aquí se guarda el monto total de la orden de compra. Es obligatorio, no puede quedar vacío y debe ser mayor o igual a cero, ya que representa el total comprometido para esa orden.
 observaciones	varchar(255)	False	Aquí se pueden registrar comentarios adicionales relacionados con la orden de compra. Es un campo opcional y puede quedar vacío.
 id_empleado	int	True	Aquí se guarda el identificador del empleado que

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
			gestionó la orden de compra. Es un campo obligatorio que no puede quedar vacío y funciona como clave foránea para asegurar que cada orden esté asociada a un empleado registrado en el sistema.
 metodo_pago	varchar(30)	True	Aquí se almacena el método de pago acordado para la orden de compra. Es un dato obligatorio y solo puede tomar los valores permitidos (“efectivo” o “transferencia”), para clasificar y auditar correctamente los pagos.
 forma_envio	varchar(50)	True	Aquí se guarda la forma de envío de la orden de compra. Es un dato obligatorio que solo puede tomar los valores permitidos (“terrestre”, “aéreo” o “marítimo”), para reflejar el medio logístico utilizado.

PRIMARY KEY NAME	COLUMNS	COMMENTS
 PK_Tbl_OrdenCompra	id_orden_compra	Identificador único de cada orden de compra realizada a proveedores. No se puede repetir, debe ser entero positivo y no nulo. Permite rastrear las órdenes, asociarlas con proveedores y empleados responsables.

TYPE / NAME	COLUMNS	COMMENTS
 «index» IXFK_Tbl_OrdenCompra_Tbl_Empleado	id_empleado	Acelera consultas que muestran órdenes de compra realizadas por un empleado específico.
 «index» IXFK_Tbl_OrdenCompra_Tbl_Proveedor	id_proveedor	Mejora el rendimiento de consultas que relacionan órdenes con proveedores.
 «check» CK_OrdenCompra_IdOrdenCompra	id_orden_compra	Asegura que el ID sea positivo.
 «check» CK_OrdenCompra_FechaOrden	fecha_orden	La fecha de orden no puede ser futura.
 «check» CK_OrdenCompra_FechaEntrega	fecha_entrega	La entrega debe ser nula o posterior a la fecha de orden.
 «check» CK_OrdenCompra_EstadoOrden	estado_orden	Restringe a “pendiente” o “recibida”.
 «check» CK_OrdenCompra_TotalOrden	total_orden	El total no puede ser negativo.

 «check» CK_OrdenCompra_MetodoPago	metodo_pago	Restringido a “efectivo” o “transferencia”.
 «check» CK_OrdenCompra_FormaEnvio	forma_envio	Solo admite “terrestre”, “aéreo” o “marítimo”.






FOREIGN KEY NAME	COLUMNS	REFERENCES
 FK_Tbl_OrdenCompra_Tbl_Empleado	id_empleado	Tbl_Empleado(id_empleado)
 FK_Tbl_OrdenCompra_Tbl_Proveedor	id_proveedor	Tbl_Proveedor(id_proveedor)



Tbl_DetalleOrdenCompra


Database table in package 'Tables'







Especifica los repuestos solicitados en cada orden de compra. Indica cantidad, precio unitario y subtotal de cada ítem. Se vincula tanto a la orden de compra como al repuesto para llevar un control detallado del pedido.



Tbl_DetalleOrdenCompra
Version 1.0 Phase 1.0 Proposed
LPT-DELL created on 16/09/2025. Last modified 16/09/2025
DBMS SQL Server 2012

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
 id_detalle_orden_compra	int	True	Representa el identificador único de cada línea de detalle de una orden de compra. Es un número entero positivo, obligatorio y único. Permite diferenciar cada producto o repuesto dentro de una orden de compra.
 id_orden_compra	int	True	Aquí se guarda el identificador de la orden de compra a la que pertenece este detalle. Es un dato obligatorio que no puede quedar vacío y funciona como clave foránea para asegurar que cada línea de detalle esté vinculada a una orden de compra existente en la base de datos.
 id_repuesto	int	True	Aquí se almacena el identificador del repuesto solicitado en la orden de compra. Es obligatorio, no puede quedar vacío y funciona como clave foránea para garantizar que solo se registren repuestos existentes en la base de datos.
 cantidad	int	True	Aquí se guarda la cantidad del repuesto solicitado en esta línea de detalle. Es un campo obligatorio, no puede quedar vacío y debe ser un número entero mayor que cero, ya que representa la cantidad a comprar.
 precio_unitario	decimal(10,2)	True	Aquí se almacena el precio unitario acordado para cada unidad del repuesto en esta línea de

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
			detalle. Es obligatorio, no puede quedar vacío y debe ser un número decimal mayor o igual a cero.
 subtotal	decimal(12,2)	True	Aquí se almacena el importe total de la línea de detalle calculado como la multiplicación entre la cantidad solicitada y el precio unitario de ese repuesto; no puede quedar vacío y debe ser mayor o igual a cero.
 id_respuesto	int	False	

PRIMARY KEY NAME	COLUMNS	COMMENTS
 PK_Tbl_DetalleOrdenCompra	id_detalle_orden_compra	Representa el identificador único de cada línea de detalle de una orden de compra. Es un número entero positivo, obligatorio y único. Permite diferenciar cada producto o repuesto dentro de una orden de compra.

TYPE / NAME	COLUMNS	COMMENTS
 «index» IXFK_Tbl_DetalleOrdenCompra_Tbl_OrdenCompra	id_orden_compra	Mejora el rendimiento de las consultas que unen los detalles con sus órdenes de compra.
 «index» IXFK_Tbl_DetalleOrdenCompra_Tbl_Repuesto	id_respuesto	Facilita búsquedas y uniones entre los repuestos y el detalle de órdenes de compra.
 «check» CK_DetalleOrdenCompra_IdDetalle	id_detalle_orden_compra	Verifica que el ID sea mayor que 0.
 «check» CK_DetalleOrdenCompra_Cantidad	cantidad	Obliga a que la cantidad pedida sea positiva.
 «check» CK_DetalleOrdenCompra_PrecioUnitario	precio_unitario	Asegura que el precio unitario no sea negativo.
 «check» CK_DetalleOrdenCompra_Subtotal	subtotal	Comprueba que el subtotal sea coherente con cantidad × precio unitario y no sea negativo.

FOREIGN KEY NAME	COLUMNS	REFERENCES
 FK_Tbl_DetalleOrdenCompra_Tbl_OrdenCompra	id_orden_compra	Tbl_OrdenCompra(id_orden_compra)
 FK_Tbl_DetalleOrdenCompra_Tbl_Repuesto	id_respuesto	Tbl_Repuesto(id_vehiculorepuesto)







)
--	--	---



Tbl_CitaServicio


Database table in package 'Tables'




La tabla Tbl_CitaServicio almacena la información de las citas programadas por los clientes de NicaMotores para recibir servicios de mantenimiento o reparación en sus vehículos. Permite llevar un control de quién agenda la cita, para qué vehículo, qué servicio solicita, la fecha y hora asignada, y el estado de la cita (pendiente, confirmada, cancelada o atendida). Este registro es fundamental para la gestión de la agenda del taller, evitando duplicidades, mejorando la organización y garantizando un adecuado seguimiento de los servicios solicitados por los clientes.





Tbl_CitaServicio
Version 1.0 Phase 1.0 Proposed
LPT-DELL created on 22/09/2025. Last modified 22/09/2025
DBMS SQL Server 2012





COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
 id_cita	int	True	Aquí se almacena el identificador numérico único de cada cita registrada en la base de datos. No puede repetirse, debe ser un número entero positivo y no puede quedar vacío. Este campo sirve como clave primaria para identificar de manera inequívoca cada cita de servicio agendada por un cliente.
 id_cliente	int	True	Este campo almacena el identificador del cliente que solicita la cita de servicio. Es un número entero que establece una relación con la tabla Tbl_Cliente, garantizando que toda cita esté asociada a un cliente registrado en el sistema. No puede quedar vacío, ya que toda cita debe corresponder a un cliente específico.
 id_vehiculo	int	False	Aquí se almacena el identificador del vehículo que pertenece a NicaMotores y que será atendido en la cita. Es una clave foránea hacia la tabla Tbl_Vehiculo. Este campo es opcional porque la cita puede ser para un vehículo externo.
 id_vehiculo_externo	int	False	Este atributo registra el identificador del vehículo externo (no comprado en NicaMotores) que será atendido en la cita. Se relaciona con la tabla Tbl_VehiculoExterno. Es opcional, ya que la cita también puede ser para un vehículo interno.
 id_servicio	int	True	Este campo guarda el identificador del servicio solicitado en la cita. Se relaciona con la tabla Tbl_Servicio y permite registrar si se trata de un mantenimiento, reparación u otro tipo de servicio. No puede quedar vacío porque toda cita debe estar vinculada a un servicio específico.
 fecha_cita	date	True	Aquí se almacena la fecha agendada para la cita. Debe ser una fecha igual o posterior a la fecha

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
			actual (no puede estar en el pasado). No puede quedar vacía ya que toda cita debe tener definida una fecha de realización.
 hora_cita	time(0)	True	Este atributo guarda la hora específica de la cita, lo que permite organizar el calendario de servicios en el taller. No puede estar vacío porque es necesario definir un horario para atender cada cita. En futuras versiones se deberá validar que no se programen más de dos citas en el mismo horario y que no existan traslapes de tiempo de acuerdo con la duración del servicio. Esto podrá implementarse mediante triggers o reglas de negocio en la aplicación.
 estado	varchar(20)	True	Este campo almacena el estado actual de la cita, el cual puede tomar valores como Pendiente, Confirmada, Cancelada o Atendida. Permite dar seguimiento al ciclo de vida de la cita, desde que es agendada hasta que se atiende o se cancela. No puede estar vacío.

PRIMARY KEY NAME	COLUMNS	COMMENTS
 PK_Tbl_CitaServicio	id_cita	Asegura la unicidad de cada cita de servicio, identificando de manera exclusiva cada registro.

TYPE / NAME	COLUMNS	COMMENTS
 «index» IXFK_Tbl_CitaServicio_Tbl_Cliente	id_cliente	Índice asociado a la restricción FK_Tbl_CitaServicio_Tbl_Cliente. Optimiza las búsquedas de citas programadas por cliente, permitiendo acceder de manera rápida al historial de citas y agilizar reportes que relacionen clientes con los servicios solicitados.
 «index» IXFK_Tbl_CitaServicio_Tbl_Servicio	id_servicio	Índice creado para la restricción FK_Tbl_CitaServicio_Tbl_Servicio. Permite optimizar consultas relacionadas con la agenda de servicios, verificando la disponibilidad y frecuencia de citas por tipo de servicio, y contribuye a gestionar la carga de trabajo en el taller.
 «index» IXFK_Tbl_CitaServicio_Tbl_Vehiculo	id_vehiculo	Índice asociado a la restricción FK_Tbl_CitaServicio_Tbl_Vehiculo. Facilita las consultas de citas vinculadas a vehículos vendidos por la empresa, permitiendo generar reportes de servicios preventivos o mantenimientos programados por vehículo.

 «index» IXFK_Tbl_CitaServicio_Tbl_VehiculoExterno	id_vehiculo_externo	Índice creado para la restricción FK_Tbl_CitaServicio_Tbl_VehiculoExterno. Mejora el rendimiento en la búsqueda de citas de vehículos externos, garantizando trazabilidad y acceso rápido a la información de clientes que no adquirieron vehículos directamente en NicaMotores.
 «check» CK_Tbl_CitaServicio_Fecha	fecha_cita	Garantiza que la cita no se pueda agendar en una fecha pasada.
 «check» CK_Tbl_CitaServicio_Estado	estado	Restringe los valores posibles del estado de la cita a opciones válidas.
 «check» CK_Tbl_CitaServicio_Vehiculo	id_vehiculo, id_vehiculo_externo	Asegura que toda cita esté asociada al menos a un vehículo (interno o externo), no pudiendo quedar ambos campos vacíos.

FOREIGN KEY NAME	COLUMNS	REFERENCES
 FK_Tbl_CitaServicio_Tbl_Cliente	id_cliente	Tbl_Cliente(id_cliente)
 FK_Tbl_CitaServicio_Tbl_Servicio	id_servicio	Tbl_Servicio(id_servicio)
 FK_Tbl_CitaServicio_Tbl_VehiculoExterno	id_vehiculo_externo	Tbl_VehiculoExterno(id_vehiculo_externo)
 FK_Tbl_CitaServicio_Tbl_Vehiculo	id_vehiculo	Tbl_Vehiculo(id_vehiculo)