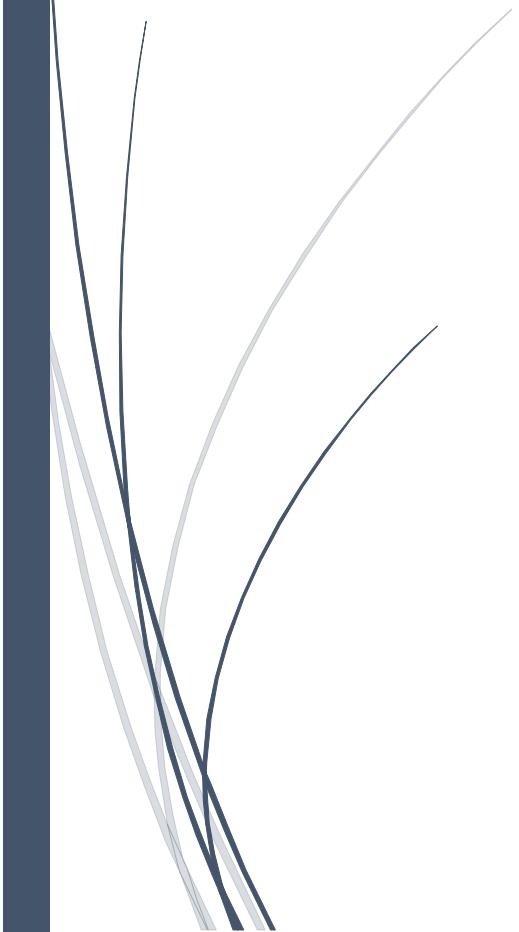




27-11-2025

Entrega FINAL

Logística WayCool

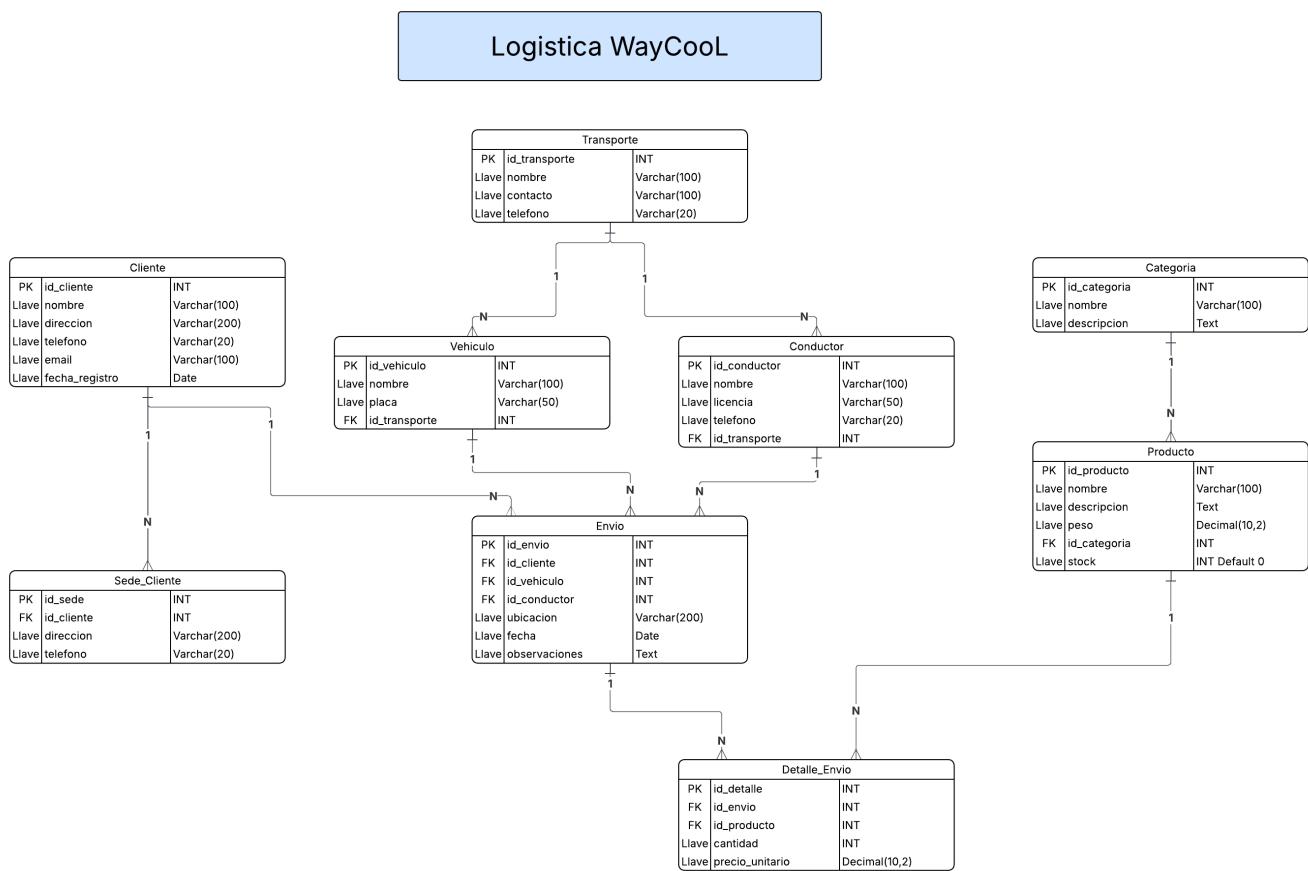


Jorge Adolfo Leitner

Problemática:

Una empresa de logística en crecimiento enfrentaba caos operativo: no podía rastrear envíos en tiempo real, los clientes reclamaban por demoras sin información clara, los conductores asignados a vehículos inadecuados, y los inventarios de productos transportados se descontrolaban. La falta de integración entre datos de clientes, flota de vehículos, conductores y productos generaba pérdidas económicas y mala reputación. Era urgente implementar un sistema que relacionara todas estas variables para optimizar rutas, asignar recursos eficientemente y brindar visibilidad completa del ciclo logístico.

Diagrama Entidad Relación:



Listado de Tablas:

Tabla Cliente:

Campo	Tipo Dato	PK	FK	Null
id_cliente	INT	X		
nombre	VARCHAR(100)			
direccion	VARCHAR(200)			
telefono	VARCHAR(20)			
email	VARCHAR(100)			
fecha_registro	DATE			

Tabla Sede_Cliente:

Campo	Tipo Dato	PK	FK	Null
id_sede	INT	x		
id_cliente	INT		x	
direccion	VARCHAR(200)			
telefono	VARCHAR(20)			

Tabla Transporte:

Campo	Tipo Dato	PK	FK	Null
id_transporte	INT	X		
nombre	VARCHAR(100)			
contacto	VARCHAR(100)			
telefono	VARCHAR(20)			

Tabla Vehículo:

Campo	Tipo Dato	PK	FK	Null
id_vehiculo	INT	x		
nombre	VARCHAR(100)			
placa	VARCHAR(50)			
id_transporte	INT		x	

Tabla Conductor:

Campo	Tipo Dato	PK	FK	Null
id_conductor	INT	x		
nombre	VARCHAR(100)			
licencia	VARCHAR(50)			
telefono	VARCHAR(20)			
id_transporte	INT		x	

Tabla Envío:

Campo	Tipo Dato	PK	FK	Null
id_envio	INT	x		
id_cliente	INT		x	
id_vehiculo	INT		x	
id_conductor	INT		x	
ubicacion	VARCHAR(200)			
fecha	DATE			x
observaciones	TEXT			x

Tabla Producto:

Campo	Tipo Dato	PK	FK	Null
id_producto	INT	x		
nombre	VARCHAR(100)			
descripción	TEXT			x
peso	DECIMAL(10,2)			x
id_categoria	INT		x	x
stock	INT			

Tabla Categoria:

Campo	Tipo Dato	PK	FK	Null
id_categoria	INT	x		
nombre	VARCHAR(100)			
descripción	TEXT			x

Tabla Detalle_Envío:

Campo	Tipo Dato	PK	FK	Null
id_detalle	INT	x		
id_envio	INT		x	
id_producto	INT		x	
cantidad	INT			
precio_unitario	DECIMAL(10,2)			x

Tabla Auditoria_Stock:

Campo	Tipo Dato	PK	FK	Null
id_auditoria	INT	x		
id_producto	INT			
nombre_producto	VARCHAR(100)			x
operacion	ENUM('INSERT', 'UPDATE', 'DELETE')			
stock_anterior	INT			x
stock_nuevo	INT			x
diferencia	INT			x
usuario	VARCHAR(100)			x
fecha_hora	DATETIME			x
motivo	VARCHAR(200)			x

Resumen de Entidades y sus Relaciones:

ENTIDAD	RELACIONES DIRECTAS	TIPO
Cliente	Sede_Cliente, Envio	1:N
Sede_Cliente	Cliente	N:1
Transporte	Vehiculo, Conductor	1:N
Vehiculo	Transporte, Envio	N:1, 1:N
Conductor	Transporte, Envio	N:1, 1:N
Envio	Cliente, Vehiculo, Conductor, Detalle_Envio	N:1, N:1, N:1, 1:N
Producto	Categoría, Detalle_Envio	N:1, 1:N
Categoría	Producto	1:N
Detalle_Envio	Envio, Producto	N:1, N:1
Auditoria_Stock	Producto (referencia histórica)	N:1 (sin FK)

Relaciones Definidas:

1. Cliente - Sede_Cliente (1:N)

- **Tipo:** Uno a Muchos
- **Descripción:** Un cliente puede tener múltiples sedes, pero cada sede pertenece a un solo cliente
- **Clave foránea:** Sede_Cliente.id_cliente → Cliente.id_cliente

2. Transporte - Vehiculo (1:N)

- **Tipo:** Uno a Muchos

- **Descripción:** Un tipo de transporte puede tener múltiples vehículos, pero cada vehículo pertenece a un solo tipo de transporte
- **Clave foránea:** `Vehiculo.id_transporte → Transporte.id_transporte`

3. Transporte - Conductor (1:N)

- **Tipo:** Uno a Muchos
- **Descripción:** Un tipo de transporte puede tener múltiples conductores, pero cada conductor trabaja para un solo tipo de transporte
- **Clave foránea:** `Conductor.id_transporte → Transporte.id_transporte`

4. Cliente - Envio (1:N)

- **Tipo:** Uno a Muchos
- **Descripción:** Un cliente puede realizar múltiples envíos, pero cada envío pertenece a un solo cliente
- **Clave foránea:** `Envio.id_cliente → Cliente.id_cliente`

5. Vehiculo - Envio (1:N)

- **Tipo:** Uno a Muchos
- **Descripción:** Un vehículo puede realizar múltiples envíos, pero cada envío es transportado por un solo vehículo
- **Clave foránea:** `Envio.id_vehiculo → Vehiculo.id_vehiculo`

6. Conductor - Envio (1:N)

- **Tipo:** Uno a Muchos
- **Descripción:** Un conductor puede realizar múltiples envíos, pero cada envío es conducido por un solo conductor
- **Clave foránea:** `Envio.id_conductor → Conductor.id_conductor`

7. Categoria - Producto (1:N)

- **Tipo:** Uno a Muchos
- **Descripción:** Una categoría puede contener múltiples productos, pero cada producto pertenece a una sola categoría
- **Clave foránea:** `Producto.id_categoria → Categoria.id_categoria`

8. Envio - Detalle_Envio (1:N)

- **Tipo:** Uno a Muchos
- **Descripción:** Un envío puede contener múltiples detalles (productos), pero cada detalle pertenece a un solo envío
- **Clave foránea:** `Detalle_Envio.id_envio → Envio.id_envio`

9. Producto - Detalle_Envio (1:N)

- **Tipo:** Uno a Muchos
- **Descripción:** Un producto puede estar en múltiples detalles de envío, pero cada detalle corresponde a un solo producto
- **Clave foránea:** `Detalle_Envio.id_producto → Producto.id_producto`

Relación Muchos a Muchos (N:M) Indirecta:

Envio - Producto (N:M)

- **Tipo:** Muchos a Muchos (a través de tabla intermedia)
- **Descripción:** Un envío puede contener múltiples productos, y un producto puede estar en múltiples envíos
- **Tabla intermedia:** Detalle_Envio
- **Implementación:**
 - Envio → Detalle_Envio (1:N)
 - Producto → Detalle_Envio (1:N)

10. Producto - Auditoria_Stock (1:N)

Descripción de la Relación:

- **Tipo:** Uno a Muchos (1:N)
- **Descripción:** Un producto puede tener múltiples registros de auditoría (cada vez que cambia su stock), pero cada registro de auditoría corresponde a un solo producto
- **Clave foránea:** Auditoria_Stock.id_producto → Producto.id_producto

Nota: La clave foránea Auditoria_Stock.id_producto → Producto.id_producto

NO está implementada como CONSTRAINT para preservar el historial de auditoría incluso si se eliminan productos.

Creación de Vistas:

1. vw_envios_clientes

- **Qué hace:** Muestra los envíos con información básica del cliente y el vehículo
- **Join:** Envío → Cliente → Vehículo
- **Columnas clave:** id_envio, fecha, ubicación, nombre_cliente, telefono_cliente, vehiculo, placa_vehiculo
- **Para qué sirve:** Para ver rápidamente quién envió algo, cuándo y con qué vehículo

2. vw_vehiculos_transporte

- **Qué hace:** Muestra todos los vehículos con su tipo de transporte
- **Join:** Vehículo → Transporte
- **Columnas clave:** id_vehiculo, nombre_vehiculo, placa, tipo_transporte, contacto_transporte
- **Para qué sirve:** Para saber qué vehículos pertenecen a qué tipo de transporte

3. vw_productos_categoria

- **Qué hace:** Muestra los productos con su categoría correspondiente
- **Join:** Producto → Categoría

- **Columnas clave:** id_producto, nombre_producto, descripcion, peso, stock, categoria
- **Para qué sirve:** Para ver qué productos hay en cada categoría

4. vw_historial_stock

- **Qué hace:** Muestra el historial completo de cambios de stock
- **Join:** Auditoria_Stock (tabla única)
- **Columnas clave:** id_auditoria, id_producto, nombre_producto, operacion, stock_anterior, stock_nuevo, diferencia, usuario, fecha_hora, motivo, tipo_movimiento
- **Para qué sirve:** Para auditar todos los cambios de inventario y saber quién, cuándo y por qué cambió el stock

Creación de Funciones:

1. fn_total_envios_cliente

- **Qué hace:** Cuenta cuántos envíos ha realizado un cliente específico
- **Parámetro:** p_id_cliente (el ID del cliente)
- **Retorna:** Número total de envíos (INT)
- **Lógica simple:** Solo cuenta los registros en la tabla Envio para ese cliente

Ejemplos de Uso:

-- Ver cuántos envíos ha hecho el cliente con ID 1 --

```
SELECT fn_total_envios_cliente(1) AS total_envios;
```

-- Mostrar todos los clientes con su total de envíos --

```
SELECT
    nombre,
    fn_total_envios_cliente(id_cliente) AS total_envios
FROM Cliente;
```

2. fn_valor_total_envio

- **Qué hace:** Calcula el valor total de un envío sumando todos sus productos
- **Parámetro:** p_id_envio (el ID del envío)
- **Retorna:** Valor total del envío (DECIMAL)
- **Lógica simple:** Multiplica cantidad por precio unitario de cada producto y los suma

Ejemplos de Uso:

-- Ver el valor total del envío con ID 1 --

```
SELECT fn_valor_total_envio(1) AS valor_total;
```

-- Mostrar todos los envíos con su valor total

```
SELECT
```

```
id_envio,  
fecha,  
fn_valor_total_envio(id_envio) AS valor_total  
FROM Envio;
```

Stored Procedures:

1. sp_insertar_cliente

- **Qué hace:** Inserta un nuevo cliente en la base de datos
- **Parámetros:**

p_nombre: Nombre del cliente

p_direccion: Dirección del cliente

p_telefono: Teléfono del cliente

p_email: Email del cliente

- **Lógica:** Simplemente inserta el cliente con fecha actual
- **Retorna:** Mensaje de confirmación

Ejemplos de Uso:

-- Insertar un nuevo cliente --

```
CALL sp_insertar_cliente('Juan Pérez', 'Av. Principal 123', '11-1234-5678', 'juan@email.com');
```

2. sp_actualizar_ubicacion_envio

- **Qué hace:** Cambia la ubicación de un envío existente
- **Parámetros:**

p_id_envio: ID del envío a modificar

p_nueva_ubicacion: Nueva ubicación del envío

- **Lógica:** Actualiza solo el campo ubicación
- **Retorna:** Mensaje de confirmación

Ejemplos de Uso:

-- Cambiar ubicación del envío 1 --

```
CALL sp_actualizar_ubicacion_envio(1, 'Bodega Central');
```

Triggers:

1. tr_fecha_registro_cliente

- **Cuando se activa:** Antes de insertar un nuevo cliente
- **Qué hace:** Establece automáticamente la fecha actual
- **Lógica:** Simplemente asigna la fecha actual al campo fecha_registro

- **Ventaja:** No tienes que preocuparte por poner la fecha manualmente

2. tr_actualizar_stock_envio

- **Cuando se activa:** Despues de insertar un detalle de envío
- **Qué hace:** Resta automáticamente la cantidad enviada del stock
- **Lógica:** Actualiza el stock del producto restando la cantidad enviada
- **Ventaja:** Mantiene el inventario actualizado automáticamente

Triggers adicionales en el código:

- tr_restaurar_stock_envio
 - Restaura stock cuando se elimina un detalle de envío
- tr_modificar_stock_envio
 - Ajusta stock cuando se actualiza un detalle de envío

Tres Triggers de Auditoría:

a) tr_auditoria_stock_insert

- Se activa cuando se **crea** un nuevo producto
- Registra el stock inicial

b) tr_auditoria_stock_update

- Se activa cuando se **modifica** el stock de un producto
- Solo registra si el stock cambió
- Identifica si fue aumento o reducción

c) tr_auditoria_stock_delete

- Se activa cuando se **elimina** un producto
- Registra el stock final antes de eliminar

CREACIÓN DE USUARIOS (DCL.SQL)

1. USUARIO ADMINISTRADOR:

Permisos: Todos los privilegios sobre la base de datos.

2. USUARIO GERENTE:

Permisos: Lectura, Inserción y Modificación (SELECT, INSERT, UPDATE).

No puede borrar registros (DELETE) ni alterar tablas (ALTER/DROP).

3. USUARIO AUDITOR:

Permisos: Solo Lectura (SELECT).

Ideal para reportes y auditoría sin riesgo de modificar datos.

TRANSACCIONES (TCL.SQL)

Reporte de Unidades por Categoría (Reporte.xlsx) :

Suma de stock:

