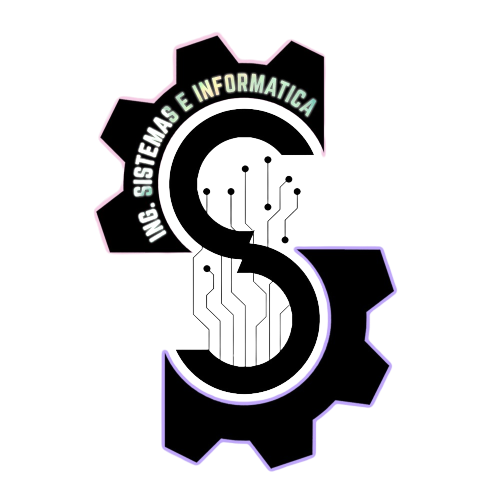
**“Año de la recuperación y consolidación de la economía peruana”**

**FACULTAD DE INGENIERÍA**

**ESCUELA ACADÉMICA PROFESIONAL DE:**

**INGENIERÍA DE SISTEMAS E INFORMATICA**

**Asignatura:**

Diseño de software

**NRC: 17205**

**Grupo:** Error 404

**Docente:** Ing.HUGO ESPETIA HUAMANGA

**INTEGRANTES**

1. Zavala Huacarpuma Juan Aldair (100%)
2. Lizarraga Aliaga Alexis Joaquin (100%)
3. Casilla Mamani Ronaldo (100%)
4. Challco Ccohuanqui Samir Junior (100%)

**Enlace de GITHUB:** [AQUÍ](https://github.com/NothingToSayH/DS_Base-de-datos---Proyecto-Final)

**Cusco - Perú - 2025**

ÍNDICE

[**Descripción del Proyecto y el Diseño de la Base de Datos**](#_x8l1i1orv9cy) **3**

[**PROYECTO**](#_hauax2s66r3r) **3**

[**Diseño de la Base de Datos**](#_4ryabj2ed0ie) **4**

[**Diagrama Entidad-Relación (DER)**](#_456opxqwa67o) **5**

[**Diagrama Lógico**](#_kshclbfvzfhu) **6**

[**Diagrama Físico (sql)**](#_1iub345rdz5k) **7**

[**Justificación de Nuestro Diseño**](#_d1hkyonj30y3) **8**

[**Explicación del Diseño de Base de Datos para Corporativo Valu Calderón**](#_vdpt6jgp2o4p) **8**

[Decisiones Generales de Diseño](#_n9yfq7vzgisu) 8

[Elección de entidades (Tablas)](#_nylknt3dd4td) 8

[Atributos y tipos de Datos](#_bh6n472med4t) 8

[Claves Primarias](#_vbt7zpmfcln0) 8

[Relaciones y Claves Foráneas](#_wrkfg4xzsq8z) 8

[Normalización y Organización](#_b7p2k69kxlt3) 8

[Flexibilidad y Escalabilidad](#_f930s7ervh6l) 9

[Elección de Motor de Base de Datos](#_fnilsdkggmdg) 9

[Esquema de Nomenclatura](#_b0aeiewizm56) 9

[Ventajas de Nuestro Diseño Actual](#_tip6jg18vaj) 9

[Flexibilidad](#_4z5iuxfsubof) 9

[Escalabilidad](#_797ezt9uqhdr) 9

[Integridad de Datos](#_5lw4gkjucja4) 9

[Trazabilidad](#_r5std77fq4n7) 9

[**Implementación de la Base de Datos y CRUD**](#_sy2rfeezpcxz) **10**

[**Procedimientos Almacenados CRUD**](#_ympt8hk941iv) **12**

[Procedimiento de Inserción (INSERT)](#_iup6dw83l92q) 13

[Procedimiento de Seleccionar (SELECT o GET)](#_1h77joml02m8) 20

[Procedimiento de Actualizar (UPDATE)](#_xfh8p44wwol4) 32

[Procedimiento de Eliminar (DELETE)](#_tqntwsjlalbe) 39

[**Reflexión**](#_jvrq7wir2smo) **47**

[**Conclusiones**](#_a4fbjtnr5dgg) **48**

[**Referencias**](#_eqm25pl8vcnv) **49**

# Descripción del Proyecto y el Diseño de la Base de Datos

## **Proyecto**

Desarrollo de un Sistema de Información para ADMINISTRAR LAS VENTAS del CORPORATIVO VALU CALDERÓN E.I.R.L.

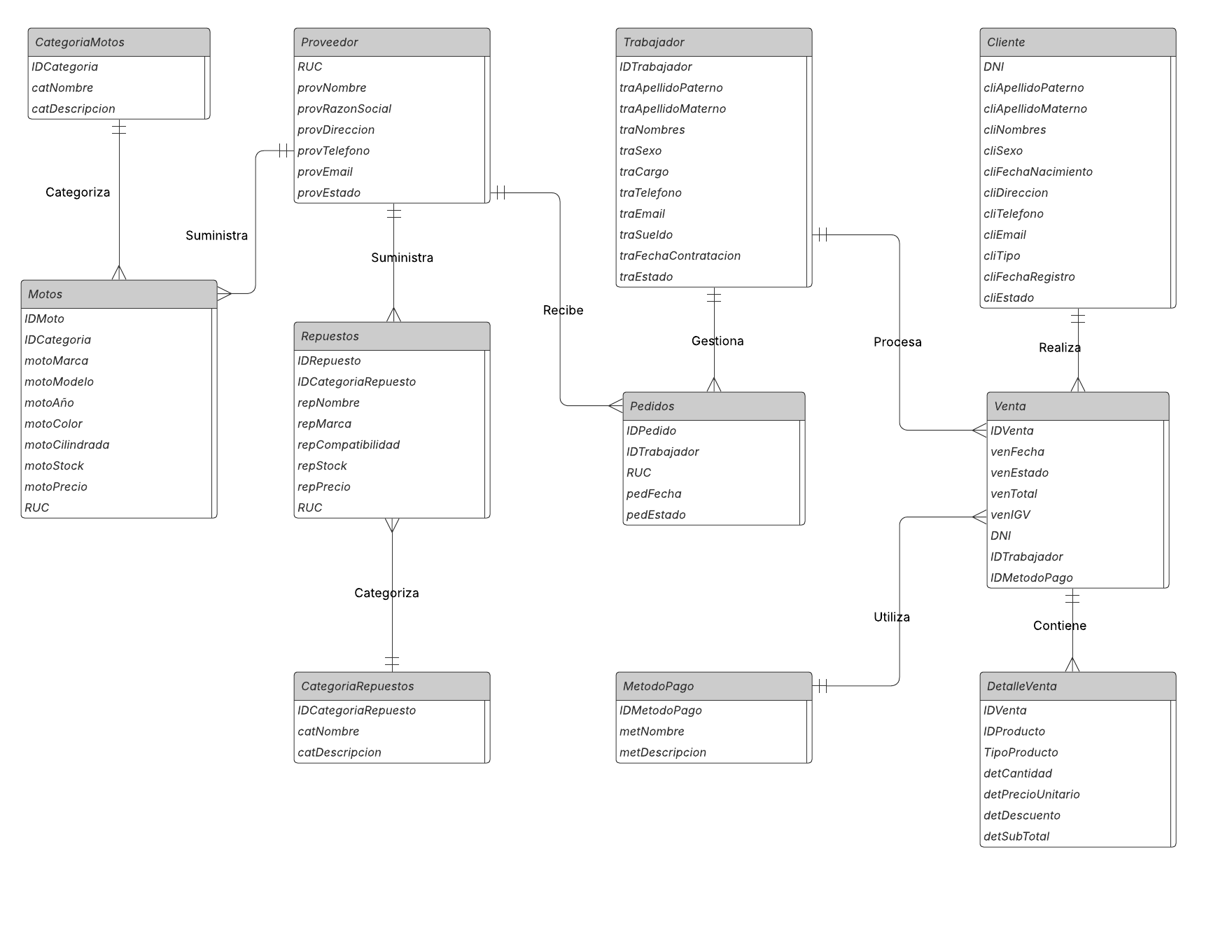
El proyecto se centra en diseñar una arquitectura de software orientada a las necesidades operativas de la empresa Corporativo Valu Calderón E.I.R.L., especializada en la venta y mantenimiento de motocicletas. La solución propuesta se basa en una arquitectura cliente-servidor, aprovechando el entorno Visual Studio 2022 y SQL Server Management Studio, tecnologías que ofrecen integración, escalabilidad y seguridad.

## **Diseño de la Base de Datos**

La **Base de Datos** es importante para el funcionamiento del sistema de información que administra las ventas del **CORPORATIVO VALU CALDERÓN E.I.R.L**., ya que permite **almacenar, organizar y gestionar de manera eficiente** toda la información relacionada con los **productos, clientes, pedidos, pagos y reportes de ventas**. Gracias a ella, el sistema puede ofrecer acceso rápido a los datos actualizados, asegurar la integridad de la información, facilitar la toma de decisiones y mejorar la eficiencia operativa. Además, garantiza la trazabilidad de cada transacción y contribuye a evitar errores humanos, duplicidades o pérdidas de información clave.

# 

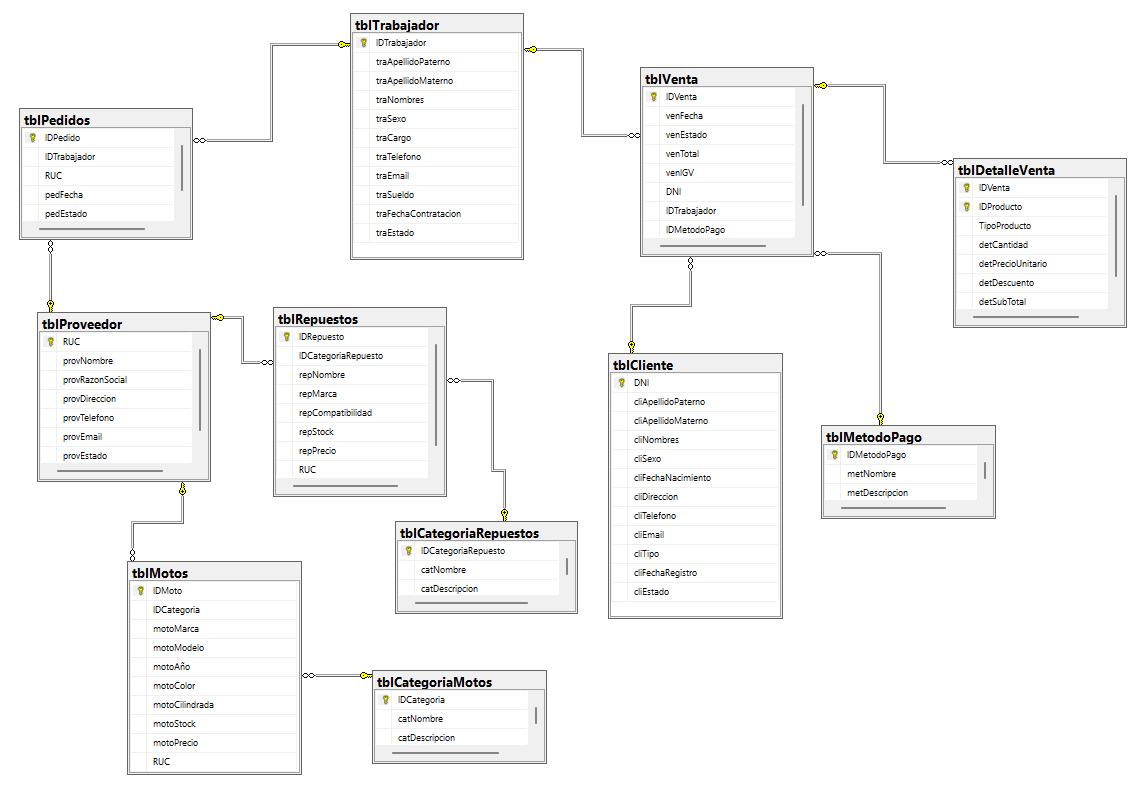
# **Diagrama Entidad-Relación (DER)**



# **Diagrama Lógico**

# 

# **Diagrama Físico (sql)**



# **Justificación de Nuestro Diseño**

# Explicación del **Diseño** de **Base de Datos** para **Corporativo Valu Calderón**

## Decisiones Generales de Diseño

### Elección de entidades (Tablas)

Se han creado entidades que representan los **elementos** clave del **negocio**: clientes, trabajadores, proveedores, motos, repuestos, pedidos, ventas, comprobantes, pagos y métodos de pago. Estas reflejan procesos reales como la venta de **productos**, la **gestión** de **inventario** y la atención al **cliente**.

### Atributos y tipos de Datos

* Se usaron **VARCHAR** para identificadores, nombres, direcciones y correos por su flexibilidad al manejar texto.
* **DECIMAL** se eligió para precios, sueldos y totales, ya que permite controlar la precisión en cálculos monetarios.
* **DATE** y **DATETIME** se usan para registrar eventos cronológicos importantes como ventas, contrataciones o pagos.
* Campos como Sexo, **TipoProducto**, y Estado usan valores controlados (**CHAR**, **VARCHAR** con **DEFAULT** o **CHECK**) para mantener la integridad de los datos.

### Claves Primarias

Cada tabla tiene una clave primaria única **(ID, DNI, RUC, etc.)** que permite **identificar** cada registro sin ambigüedades. Se escogieron códigos alfanuméricos para permitir flexibilidad en la generación de identificadores únicos.

### Relaciones y Claves Foráneas

Se usaron claves foráneas para asegurar la integridad referencial:

* Las ventas están **asociadas** a un **cliente** y a un **trabajador**.
* Los pedidos se relacionan con **trabajadores** y **proveedores**.
* Motos y repuestos se vinculan a sus respectivas **categorías** y **proveedores**.
* Los detalles de **venta** se **vinculan** con las **ventas** y permiten registrar tanto **motos** como **repuestos** mediante un campo que **distingue** el tipo de **producto**.

### Normalización y Organización

El diseño sigue principios de normalización para evitar la redundancia.

* Las categorías están en tablas separadas.
* Los métodos de pago, comprobantes y tipos de comprobante se mantienen normalizados para facilitar su reutilización y control.

### Flexibilidad y Escalabilidad

La estructura permite fácilmente agregar más productos, **tipos** de **comprobantes**, **métodos** de **pago** o nuevas categorías sin alterar el diseño base. También se prevé el seguimiento del estado de **ventas**, **pedidos** y **pagos** para facilitar la gestión operativa.

### Elección de Motor de Base de Datos

Nuestro diseño está plasmado en el software SQL Server, lo que permite:

* Alta integridad referencial mediante claves foráneas
* Transacciones ACID
* Escalabilidad para manejar grandes volúmenes de datos
* Robustez para un sistema comercial crítico

### Esquema de Nomenclatura

* Todas las tablas tienen el prefijo "tbl" (ej: tblCliente, tblVenta)
* Los atributos llevan prefijos específicos relacionados con su tabla (ej: moto-, cli-, ven-)
* Estas características mejora la comprensión y mantenimiento del código

### Ventajas de Nuestro Diseño Actual

#### Flexibilidad

* El diseño permite manejar distintos tipos de productos (motos y repuestos)
* Soporta múltiples métodos de pago y tipos de comprobantes

#### **Escalabilidad**

* La estructura normalizada facilita la expansión a nuevas categorías o líneas de negocio
* Separación clara entre entidades maestras y transaccionales

#### **Integridad de Datos**

* Restricciones de integridad referencial garantizan consistencia
* Tipos de datos adecuados para cada atributo

#### **Trazabilidad**

* Seguimiento de ventas, pagos, trabajadores involucrados
* Historial de clientes y su actividad comercial

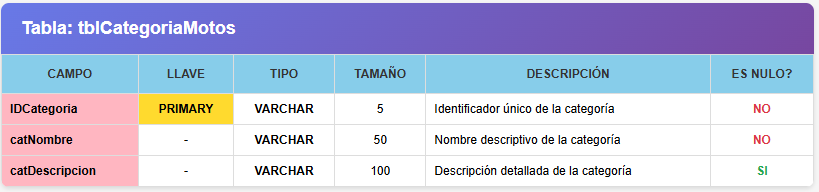
Este diseño es adecuado para un sistema de gestión de ventas y stock de una empresa comercializadora de motos y repuestos, proporcionando el balance necesario entre normalización, rendimiento y mantenibilidad.

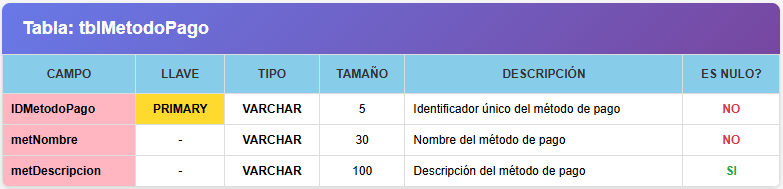
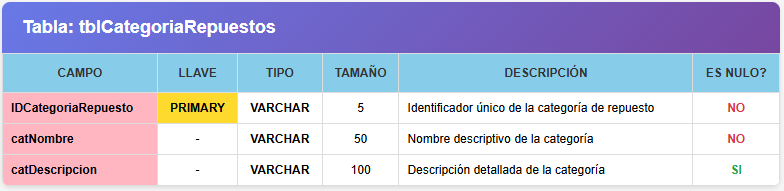
# **Implementación de la Base de Datos y CRUD**

**Creación de la Base de Datos:** Implementación de la base de datos en el motor de SqL Server Management Studio.

Creación de Tablas:

**Definición de todas las tablas** con sus respectivos **tipos de datos**, **restricciones** de integridad (NOT NULL, UNIQUE, CHECK), **claves primarias y foráneas**.

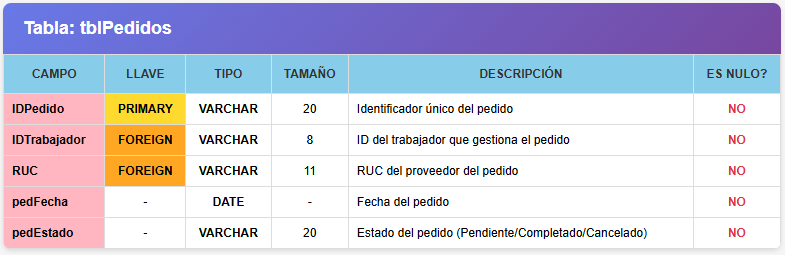


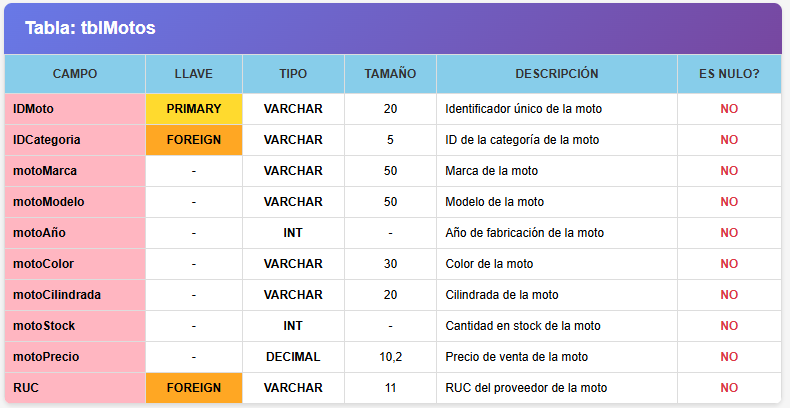


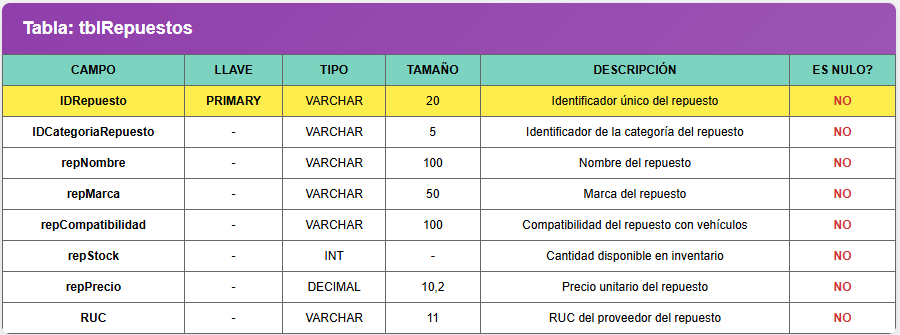












# Procedimientos Almacenados CRUD

Por cada tabla de la base de datos, **los equipos deben implementar** cuatro procedimientos almacenados:

# 

## **Procedimiento de Inserción (INSERT)**

Crear un nuevo registro en una tabla. Es el equivalente a "Añadir" o "Guardar Nuevo".

Recibe los datos para el nuevo registro como parámetros (por ejemplo, el DNI, nombre, dirección de un nuevo cliente). Internamente, utiliza el comando INSERT INTO para añadir la fila a la tabla.

Tabla Categoría Motos

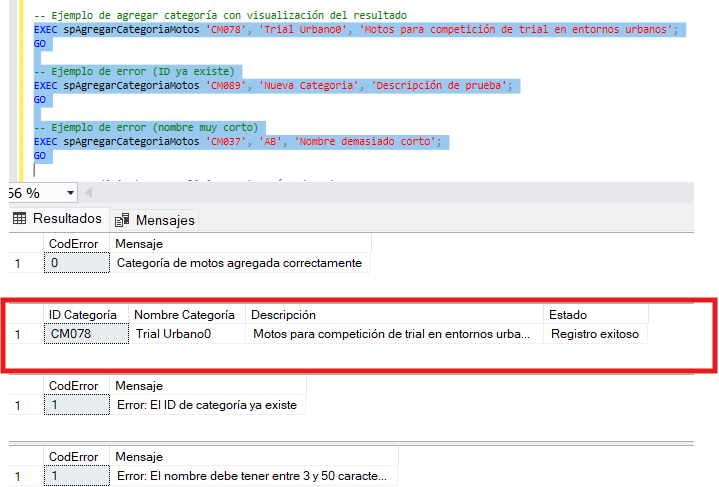
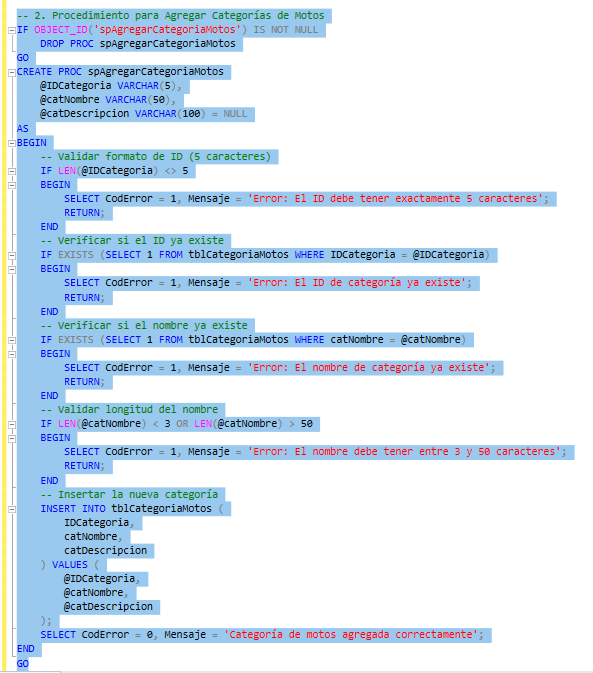


Tabla de Categoría de Repuestos

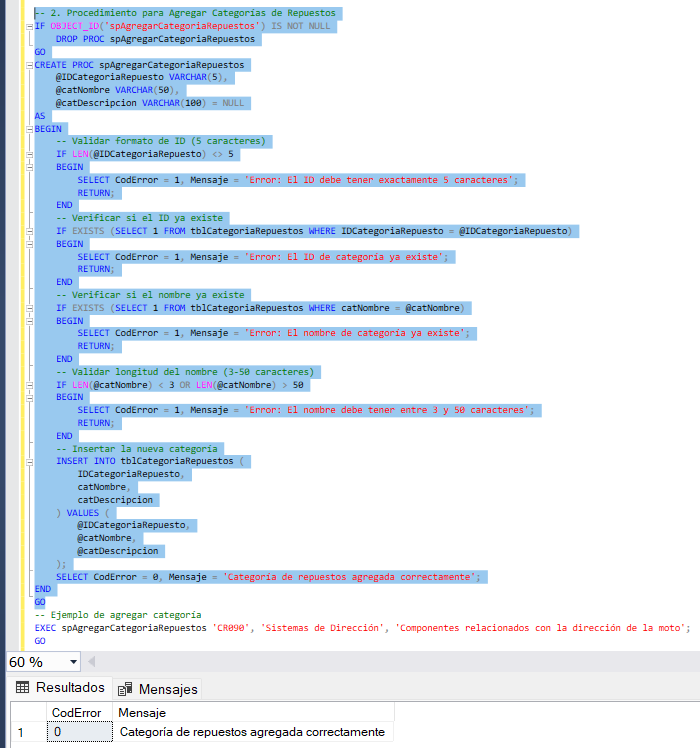


Tabla de método de pago

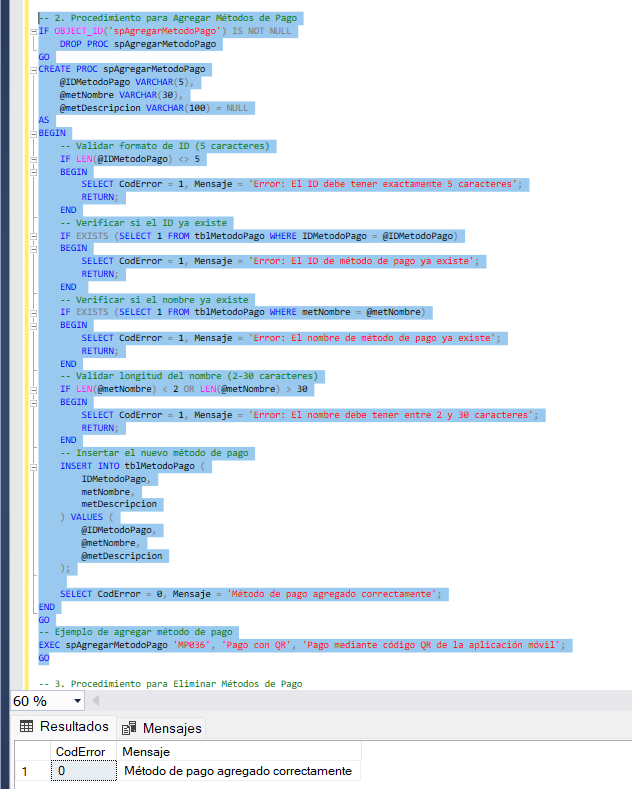
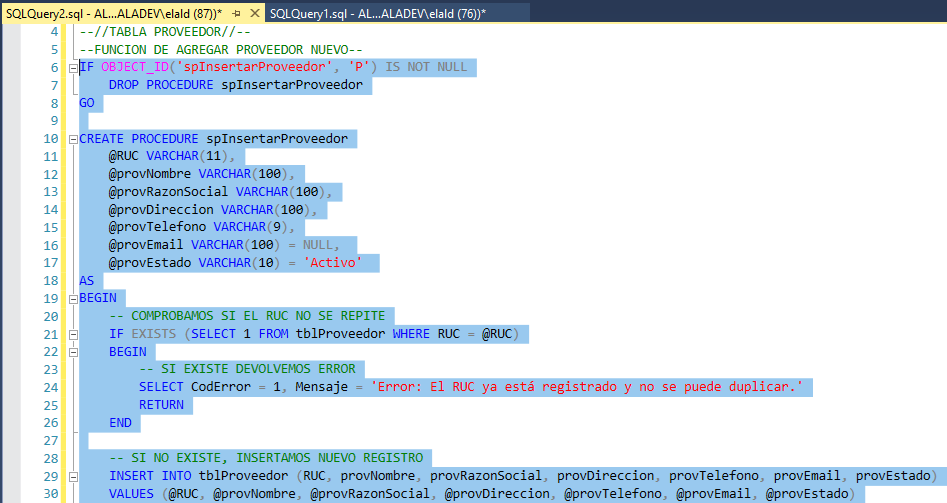


Tabla Proveedor

**Imagen 1**



**Imagen 2**

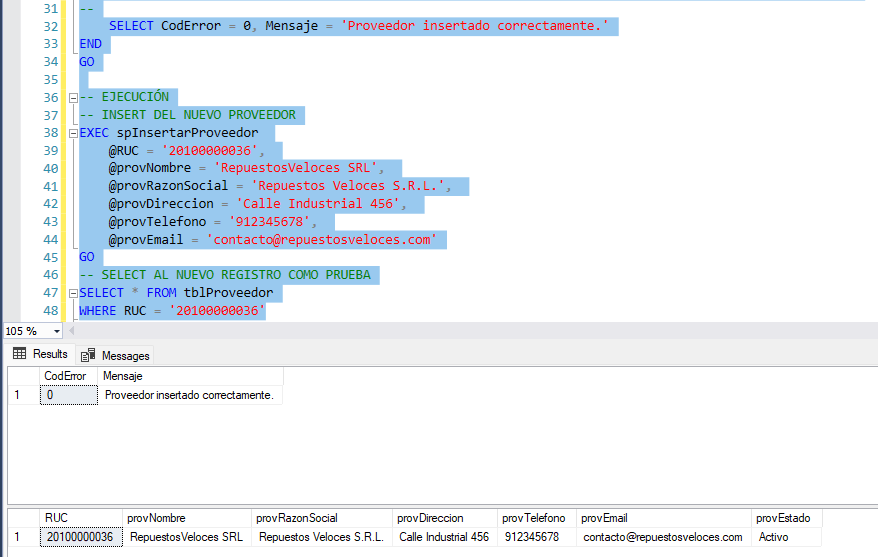
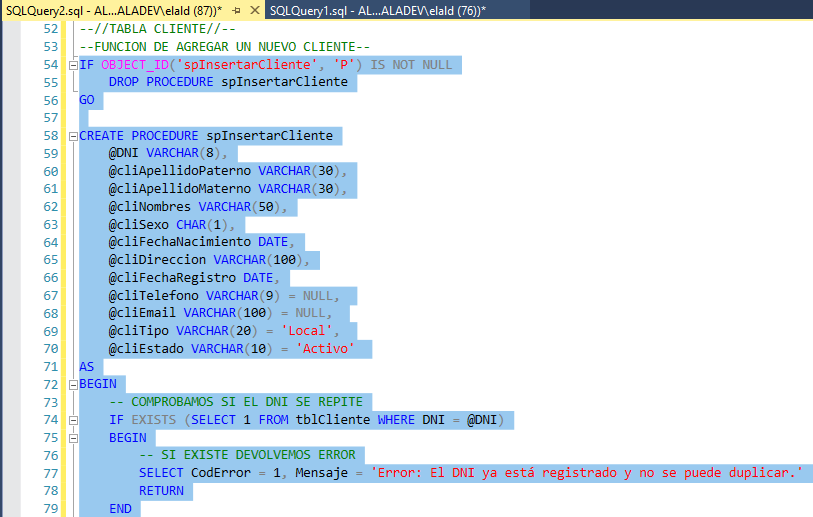


Tabla Cliente

**Imagen 1**



**Imagen 2**

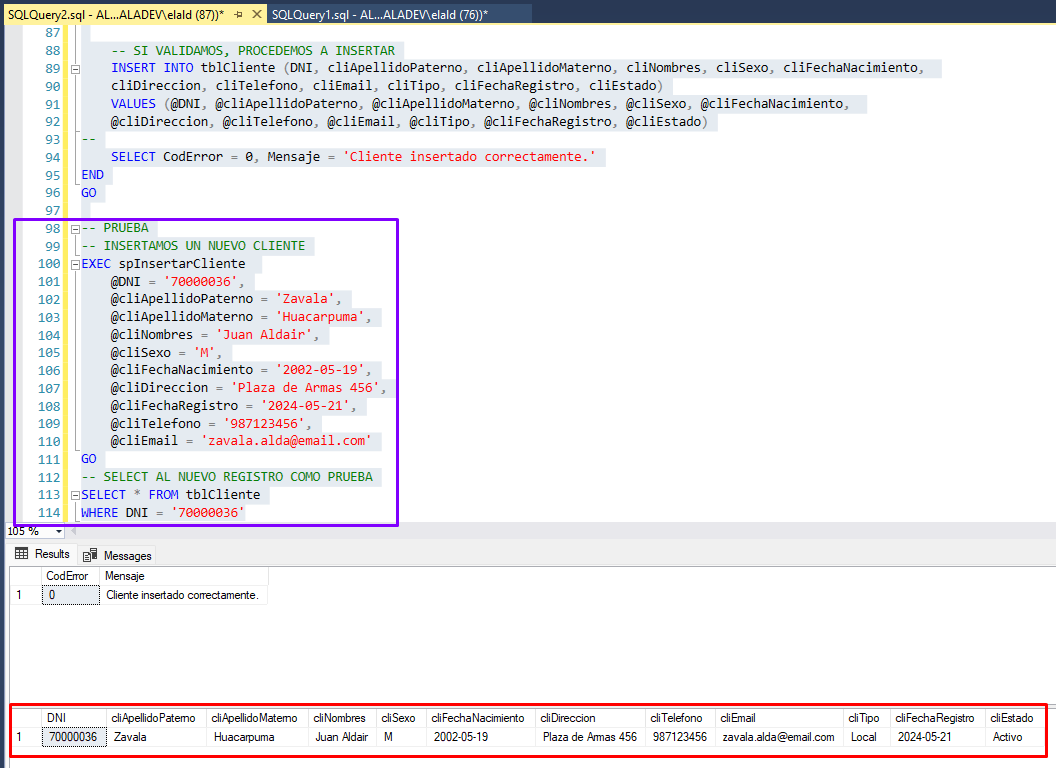
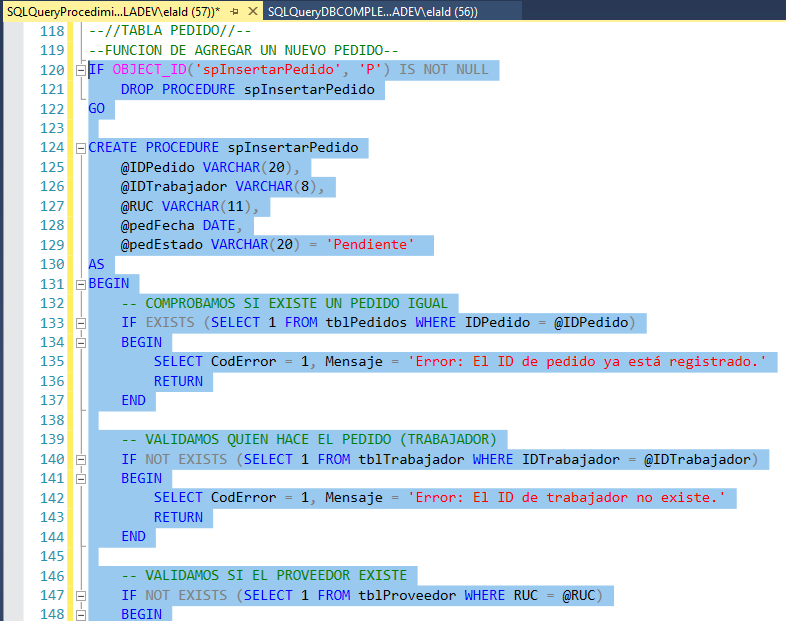


Tabla de Pedido

**Imagen 1**



**Imagen 2**

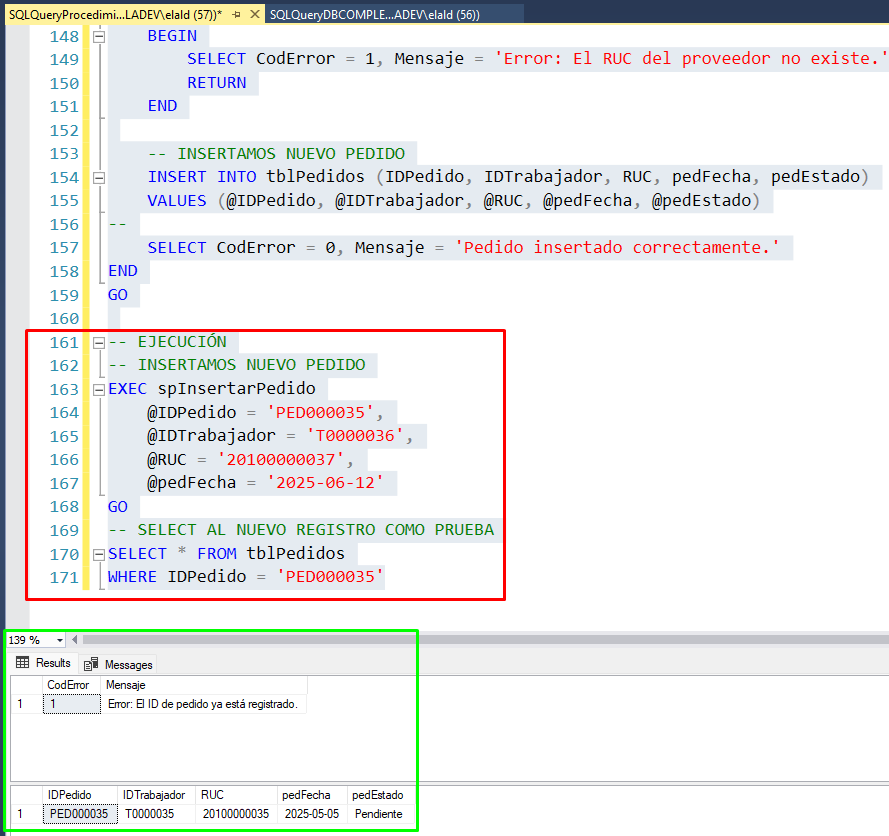
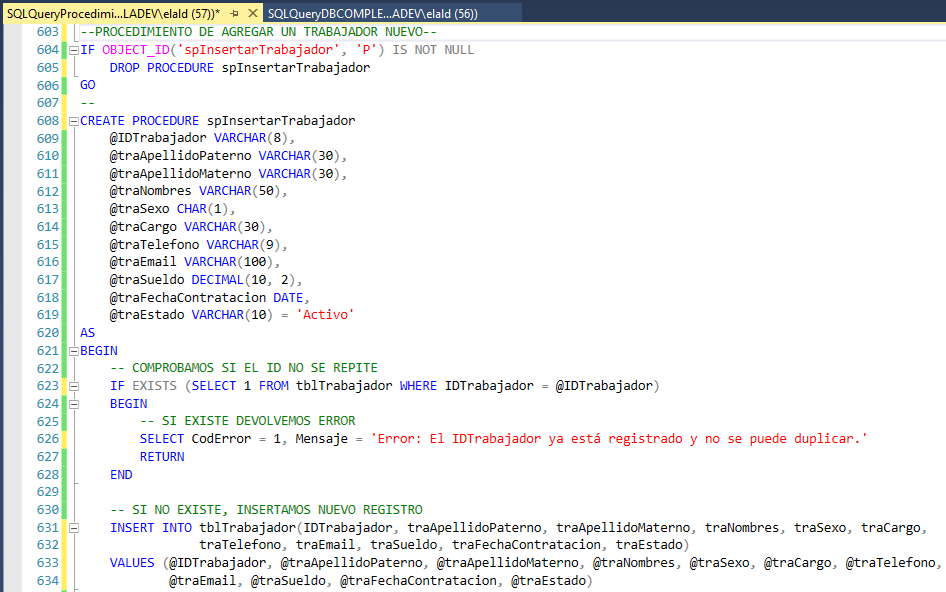


Tabla Trabajador

**Imagen 1**



**Imagen 2 Prueba**

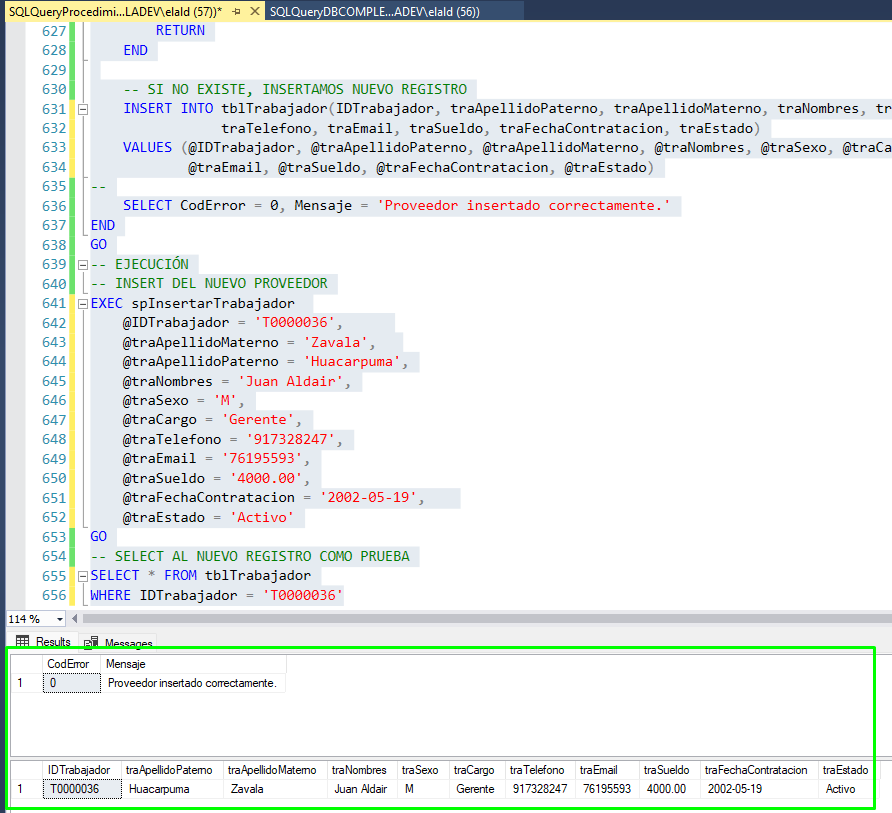


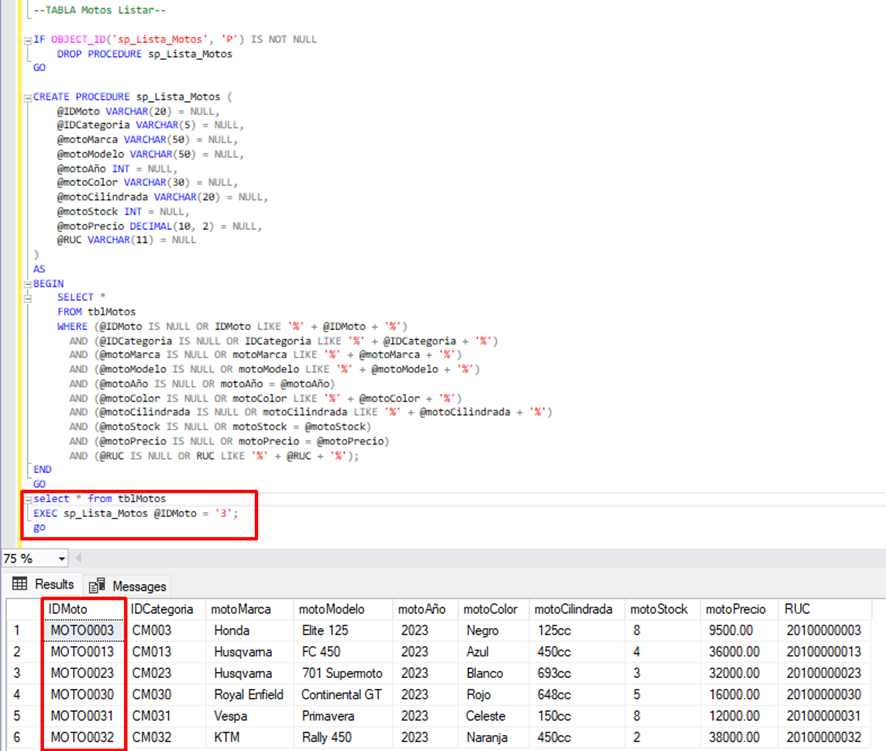
Tabla Motos

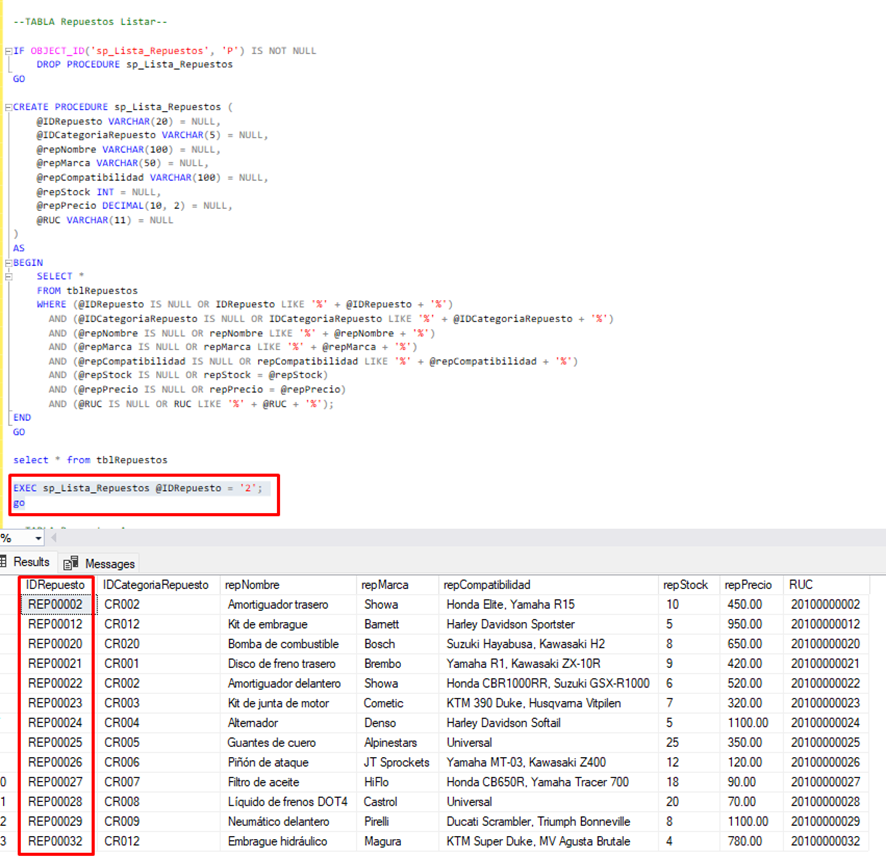
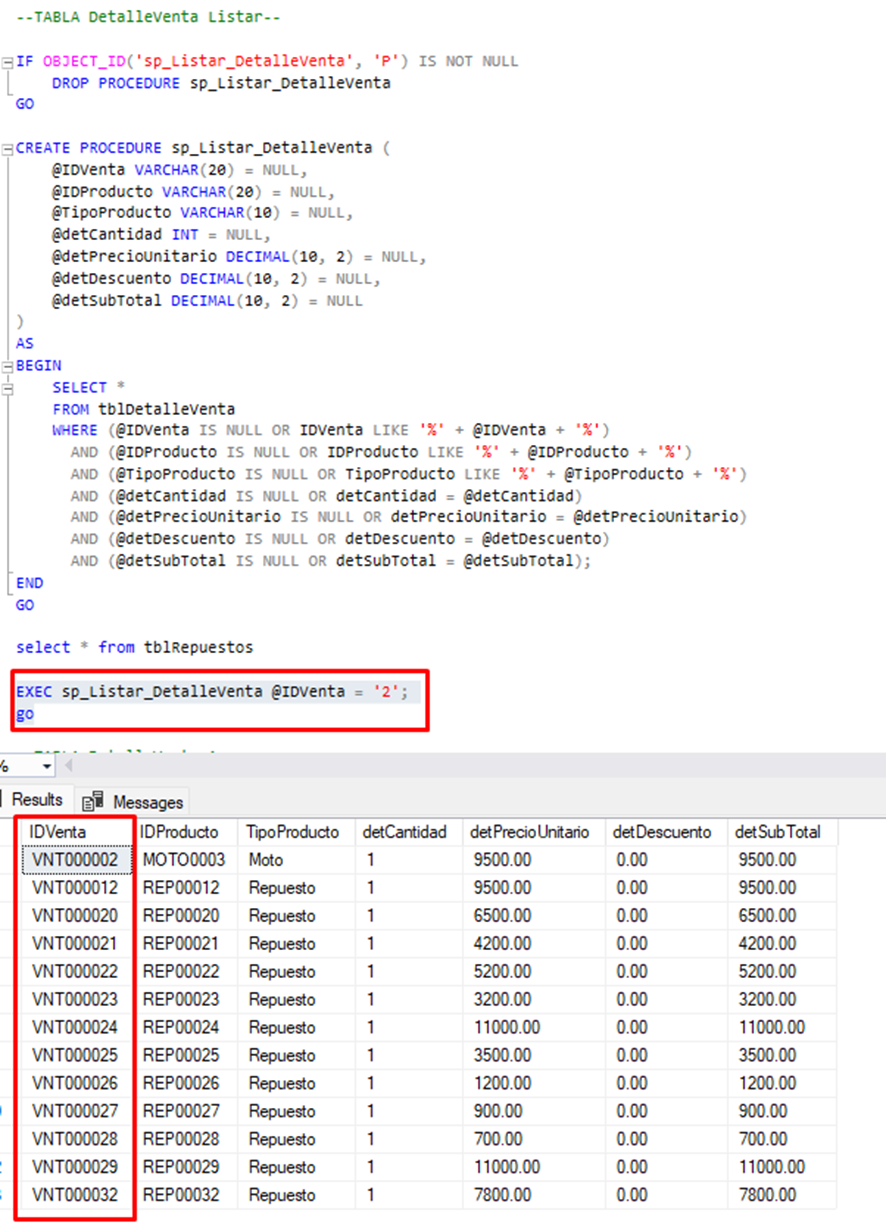
Tabla Repuestos

Tabla Detalle Venta

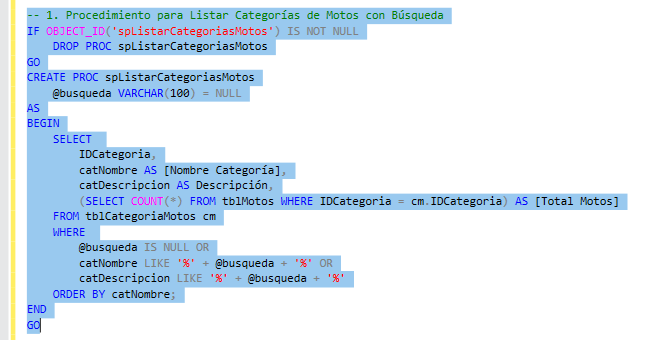


## 

## **Procedimiento de Seleccionar (SELECT o GET)**

Consultar, buscar y leer datos que ya existen en las tablas. Es la operación de "Búsqueda" o "Listado". Utiliza el comando SELECT para recuperar filas y columnas. Puede ser simple (traer todos los registros) o complejo, permitiendo filtrar por criterios específicos (como buscar clientes por apellido).

Tabla de Categoría de Motos



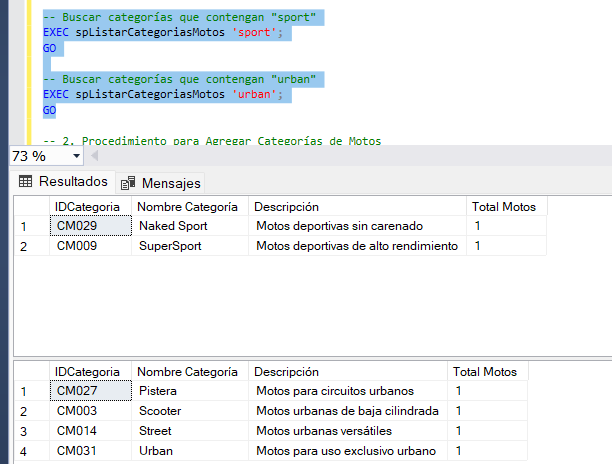
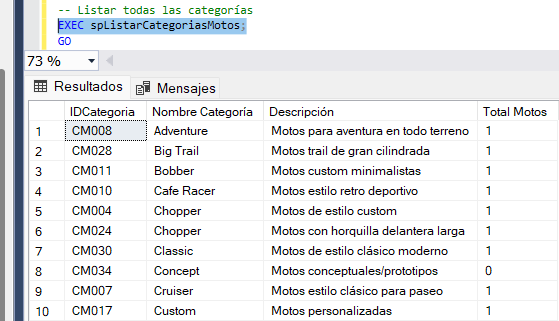


Tabla Categoría Repuestos

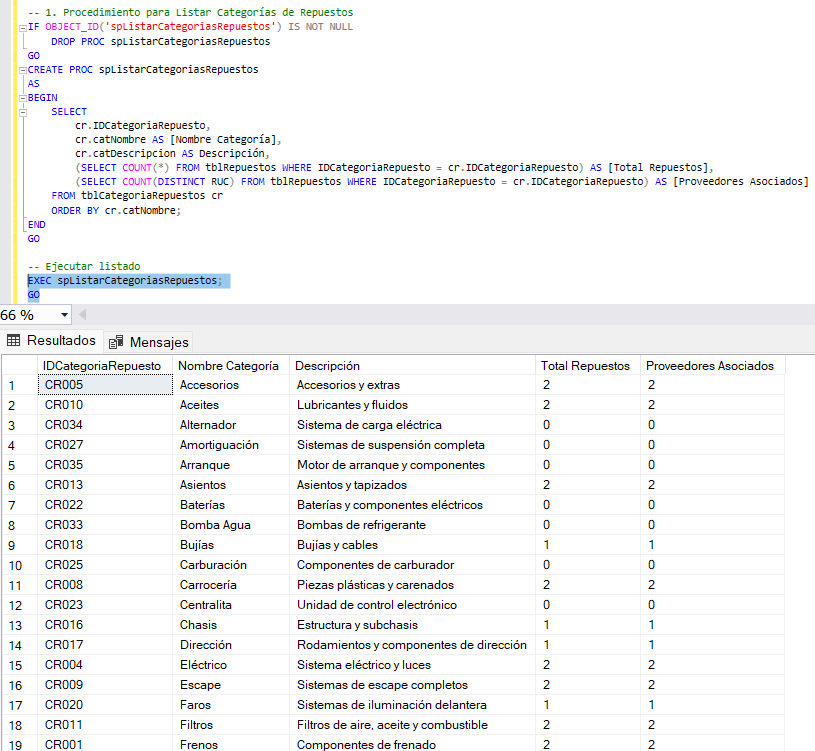
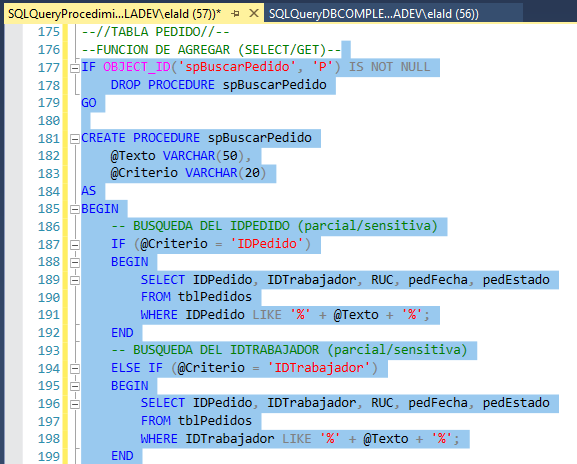
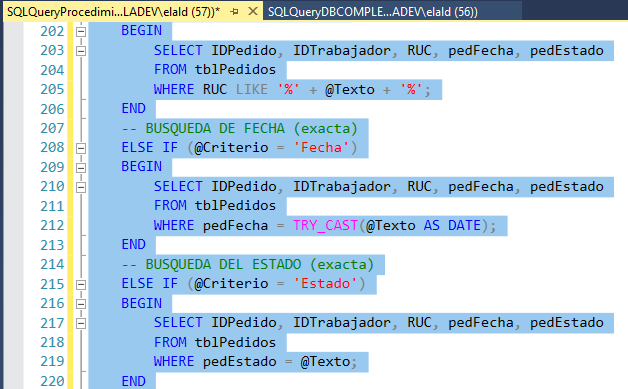


Tabla Pedido

**Imagen 1**





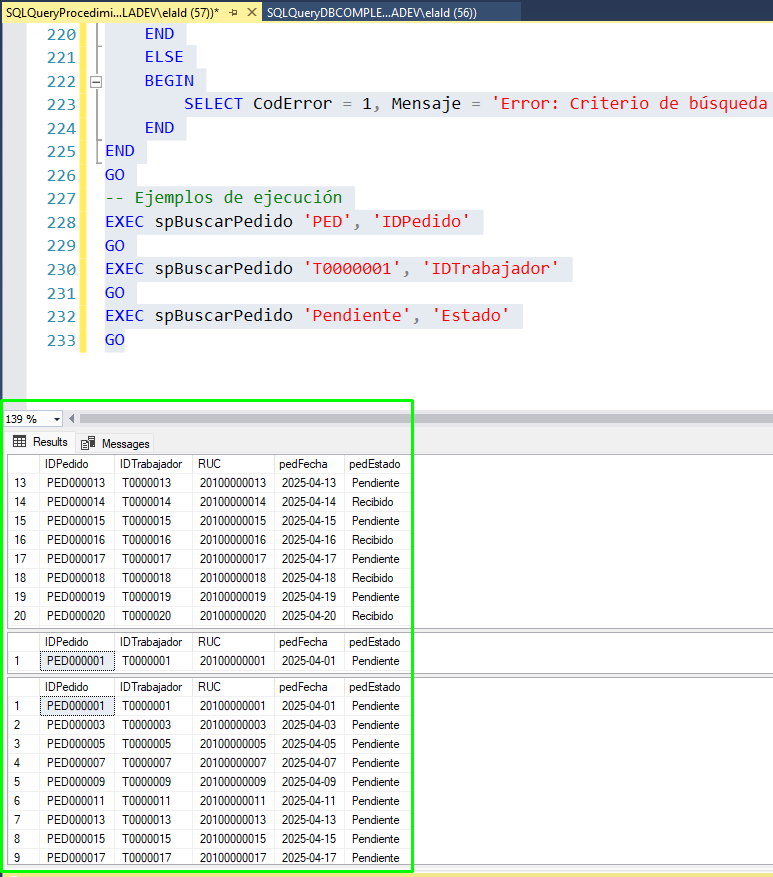


Tabla Metodo Pago

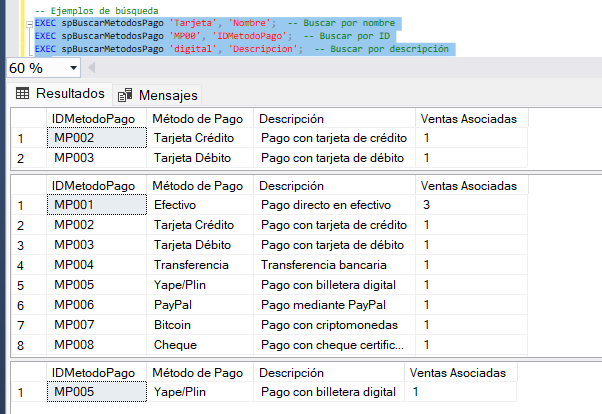
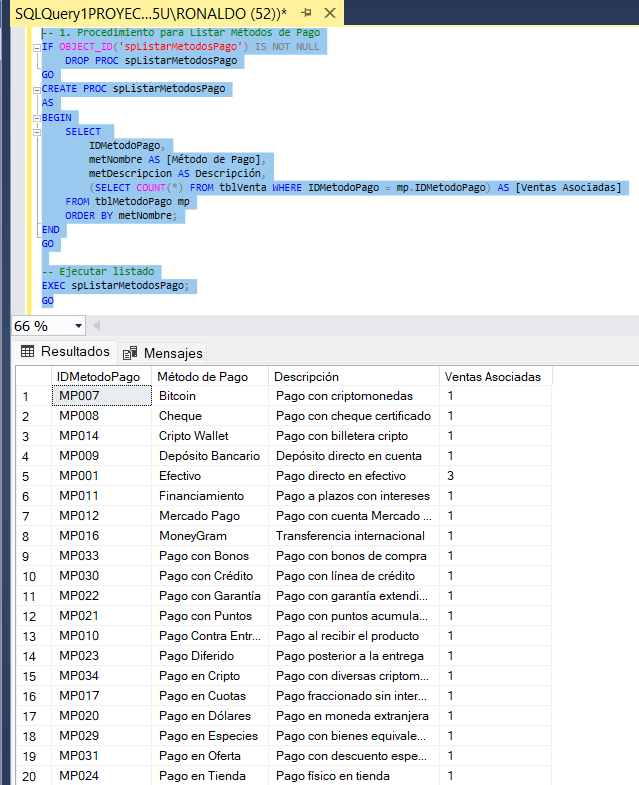
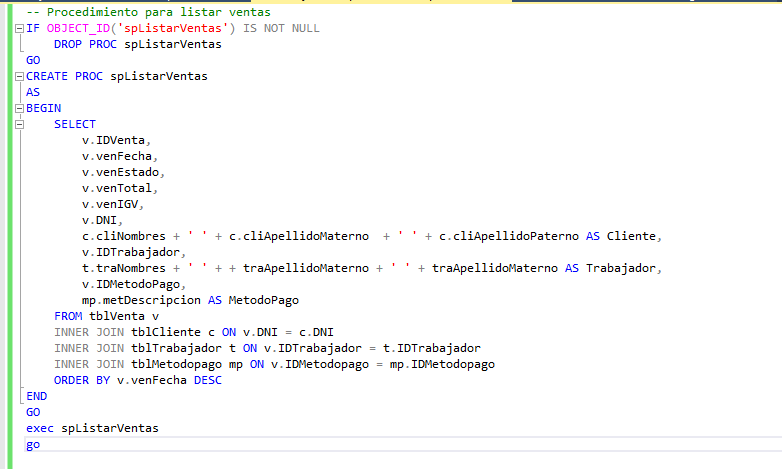


Tabla Venta



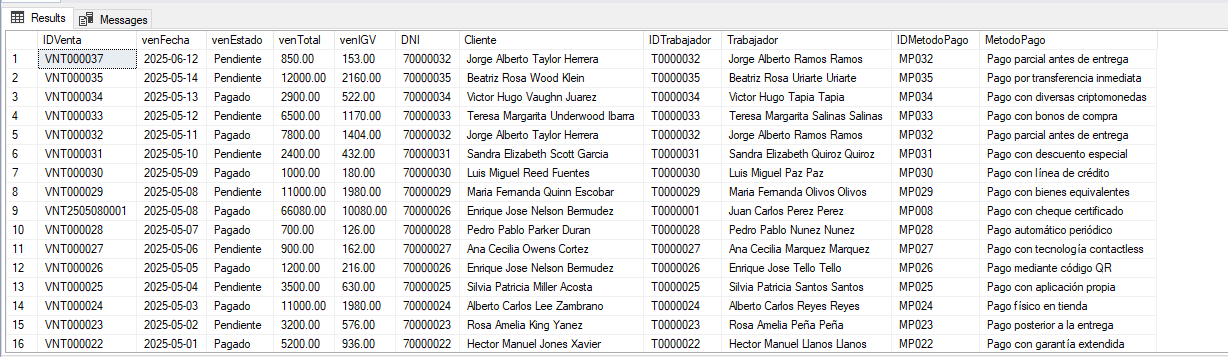
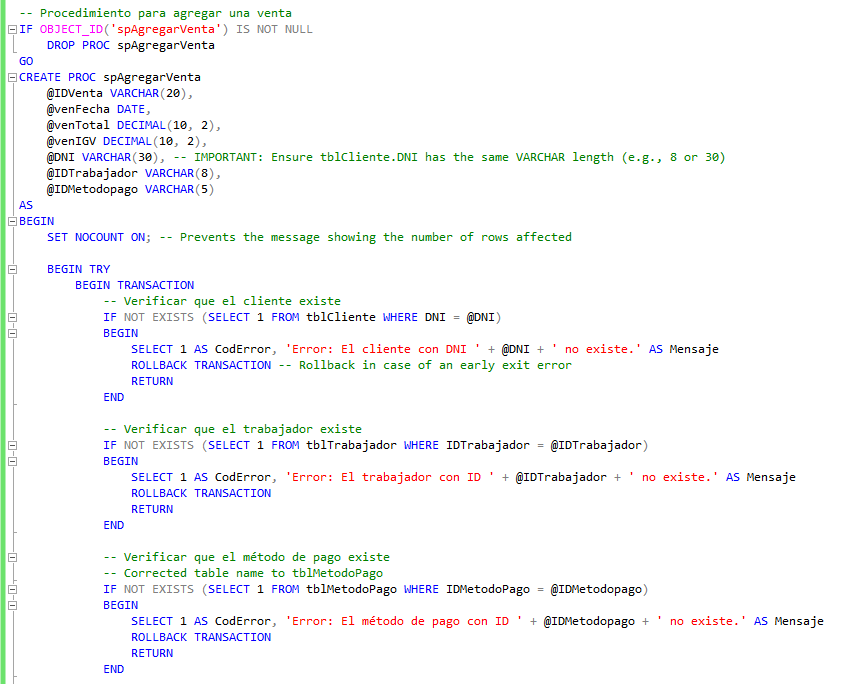


Tabla Venta



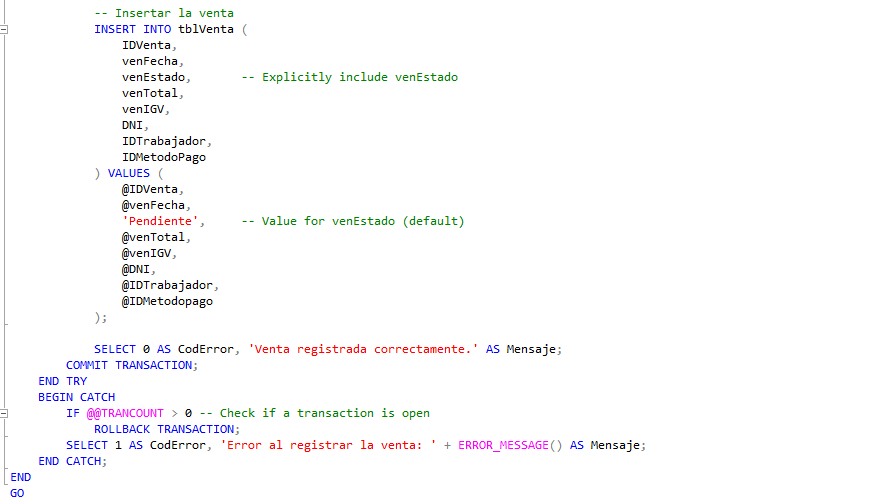
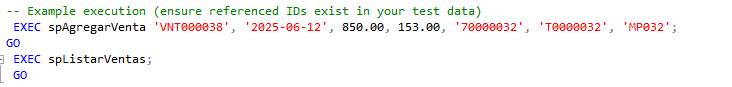
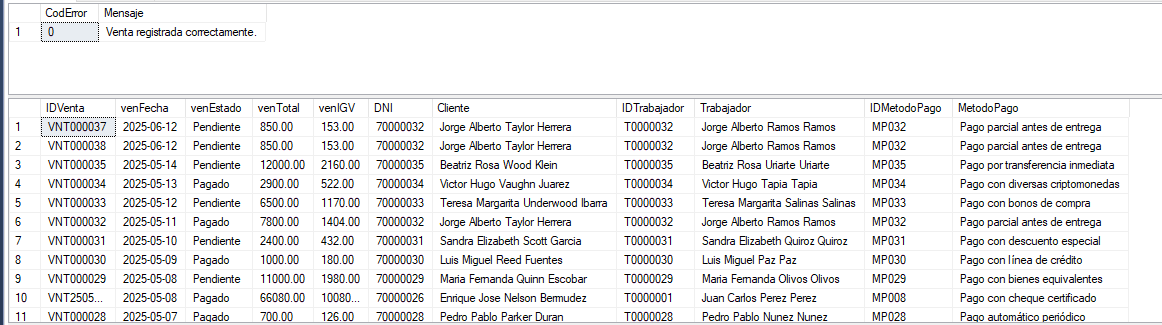
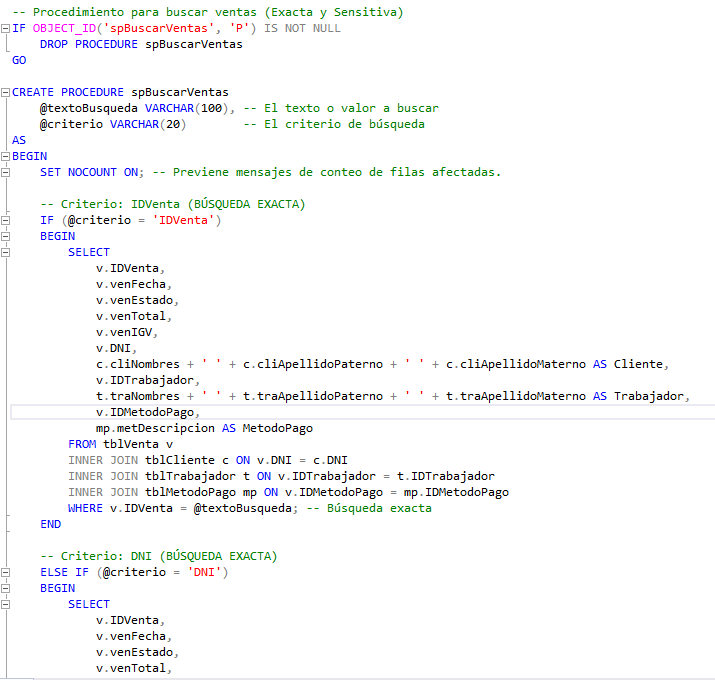
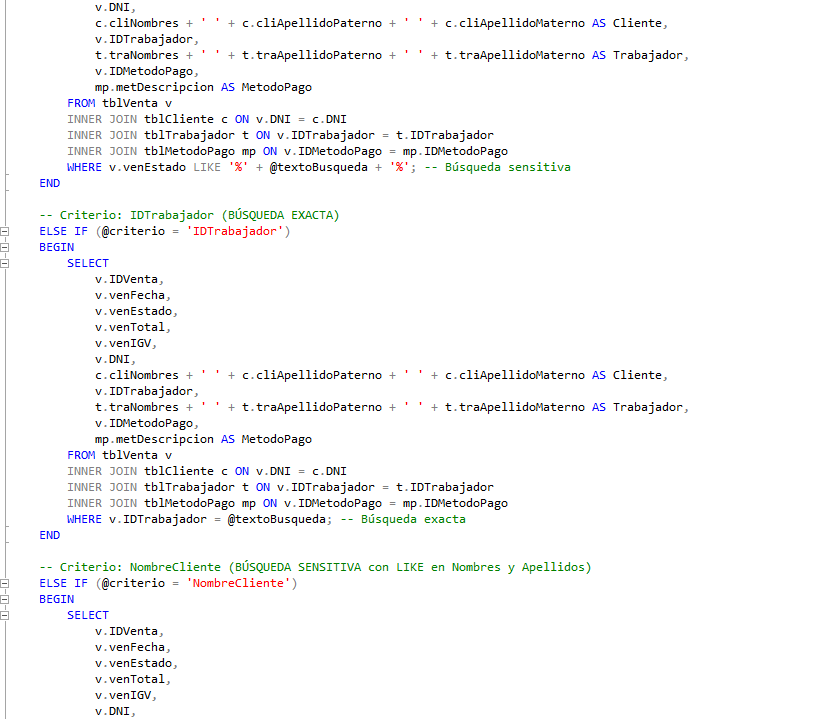
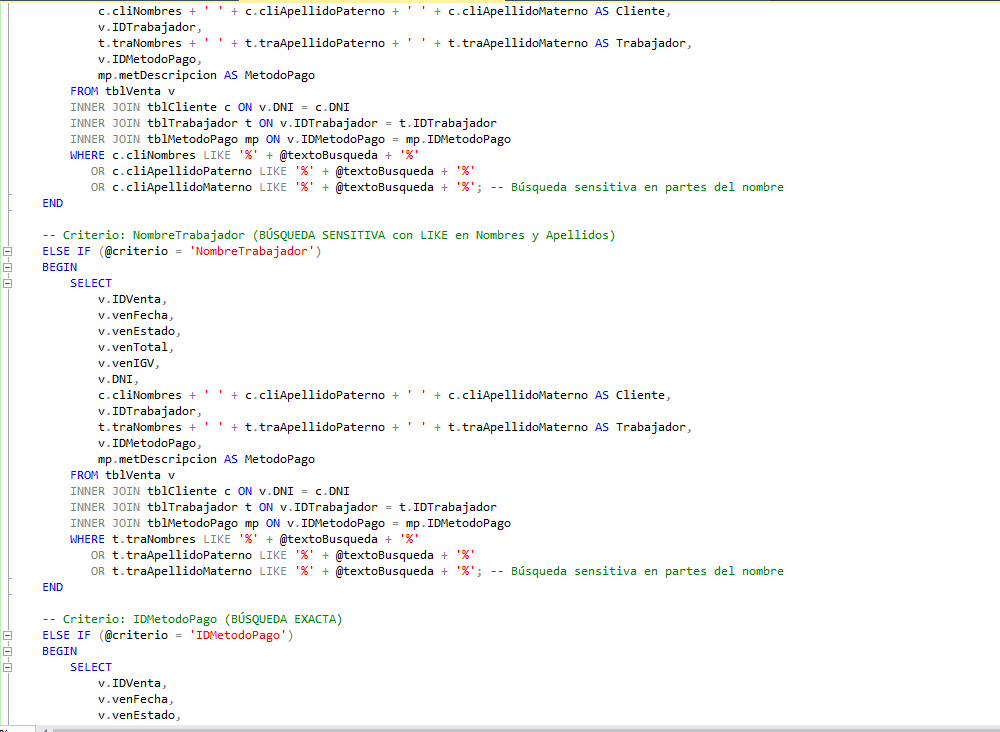
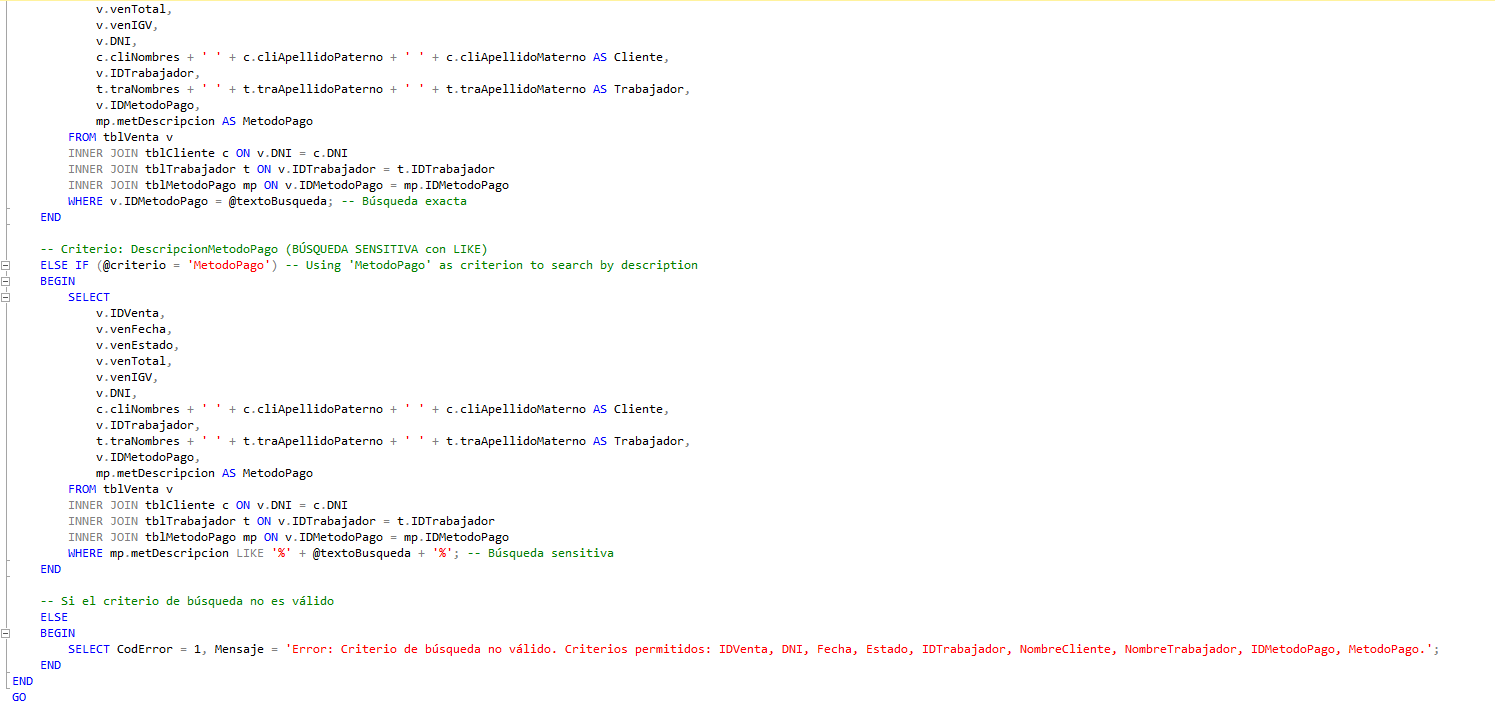
  
  


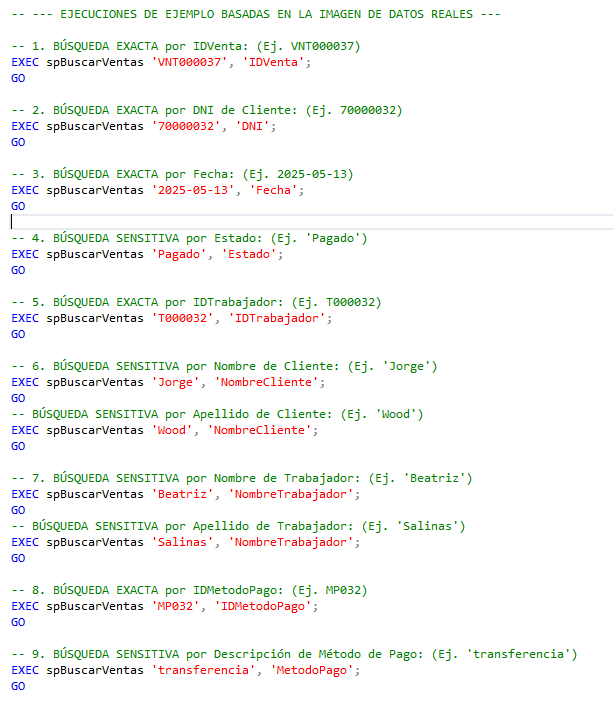
Tabla Venta con Búsqueda Sensitiva y Exacta









Ejemplo   




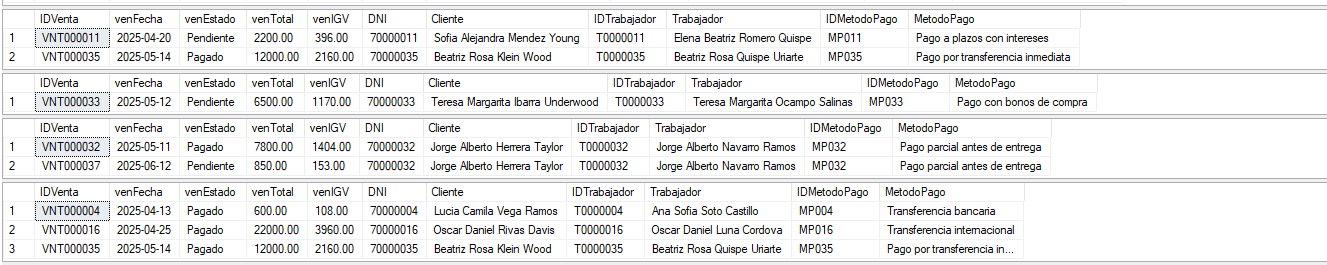
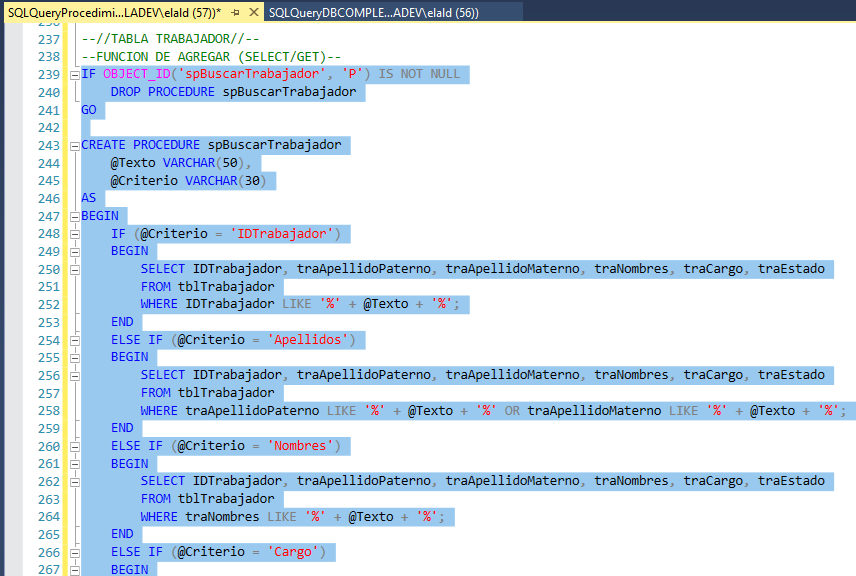


Tabla Trabajador

**Imagen 1**



**Imagen 2**

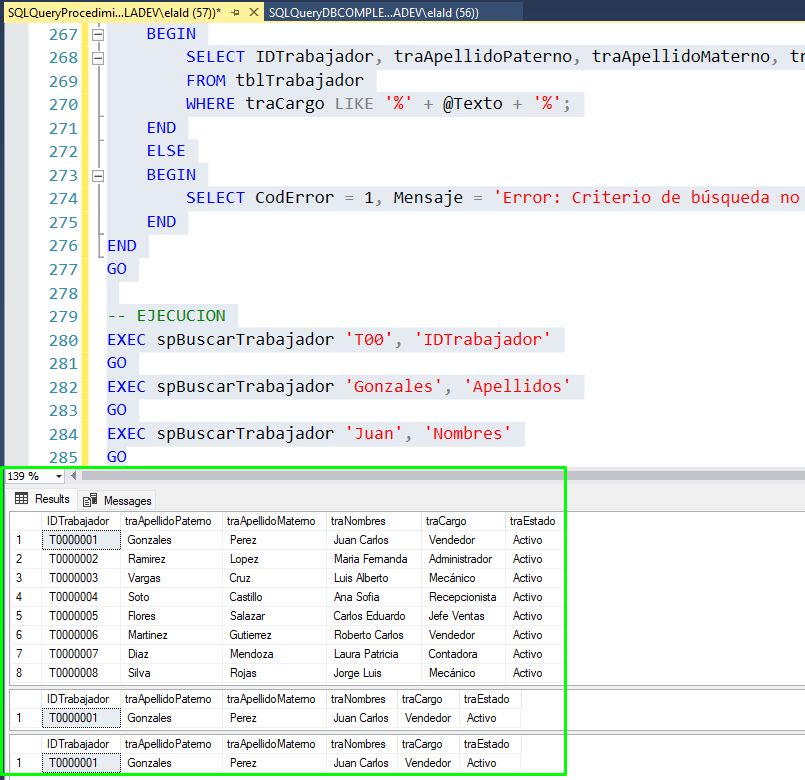
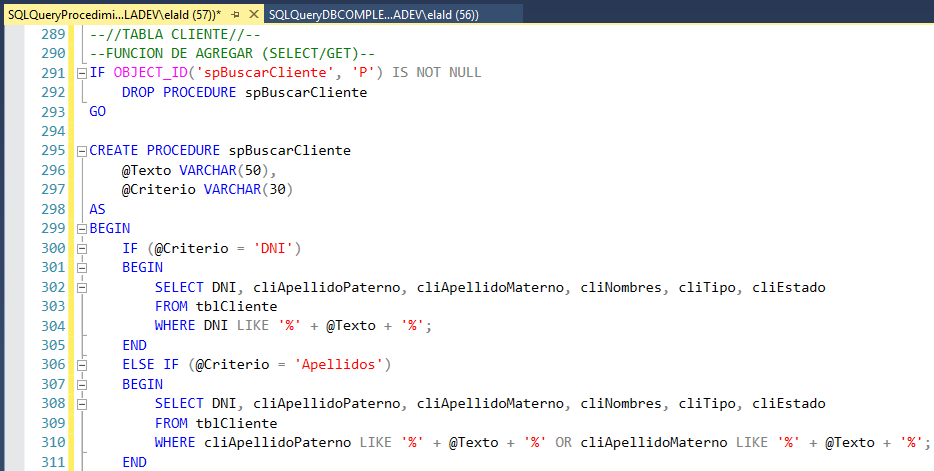
****

Tabla Cliente

**Imagen 1**



**Imagen 2 Prueba**

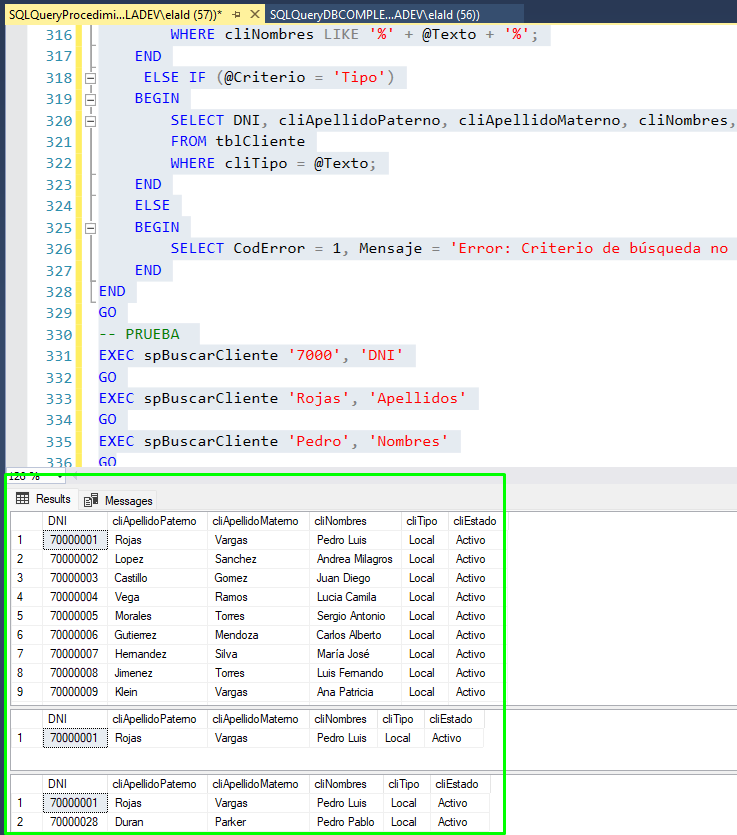
****

Tabla Motos

Tabla Repuestos

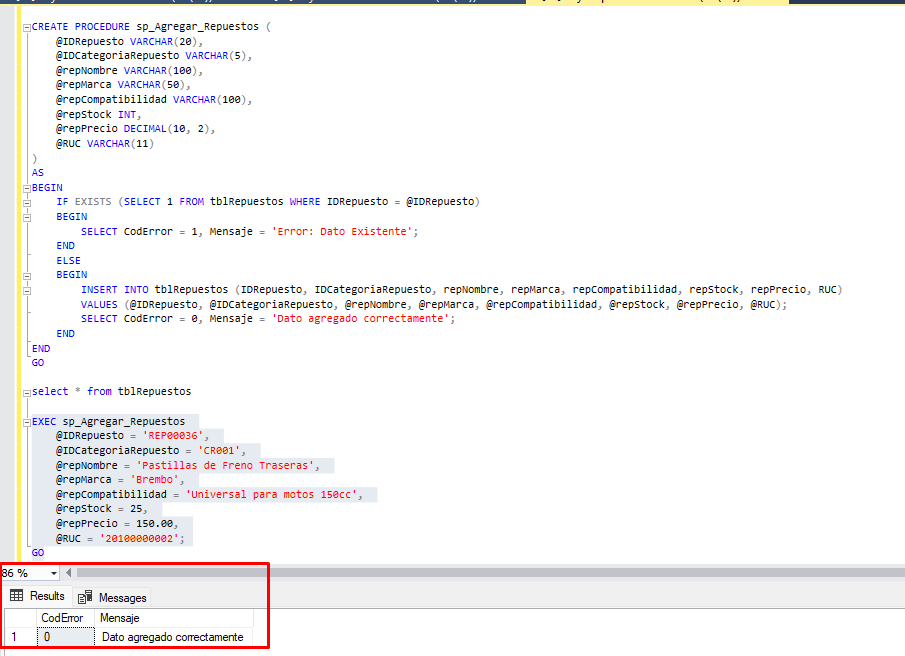
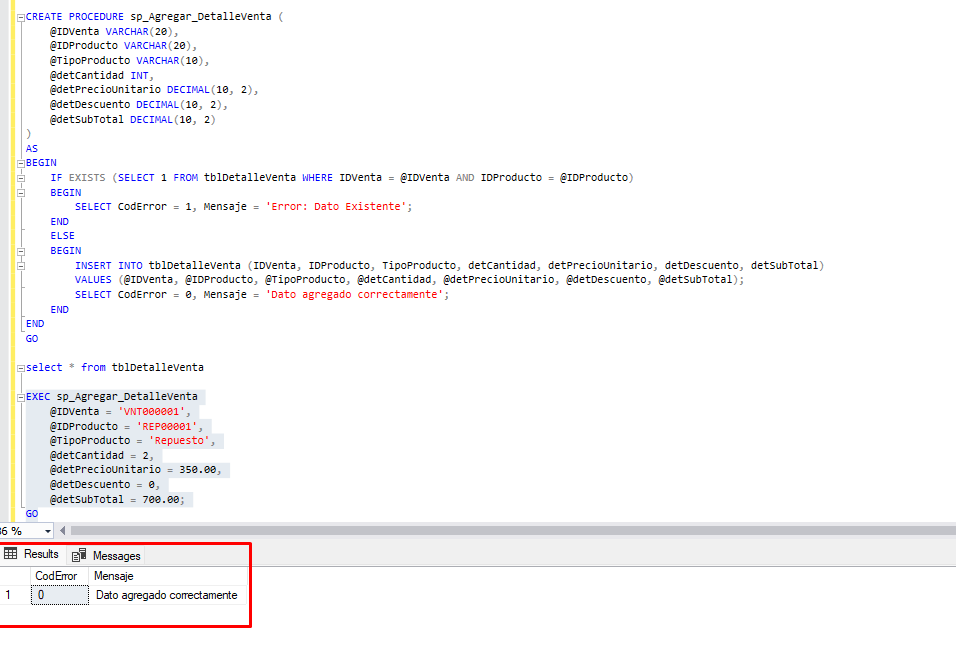


Tabla Detalle Venta

## 

## 

## 

## 

## **Procedimiento de Actualizar (UPDATE)**

Modificar un registro que ya existe. Es la función de "Editar" o "Modificar". Generalmente, recibe un identificador único (como el ID o DNI) para saber qué registro cambiar, y los nuevos valores para las columnas que se van a actualizar. Usa el comando UPDATE.

Tabla Categoría Repuesto

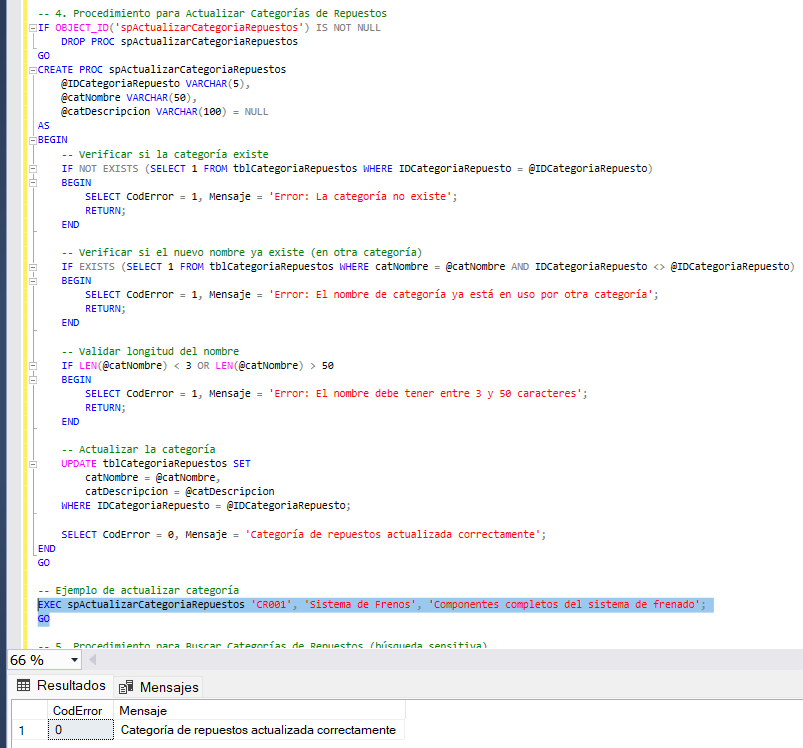
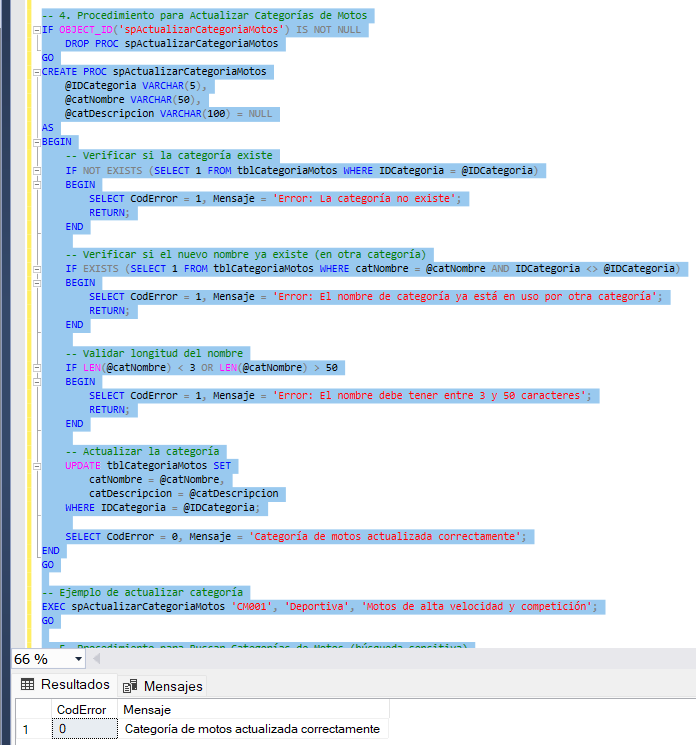


Tabla Categoría Motos



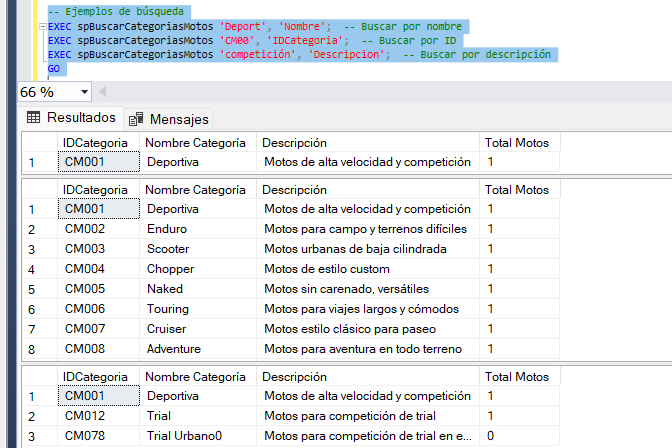
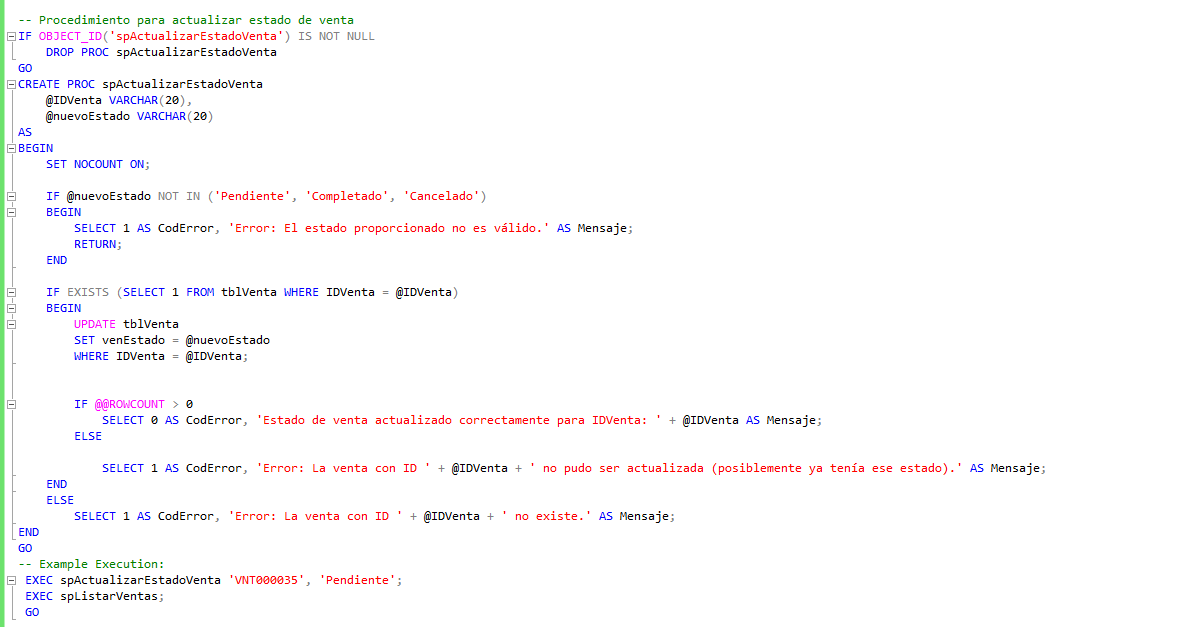
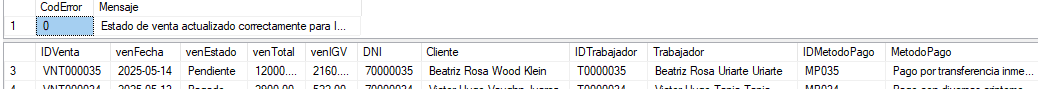
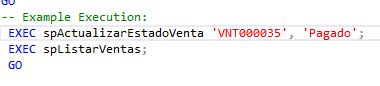


Tabla Venta





**Ejecución**



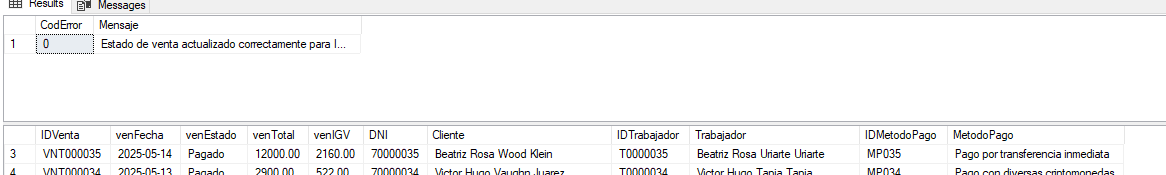
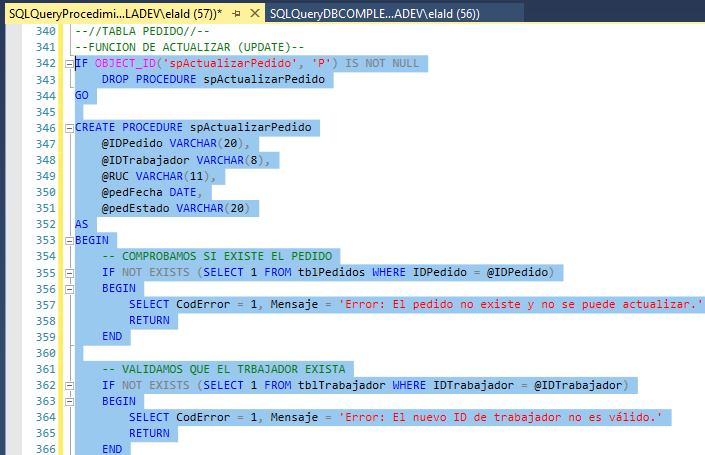


Tabla Pedido

**Imagen 1**



**Imagen 2 Prueba**

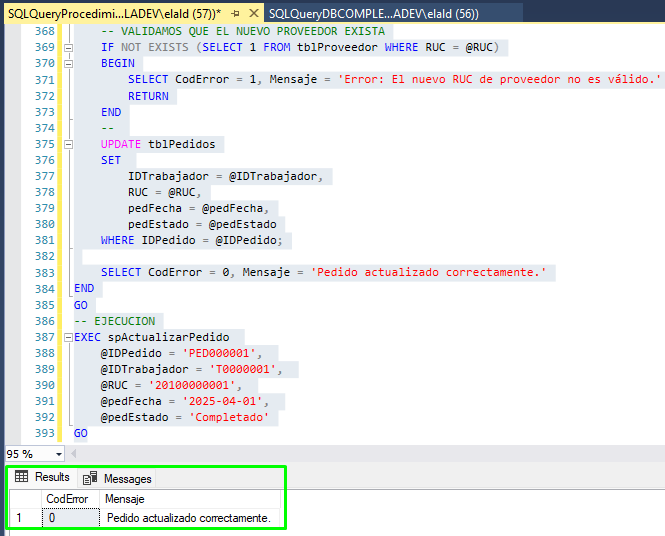
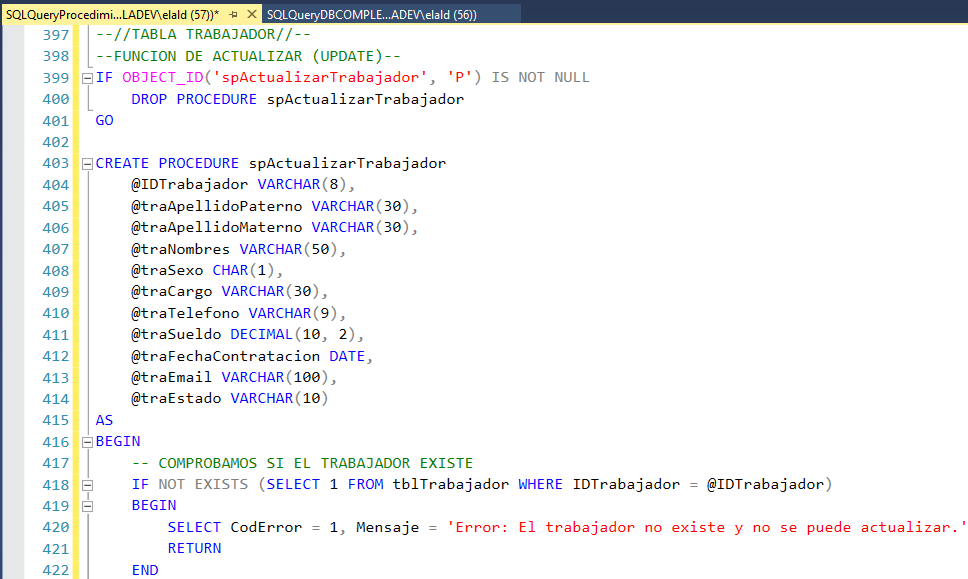


Tabla Trabajador

**Imagen 1**



**Imagen 2 Prueba**

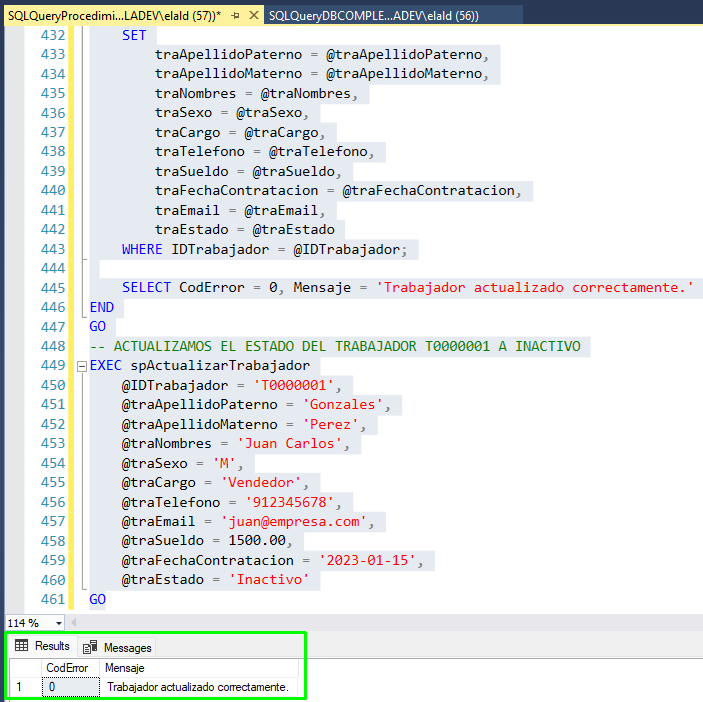
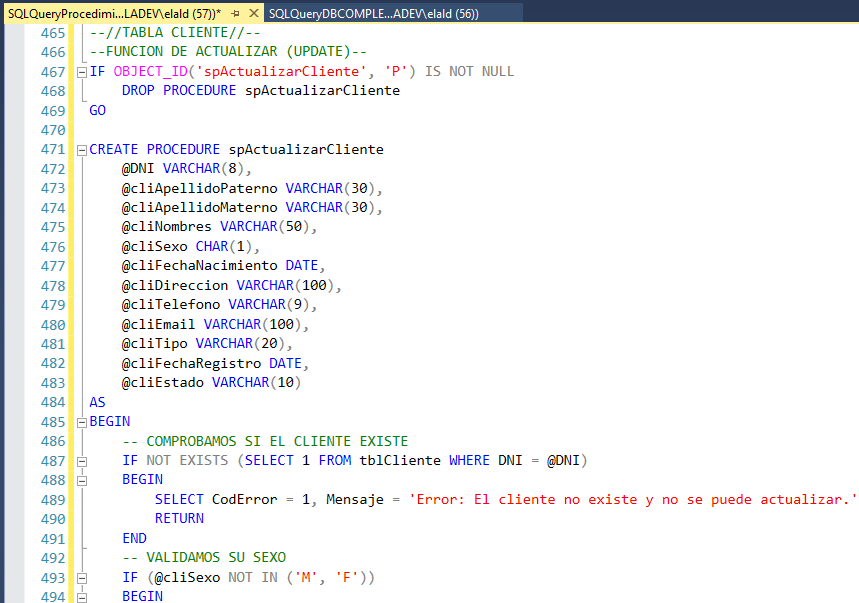


Tabla Cliente

**Imagen 1**



**Imagen 2 Prueba**

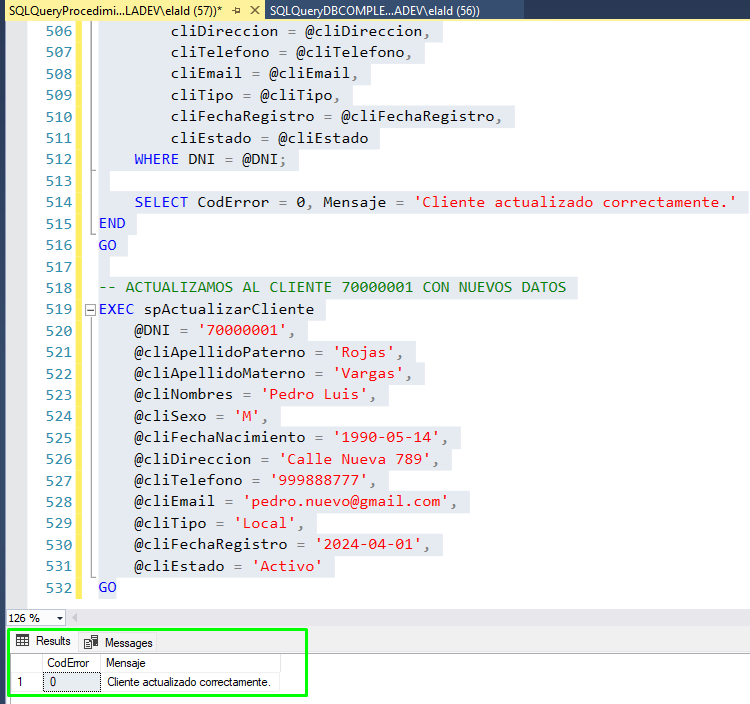


TABLA MOTOS

## 

TABLA REPUESTOS

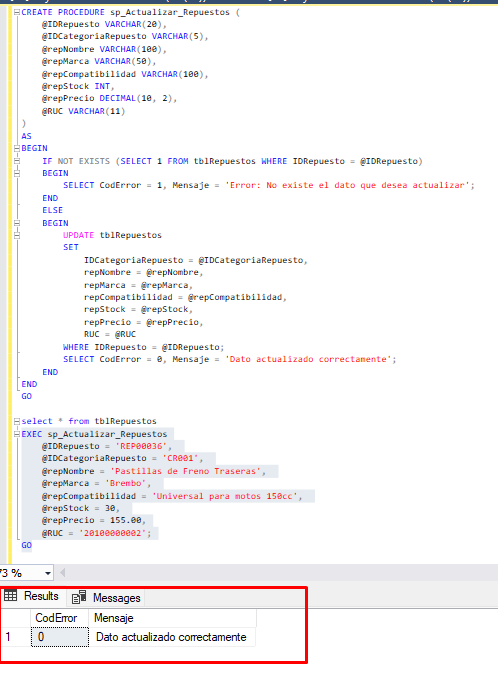
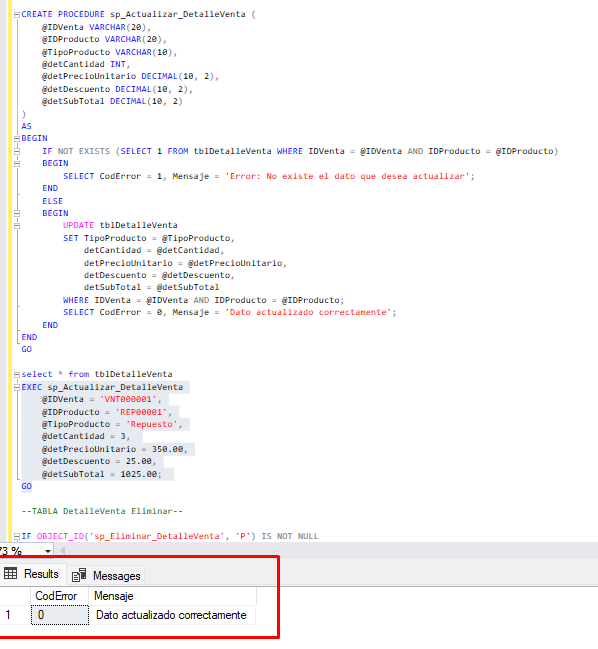


TABLA DETALLE VENTA



## **Procedimiento de Eliminar (DELETE)**

Borrar permanentemente un registro de una tabla. Es la operación de "Eliminar" o "Quitar". Es la operación más delicada. Se usa para eliminar datos que ya no son necesarios. En sistemas complejos, a menudo se prefiere "desactivar" un registro (mediante un UPDATE) en lugar de borrarlo, para mantener la integridad histórica y evitar problemas con datos relacionados.

Tabla de Método de pago

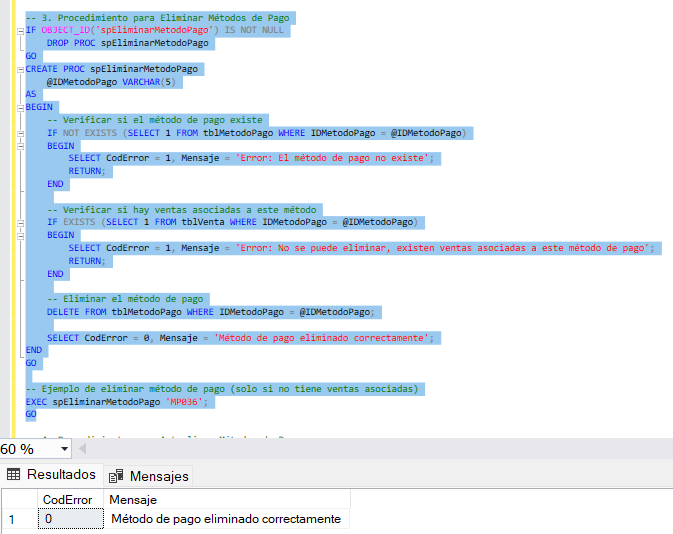
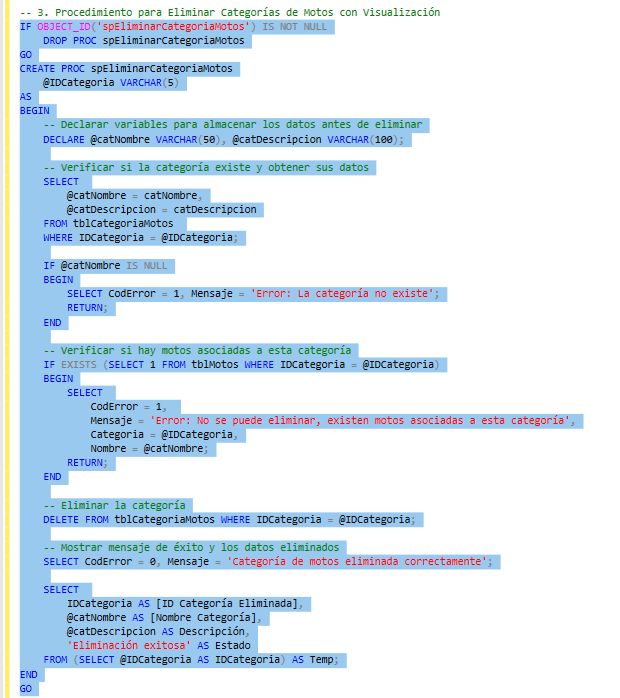


Tabla Categoría Motos



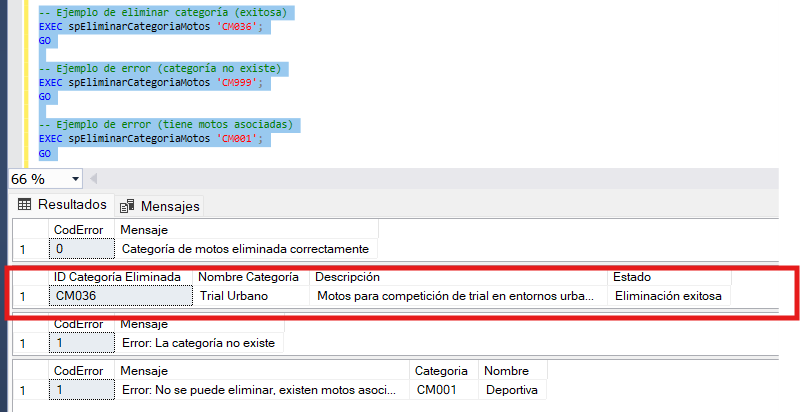


Tabla Categoría Repuesto

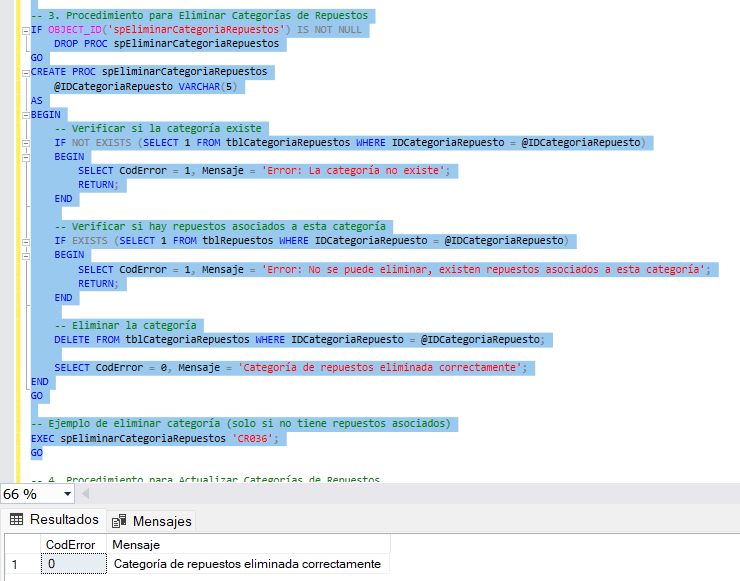
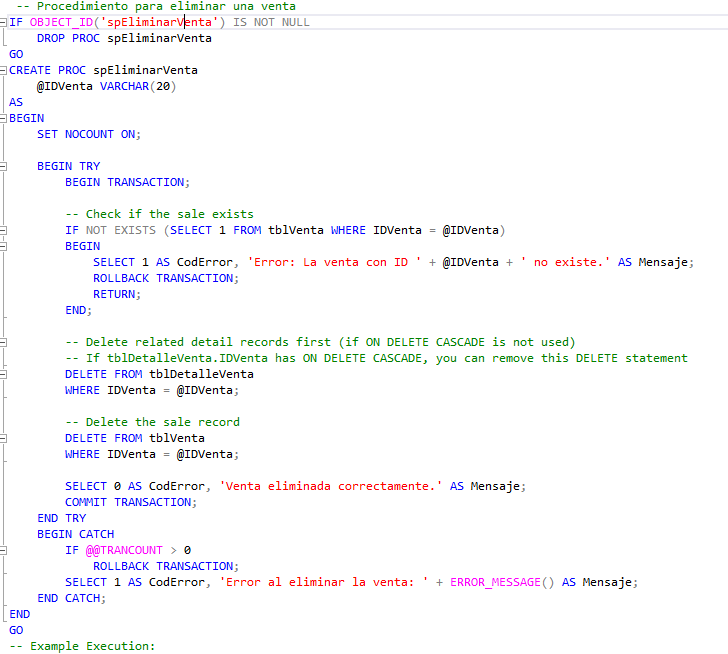
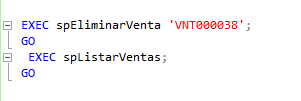


TABLA VENTA



Ejemplo





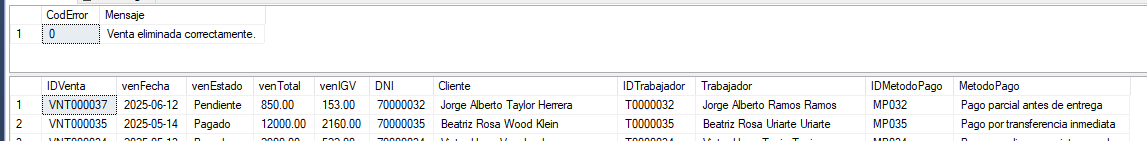
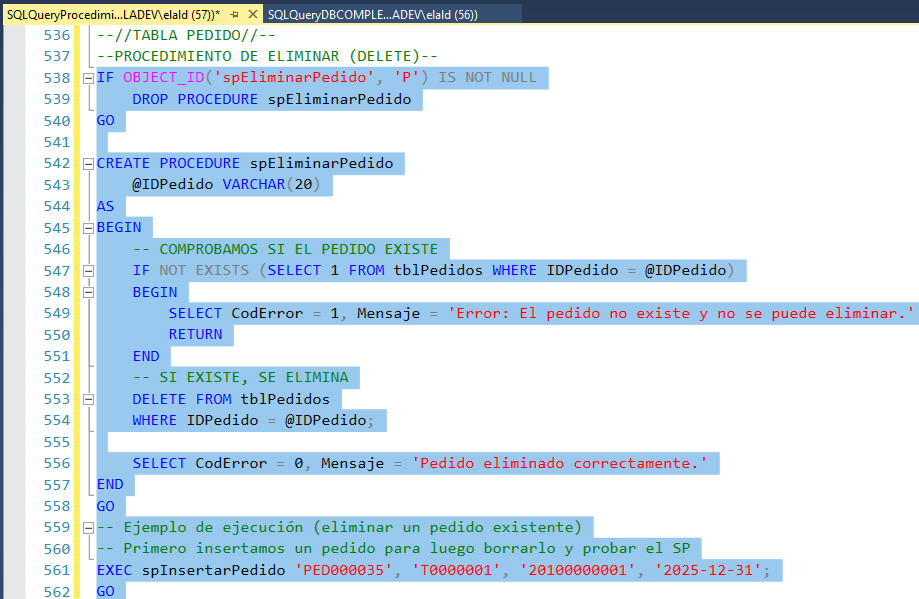


Tabla Pedidos

**Imagen 1**



**Imagen 2 Prueba**

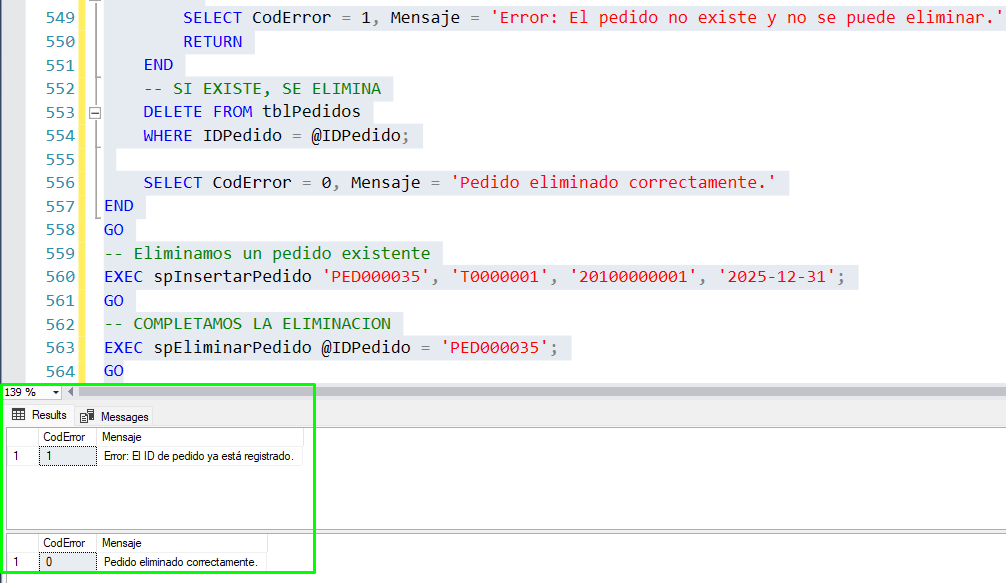


TABLA DetalleVenta

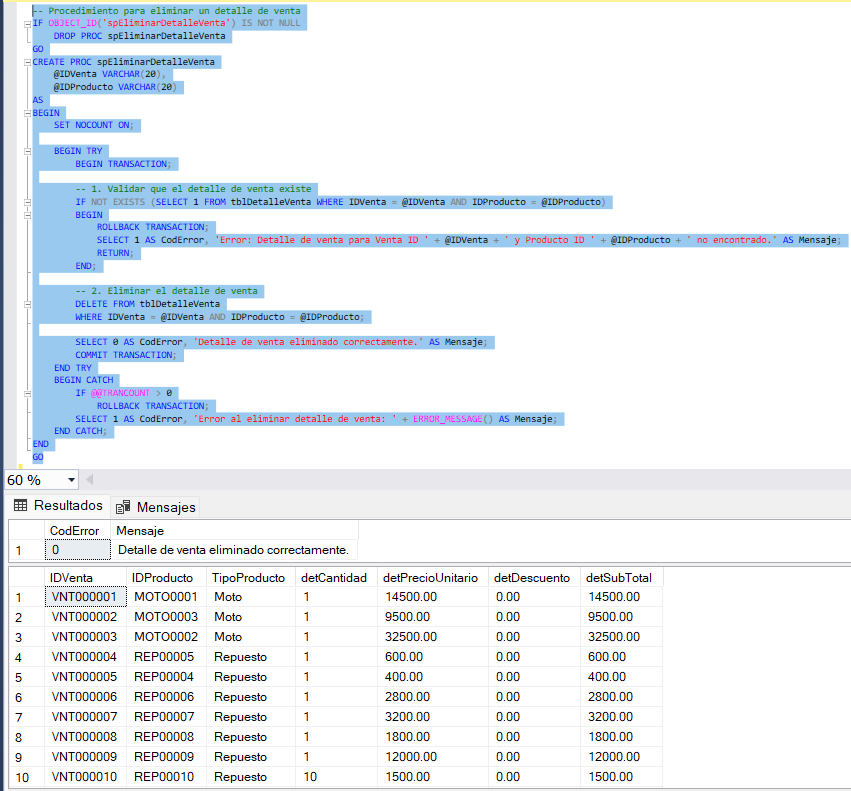
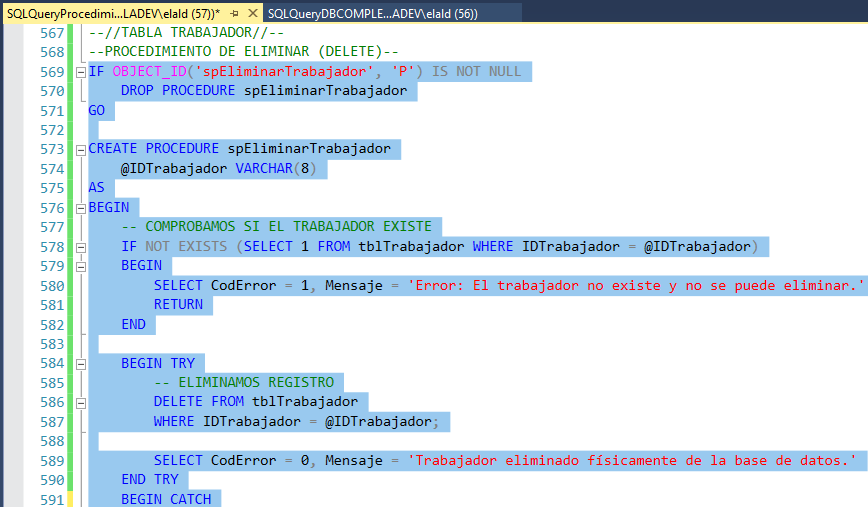


Tabla Trabajador

**Imagen 1**

**Imagen 2 Prueba**

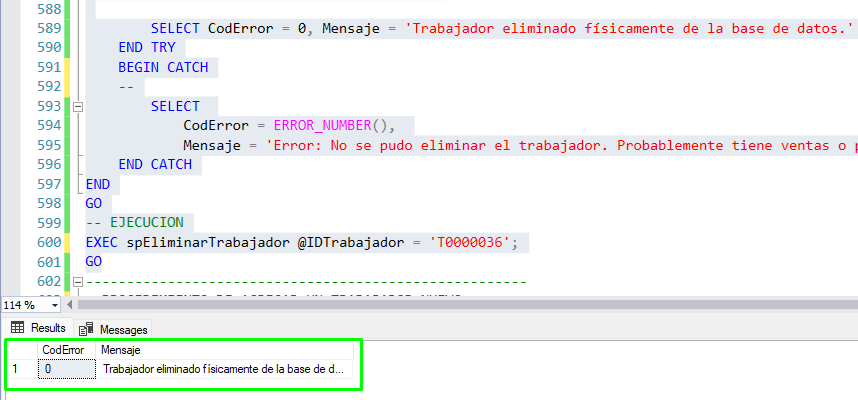
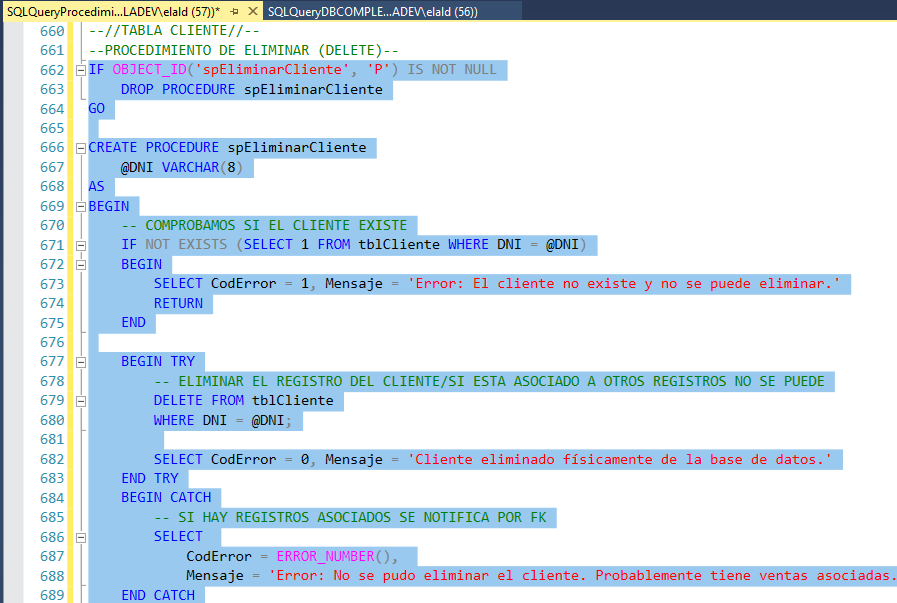


Tabla Cliente

**Imagen 1**



**Imagen 2 Prueba**

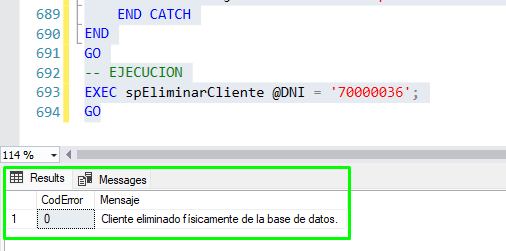


TABLA MOTOS

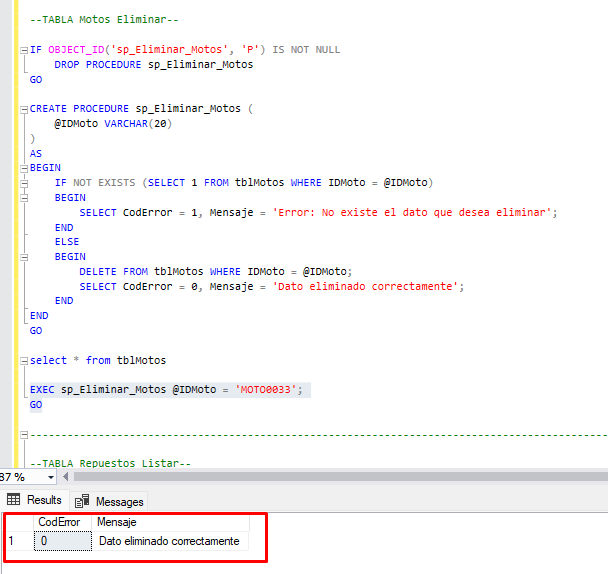


TABLA REPUESTOS

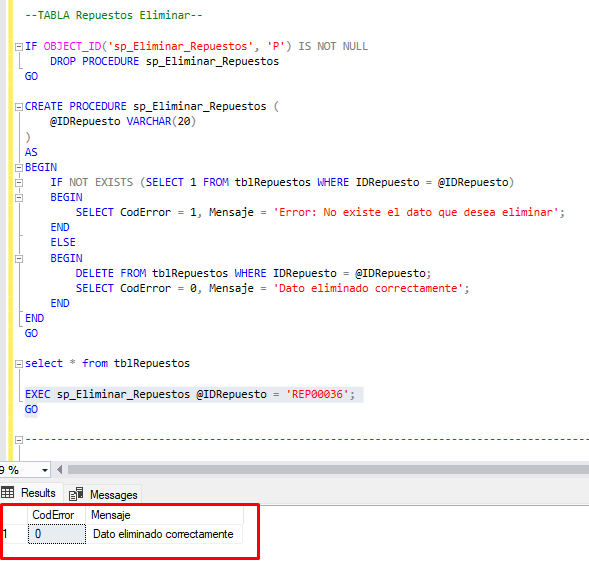
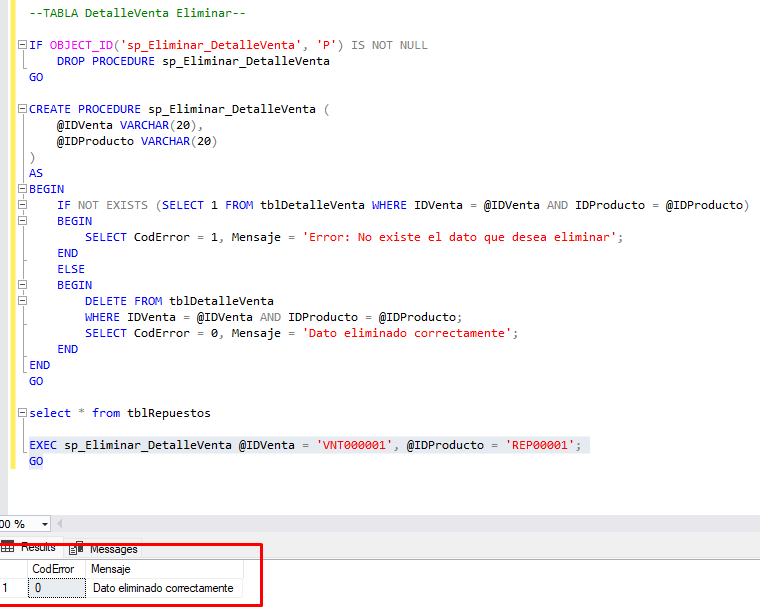
****

TABLA DETALLE VENTA

****

# 

# 

# 

# **Reflexión**

El desarrollo de este proyecto de base de datos para el **Corporativo Valu Calderón E.I.R.L.** ha sido una experiencia enriquecedora que permitió aplicar conocimientos teóricos a un contexto real de gestión de ventas de motos y repuestos. Durante el proceso, enfrentamos desafíos clave:

**Diseño de la Estructura:**La creación del **Diagrama Entidad-Relación (DER)** requirió un análisis exhaustivo de los procesos comerciales para garantizar que todas las entidades (clientes, proveedores, productos, ventas) estuvieran interconectadas de manera eficiente. La normalización fue crucial para evitar redundancias, pero también debemos equilibrar con la necesidad de rendimiento en consultas frecuentes (ej: reportes de ventas).

**Validación de Datos:**Los procedimientos almacenados CRUD (**Crear, Leer, Actualizar, Eliminar**) incorporaron validaciones complejas (ej: formato de IDs, unicidad de nombres, integridad referencial). Esto aseguró la calidad de los datos, aunque implicó un mayor esfuerzo en la depuración de errores sintácticos y lógicos.

**Dificultades Técnicas:**Errores en la sintaxis SQL (ej: paréntesis mal cerrados en INSERT) y discrepancias en los nombres de procedimientos (como spAgregarCategoriaMotos vs. spAgregarCategoriaVotos) destacaron la importancia de estandarizar la nomenclatura desde el inicio. La gestión de transacciones en operaciones críticas (ej: eliminar ventas con detalles asociados) requirió el uso de BEGIN TRY/CATCH para manejar fallos sin comprometer la base de datos.

**Aprendizajes Clave:**La documentación detallada (como el diccionario de datos) fue importante para mantener la coherencia entre el equipo. La elección de **SQL Server** como motor de base de datos demostró ser acertada por su soporte a transacciones ACID y escalabilidad, aunque en futuros proyectos podrían explorarse alternativas como PostgreSQL para costos reducidos.

# 

# **Conclusiones**

El objetivo principal de nuestro informe fue desarrollar una interfaz en el Visual Studio 2022, junto con el SQL Management Studio para el de acceso a datos, segura y estandarizada para BD Valu Calderon. Esto se logró mediante la creación de un conjunto completo de procedimientos almacenados para gestionar las operaciones fundamentales.

El diseño de los diagramas E-R, Lógico nos demostró ser eficiente, para el diseño y estructuración de las tablas, entidades, atributos, relaciones, etc. Posterior a ello el entendimiento en clases sobre el CRUD y procedimientos almacenados nos ayudo para añadir nuevos productos o proveedores fue un proceso sencillo y sin complicaciones. Además, el uso de claves foráneas y restricciones (como CHECK y NOT NULL) aseguró la integridad de los datos, reduciendo errores manuales y manteniendo la consistencia de la información.

La implementación de estos procedimientos almacenados ha establecido una capa de lógica de negocio directamente en la base de datos. Esto centraliza las reglas, mejora el rendimiento y, lo más importante, aumenta significativamente la seguridad al limitar el acceso directo a las tablas.

El sistema ahora cuenta con una base sólida y organizada para la gestión de datos, lista para ser integrada de manera segura y eficiente con cualquier aplicación o interfaz de usuario.

# **Referencias**

*CRUD (Crear, Leer, Actualizar, Eliminar)*. (s/f). Microsoft.com. Recuperado el 12 de junio de 2025, de <https://learn.microsoft.com/es-es/iis-administration/api/crud>

# **Anexos**

