



**SQL**

**Entrega Proyecto Final** -Base de datos relacional para una empresa de fabricación y distribución de hardware Gamer

**Jhinner Alejandro Patiño Pasquel**

**2025**

## Contenido

1.	Introducción.....	4
2.	Objetivo General. ....	4
3.	Descripción general de la base de datos. ....	4
3.1.	Descripción de las tablas o campos. ....	4
3.1.1	Cliente:.....	4
3.1.2	tienda. ....	5
3.1.3	producto.....	5
3.1.4	centroDistribucion.....	6
3.1.5	venta.....	6
3.1.6	proveedor.....	7
3.1.7	empleado.....	7
3.1.8	envio.....	7
3.2.	Relaciones y claves foráneas:.....	8
3.2.1	Relación entre Venta y Cliente:.....	8
3.2.2	Relación entre Venta y Producto: .....	8
3.2.3.	Relación entre Cliente y Tienda .....	8
3.2.4	Relación entre Tienda y Centro de Distribución:.....	8
3.2.5	Relación entre Tienda y Producto: .....	8
3.2.6	Relación entre Centros de Distribución y Tiendas:.....	9
3.2.7	Relación entre Proveedor y Producto: .....	9
3.2.8	Relación entre Empleado y Centro de Distribución: .....	9
3.2.9	Relación entre Envío y Venta: .....	9
3.2.	Diagrama entidad relación (E-R):.....	9
3.3.	Creación de Vistas: .....	11
3.3.1	Vista de Ventas Detalladas (vw_ventas_detalladas): .....	11
3.3.2	Vista de Rendimiento por Tienda (vw_tiendas_rendimiento): .....	12
3.3.3	Vista de Compras por Cliente (vw_clientes_compra):.....	12
3.3.4	Procedimiento SP_REGISTRAR_CLIENTE: .....	12
3.3.5	Vista de Proveedores y Contactos (vw_proveedores_contacto):.....	12
3.3.6	Vista de Empleados y Salarios (vw_empleados_salario): .....	12
3.3.7	Vista de Envíos y Estado (vw_envios_estado):.....	13
3.4.	Creación de funciones:.....	13

3.4.1 Función para Calcular el Total de Compras de un Cliente (FN_TOTAL_COMPRAS_CLIENTE):.....	13
3.4.2 Función para Obtener la Cantidad Total Vendida de un Producto (FN_CANTIDAD_VENDIDA_PRODUCTO):.....	13
3.4.3 Función para Obtener el Monto de la Venta Más Alta (FN_VENTA_MAYOR): .....	13
3.4.4 Función para Contar el Número de Clientes por Tienda (FN_NUMERO_CLIENTES_POR_TIENDA): .....	13
<b>3.5. Creación de Procedimientos Almacenados:.....</b>	<b>14</b>
3.5.1 Procedimiento SP_REGISTRAR_VENTA: .....	14
3.5.2 Procedimiento SP_ACTUALIZAR_PRECIO_PRODUCTO: .....	14
3.5.3 Procedimiento SP_INFORME_PRODUCTO: .....	15
3.5.4 Procedimiento SP_REGISTRAR_ENVIO: .....	15
<b>3.6. Creación de Triggers: .....</b>	<b>15</b>
3.6.1 TRG_LOG_cliente: .....	15
3.6.2 TRG_LOG_PROVINCIA_2: .....	15
<b>3.7. Creación de Transacciones: .....</b>	<b>16</b>
3.7.1 Registro de un nuevo proveedor y producto:.....	16
3.7.2 Registro de un nuevo empleado y un envío:.....	16

## 1. Introducción.

Este documento presenta el desarrollo de un modelo de base de datos para una empresa dedicada a la fabricación de hardware especializado para computadoras Gamer. A lo largo del informe, se describen las entidades clave, sus atributos y relaciones, así como las vistas, funciones, procedimientos almacenados y triggers diseñados para optimizar la gestión de la información.

El sistema de base de datos propuesto permite administrar de manera eficiente los clientes, tiendas, productos, centros de distribución y ventas, garantizando la integridad de los datos y facilitando su consulta para la toma de decisiones estratégicas dentro de la empresa.

## 2. Objetivo General.

Desarrollar y documentar una base de datos relacional que modele el funcionamiento de una empresa de fabricación y distribución de hardware Gamer, asegurando la correcta estructuración y vinculación de los datos mediante claves primarias y foráneas. Además, se busca optimizar el acceso a la información a través de vistas, procedimientos almacenados y funciones que permitan una gestión eficiente de las ventas, clientes, productos y procesos logísticos.

## 3. Descripción general de la base de datos.

La base de datos modela un sistema para una empresa de fabricación de hardware especializado para computo Gamer. Está organizada en cinco entidades principales: Cliente, Tienda, Producto, Centro de distribución y Venta. Estas entidades representan los diferentes componentes del sistema y las relaciones entre ellas.

### 3.1. Descripción de las tablas o campos.

#### 3.1.1 Cliente:

- **Campos:** código cliente (PK), nombre, teléfono, tipo (persona o empresa), email, Código\_tienda(FK).
- **Grado de la tabla:** 6 (número de campos).
- **Descripción:** Representa a los clientes de la empresa de Hardware. La clave primaria es el código cliente, ya que es único para cada cliente.

Nombre campo	Tipo dato	Llave primaria	Índice secundario	Llave foránea
<b>cedula</b>	int	Sí	No	No
<b>nombre</b>	text	No	Sí	No
<b>teléfono</b>	varchar	No	Sí	No
<b>tipo</b>	enum	No	No	No
<b>email</b>	varchar	No	No	No
<b>codigo_tienda</b>	int	No	No	Sí (Tiendas)

### 3.1.2 tienda.

- **Atributos:** código de tienda (PK), ciudad, tipología (física u online), tamaño en m<sup>2</sup> (0 si es online), Numero de la fábrica(FK).
- **Grado:** 5
- **Descripción:** Representa las tiendas físicas o virtuales de la empresa. La clave primaria es el código de tienda.

Nombre campo	Tipo dato	Llave primaria	Índice secundario	Llave foránea
<b>cod_tienda</b>	int	Sí	No	No
<b>ciudad</b>	text	No	Sí	No
<b>tipologia</b>	enum	No	Sí	No
<b>tamano_mt2</b>	numeric	No	No	No
<b>numero_fabrica</b>	int	No	No	Sí (Centros de distribución)

### 3.1.3 producto.

- **Atributos:** código de producto (PK), nombre, precio unitario.
- **Grado:** 3.

- **Descripción:** Contiene la información de los hardware fabricados por la empresa. La clave primaria es el código del producto.

Nombre campo	Tipo dato	Llave primaria	Índice secundario	Llave foránea
cod_producto	int	Sí	No	No
nombre	text	No	Sí	No
precio_unitario	real	No	No	No

### 3.1.4 centroDistribucion.

- **Atributos:** cod\_fabrica(PK), país, ciudad, número de empleados.
- **Grado:** 4.
- **Descripción:** Representa los centros de distribución que abastecen a las tiendas. La clave primaria es el número de fábrica.

Nombre campo	Tipo dato	Llave primaria	Índice secundario	Llave foránea
cod_fabrica	int	Sí	No	No
pais	text	No	Sí	No
ciudad	text	No	Sí	No
numero_empleados	numeric	No	No	No

### 3.1.5 venta.

- **Atributos:** código de venta (PK), Código\_cliente(FK), Código\_producto (FK), cantidad, precio total, fecha del pedido.
- **Grado:** 6.
- **Descripción:** Registra las ventas realizadas por los clientes. Es una entidad débil porque depende de las entidades **Clientes** y **Productos**. La clave primaria es el código de transacción.

Nombre campo	Tipo dato	Llave primaria	Llave foránea	Índice secundario
cod_venta	varchar	Sí	No	No
codigo_cliente	int	No	Sí (Clientes)	No
codigo_producto	int	No	Sí (Productos)	No
cantidad	numeric	No	No	No
precio_total	real	No	No	No
fecha_pedido	date	No	No	No

### 3.1.6 proveedor.

- **Atributos:** código de proveedor (PK), nombre, contacto, teléfono, email.
- **Grado:** 5.
- **Descripción:** Representa a los proveedores de la empresa, que suministran los materiales y componentes necesarios para la fabricación de productos.

Nombre campo	Tipo dato	Llave primaria	Índice secundario	Llave foránea
cod_proveedor	int	Sí	No	No
nombre	text	No	Sí	No
contacto	text	No	No	No
telefono	varchar	No	No	No
email	varchar	No	No	No

### 3.1.7 empleado.

- **Atributos:** código de empleado (PK), nombre, puesto, salario, cod\_fabrica (FK).
- **Grado:** 5.
- **Descripción:** Contiene información sobre los empleados de los centros de distribución, incluyendo su puesto y salario.

Nombre campo	Tipo dato	Llave primaria	Índice secundario	Llave foránea
cod_empleado	int	Sí	No	No
nombre	text	No	Sí	No
puesto	text	No	No	No
salario	numeric	No	No	No
cod_fabrica	int		No	Sí (CentroDistribucion)

### 3.1.8 envío.

- **Atributos:** código de envío (PK), código de venta (FK), fecha de envío, estado.
- **Grado:** 4.
- **Descripción:** Representa los envíos realizados para las ventas, permitiendo rastrear la fecha y el estado de cada entrega.

Nombre campo	Tipo dato	Llave primaria	Índice secundario	Llave foránea
cod_envio	int	Sí	No	No
cod_venta	varchar	No	No	Sí (Venta)
fecha_envio	date	No	No	No
estado	text	No	No	No

### 3.2. Relaciones y claves foráneas:

#### 3.2.1 Relación entre Venta y Cliente:

- **Clave foránea:** Código\_cliente en Venta
- Cada venta está asociada a un cliente.

#### 3.2.2 Relación entre Venta y Producto:

- **Clave foránea:** Código\_producto en Venta
- Cada venta está asociada a un producto.

#### 3.2.3. Relación entre Cliente y Tienda:

- **Clave foránea:** Código\_tienda en Cliente
- Cada cliente realiza sus compras en una tienda específica.

#### 3.2.4 Relación entre Tienda y Centro de Distribución:

- **Clave foránea:** Número\_fábrica en Tienda
- Cada tienda está asociada a un centro de distribución único.

#### 3.2.5 Relación entre Tienda y Producto:

- No se requiere una clave foránea ya que todas las tiendas venden los mismos productos.



### **3.2.6 Relación entre Centros de Distribución y Tiendas:**

- No hay claves foráneas directas en la tabla de Centros de Distribución, pero la relación se establece a través de la clave primaria Número\_fábrica en la tabla de Tiendas.

### **3.2.7 Relación entre Proveedor y Producto:**

- No hay una clave foránea directa, pero un proveedor puede suministrar múltiples productos.

### **3.2.8 Relación entre Empleado y Centro de Distribución:**

- Clave foránea: cod\_fabrica en Empleado
- Cada empleado está asociado a un centro de distribución.

### **3.2.9 Relación entre Envío y Venta:**

- Clave foránea: cod\_venta en Envío
- Cada envío está asociado a una venta específica.

## **3.2. Diagrama entidad relación (E-R):**

### **Resumen del modelo de datos**

Este diseño proporciona una estructura robusta para gestionar la información de clientes, tiendas, productos, centros de distribución y ventas, con un enfoque en la integridad de los datos y la facilidad para realizar consultas relacionales.

log_auditoria	
ID_LOG	INT
NOMBRE_DE_ACCION	VARCHAR(10)
NOMBRE_TABLA	VARCHAR(50)
USUARIO	VARCHAR(100)
FECHA_UPD_INS_DEL	DATE
Indexes	

log_auditoria_2	
ID_LOG	INT
CAMPO NUEVO_CAMPO ANTERIOR	VARCHAR(30...)
NOMBRE_DE_ACCION	VARCHAR(10)
NOMBRE_TABLA	VARCHAR(50)
USUARIO	VARCHAR(100)
FECHA_UPD_INS_DEL	DATE
Indexes	

proveedor	
cod_proveedor	INT
nombre	TINYTEXT
contacto	TINYTEXT
telefono	VARCHAR(2...)
email	VARCHAR(50)
Indexes	

envio	
cod_envio	INT
cod_venta	INT
fecha_envio	DATE
estado	TINYTEXT
Indexes	

producto	
cod_producto	INT
nombre	TINYTEXT
precio_unitario	DOUBLE
Indexes	

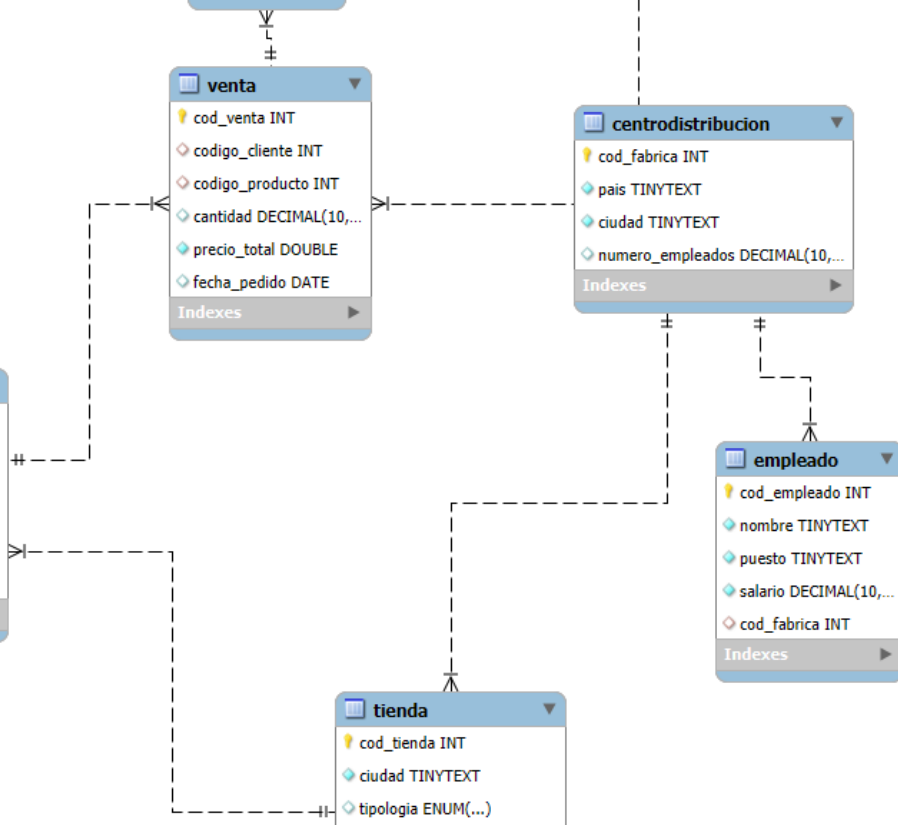
venta	
cod_venta	INT
codigo_cliente	INT
codigo_producto	INT
cantidad	DECIMAL(10,...)
precio_total	DOUBLE
fecha_pedido	DATE
Indexes	

centro_distribucion	
cod_fabrica	INT
pais	TINYTEXT
ciudad	TINYTEXT
numero_empleados	DECIMAL(10,...)
Indexes	

cliente	
cedula	INT
nombre	TINYTEXT
telefono	VARCHAR(3...)
tipo	ENUM(...)
email	VARCHAR(30)
codigo_tienda	INT
Indexes	

empleado	
cod_empleado	INT
nombre	TINYTEXT
puesto	TINYTEXT
salario	DECIMAL(10,...)
cod_fabrica	INT
Indexes	

tienda	
cod_tienda	INT
ciudad	TINYTEXT
tipologia	ENUM(...)
tamano_mt2	DECIMAL(10,...)
numero_fabrica	INT
Indexes	



**1. Relación entre Ventas y Productos (compuesto por):**

- Una venta está compuesta por uno o más productos. Cada producto debe estar asociado a una o más ventas.
- Esto significa que los productos vendidos en una transacción pueden ser variados y, a su vez, un mismo producto puede estar presente en múltiples ventas.

**2. Relación entre Ventas y Clientes (registra venta):**

- Un cliente puede registrar como mínimo una venta (si realizó compras) y como máximo N ventas.
- Una venta puede estar asociada a ningún cliente (cuando se trata de productos reservados o sin comprador específico) o estar asociada como máximo a un cliente que la registre.

**3. Relación entre Tiendas y CentroDistribución (Asociada a):**

- Una tienda está asociada a un único centro de distribución, ya que cada tienda debe depender de un centro logístico que surte sus productos.
- Un centro de distribución puede asociarse con una o más tiendas, dependiendo de su capacidad de abastecimiento.

**4. Relación entre Tiendas y Clientes (Realiza compra):**

- Los clientes realizan compras en las tiendas. Un cliente puede realizar como mínimo una compra en una tienda y como máximo N compras.
- Una tienda puede realizar ninguna venta (cuando no hay clientes o transacciones en un periodo) y puede realizar como máximo ventas a N clientes.

### **3.3. Creación de Vistas:**

**3.3.1 Vista de Ventas Detalladas (vw\_ventas\_detalladas):**

Esta vista combina información de las transacciones (venta), los clientes (cliente) y los productos (producto). Esto se logra a través de sentencias JOIN que relacionan las tablas mediante claves foráneas, permitiendo obtener un reporte consolidado con campos clave como el código de la venta, la fecha, el cliente, el producto, la cantidad vendida y el total de la transacción.

Esta vista simplifica la consulta y el análisis de datos para la empresa, al evitar tener que escribir consultas complejas en cada ocasión y al garantizar que los reportes siempre reflejen la información actualizada de la base de datos.

### **3.3.2 Vista de Rendimiento por Tienda (vw\_tiendas\_rendimiento):**

La Vista de Rendimiento por Tienda es una herramienta que consolida información de distintas fuentes en la base de datos para evaluar el desempeño comercial de cada tienda. Al combinar datos de ventas, detalles de clientes y la relación con los centros de distribución, la vista ofrece un panorama completo del rendimiento de cada tienda, ayudando a identificar oportunidades y desafíos en el proceso de ventas y distribución. Esta vista es útil para análisis gerenciales y para la toma de decisiones estratégicas basadas en datos consolidados y actualizados. Relaciona la tabla de tiendas, centroDistribucion, cliente y venta.

### **3.3.3 Vista de Compras por Cliente (vw\_clientes\_compra):**

tiene como objetivo centralizar información relacionada con los clientes, mostrando datos de contacto y la tienda en la que realizan compras, junto con un resumen de su actividad comercial: la cantidad de ventas registradas y el total acumulado en compras. Esta vista facilita el análisis y la generación de reportes sobre el comportamiento de compra de los clientes. Relaciona las tablas de cliente, tienda y venta.

### **3.3.4 Procedimiento SP\_REGISTRAR\_CLIENTE:**

El procedimiento SP\_REGISTRAR\_CLIENTE está orientado a incorporar nuevos clientes al sistema; al recibir como parámetros el nombre, teléfono, tipo, email y el código de la tienda asociada, inserta un nuevo registro en la tabla cliente, asegurando una incorporación ordenada y validada de clientes para la gestión comercial.

### **3.3.5 Vista de Proveedores y Contactos (vw\_proveedores\_contacto):**

Esta vista permite visualizar los datos de contacto de los proveedores registrados en la base de datos. Incluye información clave como el código del proveedor, su nombre, el contacto principal, el teléfono y el correo electrónico. Esta vista facilita el acceso a los datos de los proveedores, agilizando las consultas y asegurando que los responsables de adquisiciones tengan acceso rápido a la información relevante.

### **3.3.6 Vista de Empleados y Salarios (vw\_empleados\_salario):**

La Vista de Empleados y Salarios consolida información sobre los empleados que trabajan en los centros de distribución. Relaciona a cada empleado con su centro de distribución correspondiente, mostrando detalles sobre su puesto, salario, ciudad y país del centro de distribución. Esta vista es útil para realizar análisis de recursos humanos y evaluar la distribución del personal dentro de la empresa.

### **3.3.7 Vista de Envíos y Estado (vw\_envios\_estado):**

Esta vista proporciona un seguimiento detallado de los envíos realizados dentro del sistema. Relaciona la tabla de envíos con la de ventas, mostrando el código del envío, el código de la venta asociada, la fecha del pedido, la fecha del envío y su estado actual. Es una herramienta clave para el área de logística, permitiendo un monitoreo eficiente de los envíos y garantizando la trazabilidad de las entregas.

## **3.4. Creación de funciones:**

### **3.4.1 Función para Calcular el Total de Compras de un Cliente (FN\_TOTAL\_COMPRAS\_CLIENTE):**

Esta función recibe como parámetro el código de un cliente (cedula) y retorna el total acumulado que ese cliente ha gastado en ventas. Es útil para conocer el comportamiento de compra de cada cliente, identificar a los clientes más valiosos y apoyar en análisis financieros o estrategias de fidelización.

### **3.4.2 Función para Obtener la Cantidad Total Vendida de un Producto (FN\_CANTIDAD\_VENDIDA\_PRODUCTO):**

Esta función toma como parámetro el código de un producto y devuelve la suma total de la cantidad vendida para ese producto, según la información registrada en la tabla venta. Es ideal para evaluar el desempeño de cada producto en el mercado.

### **3.4.3 Función para Obtener el Monto de la Venta Más Alta (FN\_VENTA\_MAYOR):**

tiene como objetivo determinar y devolver el monto de la venta más alta registrada en la tabla venta, lo cual permite identificar la transacción de mayor impacto financiero en el sistema; para lograr esto, la función se declara sin parámetros y especifica un valor de retorno de tipo real, declarando internamente una variable local (max\_venta) en la que se almacena el resultado de una consulta que utiliza la función agregada MAX sobre el campo precio\_total de la tabla venta, envuelta en la función COALESCE para asegurar que, en caso de no existir registros, se retorne un 0 en lugar de un valor NULL, y finalmente, tras ejecutar la consulta y asignar el resultado a la variable, la función retorna dicho valor, consolidando en un único bloque todo el proceso de extracción y cálculo del monto de la venta más alta.

### **3.4.4 Función para Contar el Número de Clientes por Tienda (FN\_NUMERO\_CLIENTES\_POR\_TIENDA):**

tiene como objetivo contar y retornar el número total de clientes asociados a una tienda específica; para ello, recibe como parámetro el código de la tienda, declara internamente

una variable para almacenar el conteo, y ejecuta una consulta que utiliza la función de agregación COUNT en la tabla cliente, filtrando los registros donde el campo de clave foránea que asocia al cliente con la tienda coincide con el código proporcionado; finalmente, la función devuelve el resultado obtenido, permitiendo conocer de manera rápida y precisa la cantidad de clientes asignados a esa tienda.

### **3.5. Creación de Procedimientos Almacenados:**

#### **3.5.1 Procedimiento SP\_REGISTRAR\_VENTA:**

El procedimiento SP\_REGISTRAR\_VENTA está diseñado para registrar una nueva venta en la base de datos de manera automática, ya que recibe como parámetros el código de venta, código del cliente, el código del producto, la cantidad vendida y la fecha del pedido; internamente, declara variables locales para almacenar el precio unitario del producto y el total de la venta, luego consulta en la tabla producto para obtener el precio unitario correspondiente al código de producto proporcionado, calcula el valor total multiplicando dicha cantidad por el precio unitario, y finalmente inserta un nuevo registro en la tabla venta utilizando todos estos datos, asegurando que cada transacción se almacene con la información precisa y en conformidad con las relaciones definidas en la base de datos. Así:

- Se obtiene el precio unitario del producto desde la tabla producto.
- Se calcula el precio total multiplicando la cantidad por el precio unitario.
- Se inserta la nueva venta en la tabla venta con los datos recibidos y calculados.

#### **3.5.2 Procedimiento SP\_ACTUALIZAR\_PRECIO\_PRODUCTO:**

Es un componente clave para la gestión dinámica de precios en la base de datos, ya que permite modificar de manera centralizada el precio unitario de un producto en la tabla producto. Este procedimiento recibe dos parámetros de entrada: uno que identifica de forma única al producto mediante su código (p\_cod\_producto) y otro que especifica el nuevo valor a asignar (p\_nuevo\_precio), lo cual posibilita que, ante cambios en los costos de producción, promociones o ajustes de mercado, se actualice el precio sin tener que reescribir consultas SQL manualmente. Dentro de su estructura, se ejecuta una sentencia UPDATE que altera el campo precio\_unitario del registro correspondiente al código de producto proporcionado, garantizando la integridad de los datos mediante la verificación de la condición en el WHERE. Al agrupar esta operación en un procedimiento almacenado, se asegura la consistencia en la actualización de la información, facilitando su mantenimiento y reutilización en diversas partes de la aplicación o durante procesos batch.

### **3.5.3 Procedimiento SP\_INFORME\_PRODUCTO:**

está diseñado para generar un informe detallado del rendimiento de ventas de un producto específico, recibiendo el código del producto como parámetro de entrada y utilizando parámetros de salida para retornar el total de unidades vendidas, el ingreso total acumulado y el precio promedio ponderado de venta; internamente, el procedimiento ejecuta una consulta en la tabla venta que suma la cantidad y el monto de ventas asociados al producto, almacena estos resultados en variables temporales, y luego asigna esos valores a los parámetros de salida, calculando el precio promedio solo si la cantidad total es mayor a cero, lo que permite obtener de forma rápida y consolidada una visión cuantitativa del desempeño comercial del producto para apoyar decisiones estratégicas

### **3.5.4 Procedimiento SP\_REGISTRAR\_ENVIO:**

está diseñado para registrar un nuevo envío en la base de datos de manera automática. Recibe como parámetros el código de la venta asociada, la fecha en la que se realiza el envío y el estado del mismo.

Internamente, este procedimiento inserta un nuevo registro en la tabla envío, asegurando que cada envío esté correctamente vinculado a una venta y proporcionando trazabilidad dentro del sistema logístico de la empresa. Se recibe el código de la venta como referencia para el envío. Se almacena la fecha del envío para su seguimiento. Se registra el estado del envío, permitiendo indicar si está 'Pendiente', 'En tránsito' o 'Entregado'.

## **3.6. Creación de Triggers:**

### **3.6.1 TRG\_LOG\_cliente:**

Su objetivo es únicamente registrar en la tabla LOG\_AUDITORIA que se realizó una inserción en la tabla cliente, guardando la acción ('INSERT'), el nombre de la tabla ('cliente'), el usuario que ejecutó la acción (CURRENT\_USER()) y la fecha y hora (NOW()).

### **3.6.2 TRG\_LOG\_PROVINCIA\_2:**

Se activa después de cada inserción en la tabla cliente de la base de datos EmpresaHardware; para cada nuevo registro (accesible mediante la palabra clave NEW), concatena los valores de sus campos relevantes (como cod\_cliente, nombre, telefono, tipo, email y codigo\_tienda) en una cadena única que se inserta en el campo CAMPONUEVO\_CAMPOANTERIOR de la tabla LOG\_AUDITORIA\_2, junto con información adicional: la acción realizada (INSERT), el nombre de la tabla afectada (cliente), el usuario actual que ejecuta la operación (CURRENT\_USER()), y la fecha y hora de la inserción (NOW()); de esta manera, se registra automáticamente un log detallado para auditoría cada vez que se inserta un nuevo cliente en la tabla

### **3.7. Creación de Transacciones:**

#### **3.7.1 Registro de un nuevo proveedor y producto:**

Su objetivo es registrar un nuevo proveedor en la tabla proveedor y, simultáneamente, añadir un nuevo producto a la tabla producto.

#### **3.7.2 Registro de un nuevo empleado y un envío:**

Su objetivo es insertar un nuevo empleado en la tabla empleado y, al mismo tiempo, registrar un nuevo envío en la tabla envió.