



**Universidad Nacional Autónoma de México**  
**Facultad de Ingeniería**  
**División de Ingeniería Eléctrica**



**Bases de Datos**

**Proyecto Final**

**Profesor**

M.C. David Ricardo Ruiz Reyes

**Alumnos**

Garces Hernandez Luis Angel  
López Tavera Alexa Fernanda  
Milla Romero Monica Yolanda  
Pacheco Chavarría Arturo Iván

**Grupo**

6

**Semestre**

2025-1

**Fecha de Entrega**

27 de Noviembre, 2024



## Índice

|   |    |
|---|----|
| 1. Enunciado donde se describe el problema.....                     | 3  |
| 2. Diagrama Entidad-Relación.....                                   | 4  |
| 3. Diagrama Relacional.....   | 5  |
| 4. Código final implementado.....                                   | 8  |
| 5. Diccionario de Datos.....  | 18 |
| a) Descripción de cada tabla.....                                   | 18 |
| b) Descripción por columna de cada tabla.....                       | 21 |
| c) Descripción de Procedimientos<br>almacenados y disparadores..... | 25 |

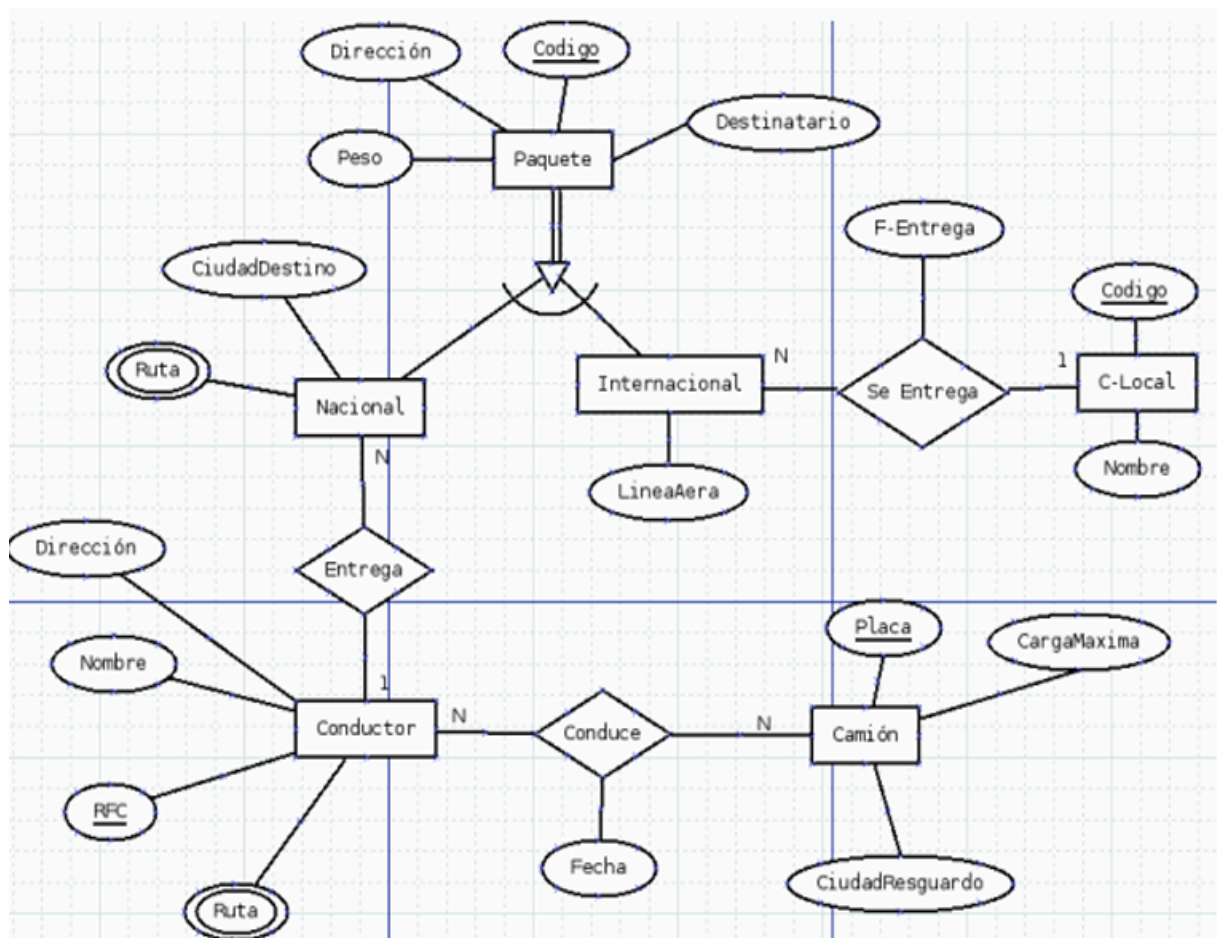
# 1. Enunciado del problema

Una empresa de logística y transporte desea gestionar de manera eficiente el envío de paquetes tanto a nivel nacional como internacional. El sistema debe permitir registrar y organizar los datos de los paquetes, conductores, camiones, rutas y las empresas locales asociadas. Además, se debe garantizar la integridad de los datos y cumplir con las siguientes condiciones:

1. Cada paquete debe ser exclusivamente nacional o internacional, y no puede pertenecer a ambos tipos.
2. Los paquetes nacionales deben incluir información de la ciudad destino y deben actualizar automáticamente un registro de conteo de envíos realizados a cada ciudad.
3. Los paquetes internacionales deben estar vinculados con la línea aérea y la compañía local asociada para su entrega.
4. La asignación de conductores y camiones debe registrarse, especificando la fecha de asignación y asegurando que los camiones cumplan con los límites de carga establecidos (mínimo 250 kg y máximo 1250 kg).
5. Es necesario rastrear cada paquete, permitiendo obtener información como el tipo de envío, peso, conductor asignado, ruta, camión utilizado y placas del camión en el caso de envíos nacionales, o la línea aérea y compañía local en el caso de envíos internacionales.

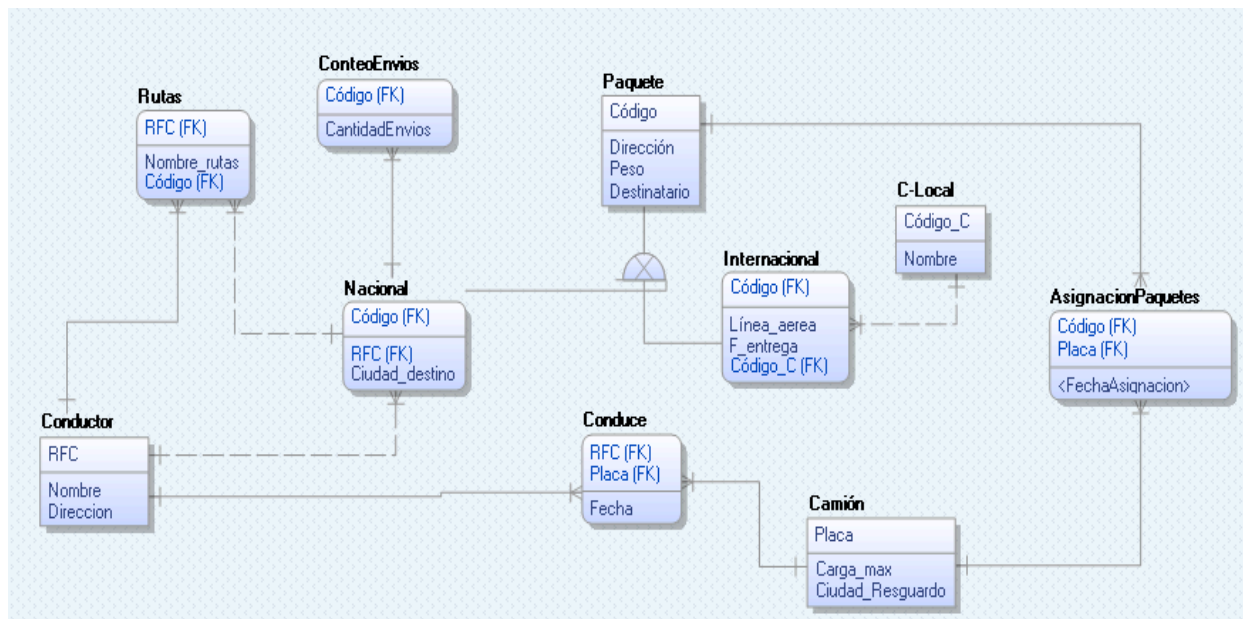
## 2. Modelo Entidad - Relación

El modelo Entidad-Relación presentado es el siguiente:



### 3. Diagrama Relacional

El modelo relacional creado a partir del modelo ER anterior es el siguiente:



Como se puede observar, se han incluido dos nuevas tablas, una de “ConteoEnvios” en la que se pretende registrar cuántos envíos nacionales se han hecho.

Y otra de “AsignacionPaquetes” que hace que se relacionen los paquetes con el camión en el que se transporten.

El código simplificado obtenido a partir del modelo relacional es el siguiente:

*--Tabla para registrar asignaciones de paquetes a camiones*

```
CREATE TABLE AsignacionPaquetes (  
    FechaAsignacion DATETIME NULL,  
   Codigo CHAR(18) NOT NULL, -- FK hacía Paquete  
    Placa VARCHAR(20) NOT NULL, -- FK hacía Camion  
    PRIMARY KEY (Codigo, Placa),  
    CONSTRAINT FK_AsignacionPaquetes_Paquete FOREIGN KEY (Codigo)  
        REFERENCES Paquete (Codigo),  
    CONSTRAINT FK_AsignacionPaquetes_Camion FOREIGN KEY (Placa)  
        REFERENCES Camion (Placa)  
);
```

*-- Tabla de compañías locales para envíos internacionales*

```
CREATE TABLE C_Local (  
    Codigo_C NUMERIC NOT NULL PRIMARY KEY,  
    Nombre VARCHAR(100) NULL  
);
```

*-- Tabla de camiones*

```
CREATE TABLE Camion (  
    Placa VARCHAR(20) NOT NULL PRIMARY KEY,  
    Carga_max NUMERIC NULL,  
    Ciudad_Resguardo VARCHAR(100) NULL,  
    CONSTRAINT CH_Camion_Carga CHECK (Carga_max >= 250 AND Carga_max  
<= 1250)  
);
```

*-- Relación entre conductores y camiones (asignación temporal)*

```
CREATE TABLE Conduce (  
    Fecha DATE NULL,  
    RFC VARCHAR(20) NOT NULL, -- FK hacia Conductor  
    Placa VARCHAR(20) NOT NULL, -- FK hacia Camion  
    PRIMARY KEY (RFC, Placa),  
    CONSTRAINT FK_Conduce_Conductor FOREIGN KEY (RFC)  
        REFERENCES Conductor (RFC),  
    CONSTRAINT FK_Conduce_Camion FOREIGN KEY (Placa)  
        REFERENCES Camion (Placa)  
);
```

-- Tabla de conductores

```
CREATE TABLE Conductor (  
    RFC VARCHAR(20) NOT NULL PRIMARY KEY,  
    Nombre VARCHAR(100) NULL,  
    Direccion VARCHAR(100) NULL  
);
```

-- Tabla para contar envíos nacionales por ciudad

```
CREATE TABLE ConteoEnvios (  
    CiudadDestino VARCHAR(100) NOT NULL PRIMARY KEY,  
    CantidadEnvios INT DEFAULT 0  
);
```

-- Tabla de envíos internacionales

```
CREATE TABLE Internacional (  
   Codigo CHAR(18) NOT NULL PRIMARY KEY, -- FK hacia Paquete  
    Linea_aerea VARCHAR(100) NULL,  
    FechaEntrega DATETIME NULL,  
    Codigo_C NUMERIC NOT NULL, -- FK hacia C_Local  
    CONSTRAINT FK_Internacional_C_Local FOREIGN KEY (Codigo_C)  
        REFERENCES C_Local (Codigo_C),  
    CONSTRAINT FK_Internacional_Paquete FOREIGN KEY (Codigo)  
        REFERENCES Paquete (Codigo) ON DELETE CASCADE  
);
```

-- Tabla de envíos nacionales

```
CREATE TABLE Nacional (  
   Codigo CHAR(18) NOT NULL PRIMARY KEY, -- FK hacia Paquete  
    CiudadDestino VARCHAR(100) NULL,  
    RFC VARCHAR(20) NOT NULL, -- FK hacia Conductor  
    CONSTRAINT FK_Nacional_Conductor FOREIGN KEY (RFC)  
        REFERENCES Conductor (RFC),  
    CONSTRAINT FK_Nacional_Paquete FOREIGN KEY (Codigo)  
        REFERENCES Paquete (Codigo) ON DELETE CASCADE  
);
```

-- Tabla general de paquetes

```
CREATE TABLE Paquete (  
   Codigo CHAR(18) NOT NULL PRIMARY KEY,  
    Direccion VARCHAR(100) NULL,  
    Peso NUMERIC NULL,  
    Destinatario VARCHAR(100) NULL  
);
```

```
-- Tabla de rutas asignadas a conductores
CREATE TABLE Rutas (
  Codigo CHAR(18) NOT NULL, -- FK hacia Nacional
  Nombre_rutas VARCHAR(50) NULL,
  RFC VARCHAR(20) NOT NULL, -- FK hacia Conductor
  PRIMARY KEY (Codigo, RFC),
  CONSTRAINT FK_Rutas_Nacional FOREIGN KEY (Codigo)
    REFERENCES Nacional (Codigo),
  CONSTRAINT FK_Rutas_Conductor FOREIGN KEY (RFC)
    REFERENCES Conductor (RFC)
);
```

Y el código completo que cumple con los requerimientos solicitados es el siguiente:

A continuación se explicará el funcionamiento de cada parte del código

## 4. Código Final Implementado

```
-- Tabla general de paquetes
CREATE TABLE Paquete (
  Codigo CHAR(18) NOT NULL,
  Direccion VARCHAR(100) NOT NULL,
  Peso NUMERIC NOT NULL,
  Destinatario VARCHAR(100) NOT NULL,
  CONSTRAINT Paquete_PK PRIMARY KEY (Codigo)
);

-- Tabla de compañías locales para envíos internacionales
CREATE TABLE C_Local (
  Codigo_C NUMERIC NOT NULL,
  Nombre VARCHAR(100) NOT NULL,
  CONSTRAINT Clocal_PK PRIMARY KEY (Codigo_C)
);

-- Tabla de conductores
CREATE TABLE Conductor (
  RFC VARCHAR(20) NOT NULL,
  Nombre VARCHAR(100) NOT NULL,
  Direccion VARCHAR(100) NOT NULL,
```



```

    CONSTRAINT RFC_PK PRIMARY KEY (RFC)
);

-- Tabla de camiones
CREATE TABLE Camion (
    Placa VARCHAR(20) NOT NULL,
    Carga_max NUMERIC NOT NULL,
    Ciudad_Resguardo VARCHAR(100) NOT NULL,
    CONSTRAINT Placa_PK PRIMARY KEY (Placa),
    CONSTRAINT CH_Camion_Carga CHECK (Carga_max >= 250 AND Carga_max
<= 1250)
);

-- Relación entre conductores y camiones (asignación temporal)
CREATE TABLE Conduce (
    Fecha DATE NOT NULL,
    RFC VARCHAR(20) NOT NULL, -- FK hacia Conductor
    Placa VARCHAR(20) NOT NULL, -- FK hacia Camion
    CONSTRAINT Conduce_PK PRIMARY KEY (RFC, Placa),
    CONSTRAINT FK_Conduce_Conductor FOREIGN KEY (RFC)
        REFERENCES Conductor (RFC),
    CONSTRAINT FK_Conduce_Camion FOREIGN KEY (Placa)
        REFERENCES Camion (Placa)
);

-- Tabla para contar envíos nacionales por ciudad
CREATE TABLE ConteoEnvios (
    CiudadDestino VARCHAR(100) NOT NULL PRIMARY KEY,
    CantidadEnvios INT DEFAULT 0
);

-- Tabla de envíos nacionales
CREATE TABLE Nacional (
   Codigo CHAR(18) NOT NULL, -- FK hacia Paquete
    CiudadDestino VARCHAR(100) NOT NULL,
    RFC VARCHAR(20) NOT NULL, -- FK hacia Conductor
    CONSTRAINT Nacional_PK PRIMARY KEY (Codigo),
    CONSTRAINT FK_Nacional_Conductor FOREIGN KEY (RFC)
        REFERENCES Conductor (RFC),
    CONSTRAINT FK_Nacional_Paquete FOREIGN KEY (Codigo)
        REFERENCES Paquete (Codigo) ON DELETE CASCADE
);

-- Tabla de envíos internacionales

```

```

CREATE TABLE Internacional (
    Codigo CHAR(18) NOT NULL, -- FK hacia Paquete
    Linea_aerea VARCHAR(100) NOT NULL,
    FechaEntrega DATETIME NOT NULL,
    Codigo_C NUMERIC NOT NULL, -- FK hacia C_Local
    CONSTRAINT Internacional_PK PRIMARY KEY (Codigo),
    CONSTRAINT FK_Internacional_C_Local FOREIGN KEY (Codigo_C)
        REFERENCES C_Local (Codigo_C),
    CONSTRAINT FK_Internacional_Paquete FOREIGN KEY (Codigo)
        REFERENCES Paquete (Codigo) ON DELETE CASCADE
);

-- Tabla de rutas asignadas a conductores
CREATE TABLE Rutas (
    Codigo CHAR(18) NOT NULL, -- FK hacia Nacional
    Nombre_rutas VARCHAR(50) NOT NULL, -- Ruta específica
    RFC VARCHAR(20) NOT NULL, -- FK hacia Conductor
    CONSTRAINT Rutas_PK PRIMARY KEY (Codigo, Nombre_rutas, RFC),
    CONSTRAINT FK_Rutas_Nacional FOREIGN KEY (Codigo) REFERENCES
Nacional (Codigo),
    CONSTRAINT FK_Rutas_Conductor FOREIGN KEY (RFC) REFERENCES
Conductor (RFC),
    CONSTRAINT CH_NombreRutas CHECK (Nombre_rutas IN ('Ruta A', 'Ruta B',
'Ruta C', 'Ruta D', 'Ruta E', 'Ruta F', 'Ruta G', 'Ruta H', 'Ruta I', 'Ruta J'))
);

-- Tabla para registrar asignaciones de paquetes a camiones
CREATE TABLE AsignacionPaquetes (
    FechaAsignacion DATETIME NOT NULL,
    Codigo CHAR(18) NOT NULL, -- FK hacia Paquete
    Placa VARCHAR(20) NOT NULL, -- FK hacia Camion
    CONSTRAINT AsignacionPaquetes_PK PRIMARY KEY (Codigo, Placa),
    CONSTRAINT FK_AsignacionPaquetes_Paquete FOREIGN KEY (Codigo)
        REFERENCES Paquete (Codigo),
    CONSTRAINT FK_AsignacionPaquetes_Camion FOREIGN KEY (Placa)
        REFERENCES Camion (Placa)
);

-- Inserciones en las tablas principales

-- Paquete
INSERT INTO Paquete (Codigo, Direccion, Peso, Destinatario)
VALUES
('PKG001', 'Calle 1, Ciudad A', 10.5, 'Juan Pérez'),

```

```

('PKG002', 'Calle 2, Ciudad B', 15.0, 'Ana López'),
('PKG003', 'Calle 3, Ciudad C', 8.3, 'Luis Gómez'),
('PKG004', 'Calle 4, Ciudad D', 12.2, 'María Torres'),
('PKG005', 'Calle 5, Ciudad E', 20.0, 'Diego Becerra'),
('PKG006', 'Calle 6, Ciudad F', 1.5, 'David Ruiz'),
('PKG007', 'Calle 7, Ciudad G', 9.0, 'Alisson Del Razo'),
('PKG008', 'Calle 8, Ciudad H', 14.6, 'Yolanda Milla');
INSERT INTO Paquete (Codigo, Direccion, Peso, Destinatario)
VALUES
('PKG009', 'Calle 9, Ciudad E', 9.0, 'Claudia Gómez'),
('PKG010', 'Calle 10, Ciudad F', 7.5, 'Miguel Ángel'),
('PKG011', 'Calle 11, Ciudad G', 11.3, 'Roberto Ruiz'),
('PKG012', 'Calle 12, Ciudad H', 12.5, 'Sandra Torres');
INSERT INTO Paquete (Codigo, Direccion, Peso, Destinatario)
VALUES
('PKG013', 'Calle 13, Ciudad A', 10.0, 'María García'),
('PKG014', 'Calle 14, Ciudad SinNombre', 15.0, 'Luis Hernández');

```

```

-- C_Local
INSERT INTO C_Local (Codigo_C, Nombre)
VALUES
(1001, 'Almacén Internacional 1'),
(1002, 'Almacén Internacional 2'),
(1003, 'Almacén Internacional 3'),
(1004, 'Almacén Internacional 4');

```

```

-- Conductor
INSERT INTO Conductor (RFC, Nombre, Direccion)
VALUES
('RFC001', 'Pedro Sánchez', 'Calle A, Ciudad X'),
('RFC002', 'Laura Hernández', 'Calle B, Ciudad Y'),
('RFC003', 'Carlos Díaz', 'Calle C, Ciudad Z'),
('RFC004', 'Sofía Jiménez', 'Calle D, Ciudad W'),
('RFC005', 'Marta Gómez', 'Calle F, Ciudad U'),
('RFC006', 'Jorge Pérez', 'Calle G, Ciudad T'),
('RFC007', 'Adriana Ramírez', 'Calle H, Ciudad S'),
('RFC008', 'Fernando López', 'Calle I, Ciudad R');

```

```

-- Camion
INSERT INTO Camion (Placa, Carga_max, Ciudad_Resguardo)
VALUES
('ABC123', 800, 'Ciudad X'),
('DEF456', 1000, 'Ciudad Y'),
('GHI789', 1200, 'Ciudad Z'),

```

```
('JKL012', 900, 'Ciudad W'),  
( 'MNO345', 850, 'Ciudad X'),  
( 'PQR678', 950, 'Ciudad Y'),  
( 'STU901', 1100, 'Ciudad Z'),  
( 'VWX234', 1050, 'Ciudad W');
```

-- Conduce

```
INSERT INTO Conduce (Fecha, RFC, Placa)  
VALUES  
( '2024-11-05', 'RFC005', 'MNO345'),  
( '2024-11-06', 'RFC006', 'PQR678'),  
( '2024-11-07', 'RFC007', 'STU901'),  
( '2024-11-08', 'RFC008', 'VWX234');
```

-- Nacional

```
INSERT INTO Nacional (Codigo, CiudadDestino, RFC)  
VALUES  
( 'PKG001', 'Ciudad A', 'RFC001'),  
( 'PKG002', 'Ciudad B', 'RFC002'),  
( 'PKG003', 'Ciudad C', 'RFC003'),  
( 'PKG004', 'Ciudad D', 'RFC004');
```

-- Internacional

```
INSERT INTO Internacional (Codigo, Linea_aerea, FechaEntrega, Codigo_C)  
VALUES  
( 'PKG005', 'Aerolínea 1', '2024-11-10', 1001),  
( 'PKG006', 'Aerolínea 2', '2024-11-12', 1002),  
( 'PKG007', 'Aerolínea 3', '2024-11-15', 1003),  
( 'PKG008', 'Aerolínea 4', '2024-11-20', 1004);
```

-- Rutas

```
INSERT INTO Rutas (Codigo, Nombre_rutas, RFC)  
VALUES  
( 'PKG001', 'Ruta A', 'RFC001'),  
( 'PKG001', 'Ruta B', 'RFC001'),  
( 'PKG002', 'Ruta C', 'RFC002'),  
( 'PKG002', 'Ruta D', 'RFC002'),  
( 'PKG003', 'Ruta E', 'RFC003'),  
( 'PKG003', 'Ruta F', 'RFC003'),  
( 'PKG004', 'Ruta G', 'RFC004'),  
( 'PKG004', 'Ruta H', 'RFC004');
```

-- AsignacionPaquetes

```
INSERT INTO AsignacionPaquetes (FechaAsignacion, Codigo, Placa)
```

```
VALUES
('2024-11-05', 'PKG009', 'MNO345'),
('2024-11-06', 'PKG010', 'PQR678'),
('2024-11-07', 'PKG011', 'STU901'),
('2024-11-08', 'PKG012', 'VWX234');
```

```
---Ver el contenido de todas las tablas
SELECT * FROM C_Local;
SELECT * FROM Camion;
SELECT * FROM Conductor;
SELECT * FROM Conduce;
SELECT * FROM Paquete;
SELECT * FROM Internacional;
SELECT * FROM Nacional;
SELECT * FROM Rutas;
SELECT * FROM ConteoEnvios;
SELECT * FROM AsignacionPaquetes;
```

```
CREATE OR ALTER TRIGGER TRG_Paquete_Exclusividad_Nacional
ON Nacional
INSTEAD OF INSERT
AS
BEGIN
    -- Verificar si algún código insertado existe en Internacional
    IF EXISTS (
        SELECT 1
        FROM INSERTED I
        WHERE EXISTS (
            SELECT 1
            FROM Internacional Int
            WHERE Int.Codigo = I.Codigo
        )
    )
    BEGIN
        -- Lanzar error si el paquete ya existe en Internacional
        THROW 50000, 'El paquete ya está registrado como Internacional y no puede
ser Nacional al mismo tiempo.', 1;
    END;

    -- Insertar los datos en la tabla Nacional si no hay conflicto
    INSERT INTO Nacional (Codigo, CiudadDestino, RFC)
    SELECT Codigo, CiudadDestino, RFC
```

```
FROM INSERTED;  
END;
```

```
CREATE OR ALTER TRIGGER TRG_Paquete_Exclusividad_Internacional  
ON Internacional  
INSTEAD OF INSERT  
AS  
BEGIN  
    -- Verificar si algún código insertado existe en Nacional  
    IF EXISTS (  
        SELECT 1  
        FROM INSERTED I  
        WHERE EXISTS (  
            SELECT 1  
            FROM Nacional N  
            WHERE N.Codigo = I.Codigo  
        )  
    )  
    BEGIN  
        -- Lanzar error si el paquete ya existe en Nacional  
        THROW 50000, 'El paquete ya está registrado como Nacional y no puede ser  
Internacional al mismo tiempo.', 1;  
    END;  
  
    -- Insertar los datos en la tabla Internacional si no hay conflicto  
    INSERT INTO Internacional (Codigo, Linea_aerea, FechaEntrega, Codigo_C)  
    SELECT Codigo, Linea_aerea, FechaEntrega, Codigo_C  
    FROM INSERTED;  
END;
```

```
CREATE or ALTER TRIGGER TRG_Validar_CargaCamion  
ON Camion  
INSTEAD OF INSERT  
AS  
BEGIN  
    -- Verificar si algún camión rompe la regla de carga  
    IF EXISTS (  
        SELECT 1  
        FROM INSERTED  
        WHERE Carga_max < 250 OR Carga_max > 1250  
    )
```

```

BEGIN
    -- Lanzar un error personalizado si la regla se rompe
    THROW 50000, 'La carga máxima debe estar entre 250 kg y 1250 kg.', 1;
END;
END;

```

```

CREATE OR ALTER TRIGGER TRG_Actualizar_CconteoEnvios ON Nacional
AFTER INSERT
AS
BEGIN
    -- Actualizar el conteo de envíos si la ciudad ya existe en ConteoEnvios
    UPDATE ConteoEnvios
    SET CantidadEnvios = CantidadEnvios + 1
    WHERE CiudadDestino IN (SELECT CiudadDestino FROM INSERTED);

    -- Insertar nuevas ciudades en ConteoEnvios si no existen
    INSERT INTO ConteoEnvios (CiudadDestino, CantidadEnvios)
    SELECT I.CiudadDestino, 1
    FROM INSERTED I
    WHERE NOT EXISTS (
        SELECT 1
        FROM ConteoEnvios CE
        WHERE CE.CiudadDestino = I.CiudadDestino
    );
END;

```

```

go
CREATE PROCEDURE ObtenerDetallesEnvio
    @CodigoPaquete CHAR(18) -- Parámetro de entrada
AS
BEGIN
    -- Declaración de variables para los resultados
    DECLARE @TipoEnvio VARCHAR(15);
    DECLARE @Direccion VARCHAR(100);
    DECLARE @Peso NUMERIC(10, 2);
    DECLARE @ConductorNombre VARCHAR(100);
    DECLARE @RutaNombre VARCHAR(50);
    DECLARE @PlacaCamion VARCHAR(20);

```

```

DECLARE @FechaConduce DATE;
DECLARE @LineaAerea VARCHAR(100);
DECLARE @CodigoCompania NUMERIC(10, 0);

-- Verificar si es Nacional
IF EXISTS (SELECT 1 FROM Nacional WHERECodigo = @CodigoPaquete)
BEGIN
    SET @TipoEnvio = 'Nacional';

    SELECT
        @Direccion = P.Direccion,
        @Peso = P.Peso,
        @ConductorNombre = C.Nombre,
        @RutaNombre = R.Nombre_rutas,
        @PlacaCamion = CN.Placa,
        @FechaConduce = CN.Fecha
    FROM Paquete P
    JOIN Nacional N ON P.Codigo = N.Codigo
    JOIN Conductor C ON N.RFC = C.RFC
    LEFT JOIN Rutas R ON N.Codigo = R.Codigo
    LEFT JOIN Conduce CN ON C.RFC = CN.RFC
    WHERE P.Codigo = @CodigoPaquete;

    SELECT
        @CodigoPaquete AS ID,
        @TipoEnvio AS TipoEnvio,
        @Direccion AS Direccion,
        @Peso AS Peso,
        @ConductorNombre AS ConductorNombre,
        @RutaNombre AS RutaNombre,
        @PlacaCamion AS PlacaCamion,
        @FechaConduce AS FechaConduce;
END

-- Verificar si es Internacional
ELSE IF EXISTS (SELECT 1 FROM Internacional WHERECodigo =
@CodigoPaquete)
BEGIN
    SET @TipoEnvio = 'Internacional';

    SELECT
        @Direccion = P.Direccion,
        @LineaAerea = I.Linea_aerea,
        @CodigoCompania = I.Codigo_C
    FROM Paquete P

```



```

JOIN Internacional I ON P.Codigo = I.Codigo
WHERE P.Codigo = @CodigoPaquete;

SELECT
    @CodigoPaquete AS ID,
    @TipoEnvio AS TipoEnvio,
    @Direccion AS Direccion,
    @LineaAerea AS LineaAerea,
    @CodigoCompania AS CodigoCompania;
END
-- Si no existe
ELSE
BEGIN
    THROW 50000, 'El código de paquete no existe.', 1;
END
END;

```

---

```

--Pruebas de Triggers y Procedimiento almacenado
--1r Trigger 'PKG005' ya está en la tabla Internacional
INSERT INTO Nacional (Codigo, CiudadDestino, RFC)
VALUES ('PKG005', 'Ciudad A', 'RFC001');

--2do Trigger 'PKG001' ya está en la tabla Nacional
INSERT INTO Internacional (Codigo, Linea_aerea, FechaEntrega, Codigo_C)
VALUES ('PKG001', 'Aerolínea 5', '2024-12-01', 1001);

--3r trigger carga de camion
INSERT INTO Camion (Placa, Carga_max, Ciudad_Resguardo)
VALUES ('INVALID1', 150, 'Ciudad Inválida');

--4to trigger actualizar conteo de envios nacionales, inserción válida en Nacional con
una ciudad existente
INSERT INTO Nacional (Codigo, CiudadDestino, RFC)
VALUES ('PKG013', 'Ciudad A', 'RFC002');
---Ciudad que no existe
INSERT INTO Nacional (Codigo, CiudadDestino, RFC)
VALUES ('PKG014', 'Ciudad SinNombre', 'RFC003');

--Para verificar que funciona vemos el contenido de la tabla de Conteo
Select * from ConteoEnvios;
--Procedimiento almacenado
EXEC ObtenerDetallesEnvio @CodigoPaquete = 'PKG001';

```

---

## 5. Diccionario de Datos

### 5.1 Descripción de cada tabla

```
CREATE TABLE AsignacionPaquetes (  
    FechaAsignacion DATETIME NULL,  
   Codigo CHAR(18) NOT NULL, -- FK hacía Paquete  
    Placa VARCHAR(20) NOT NULL, -- FK hacía Camion  
    PRIMARY KEY (Codigo, Placa),  
    CONSTRAINT FK_AsignacionPaquetes_Paquete FOREIGN KEY (Codigo)  
        REFERENCES Paquete (Codigo),  
    CONSTRAINT FK_AsignacionPaquetes_Camion FOREIGN KEY (Placa)  
        REFERENCES Camion (Placa)  
);
```

Esta tabla registra qué paquete ha sido asignado a qué camión en una fecha específica. Tiene llave primaria (Código, Placa), para asegurar que cada combinación de paquete y camión sea única y las llaves foráneas Código referencia a Paquete y Placa referencia a Camión

---

```
CREATE TABLE C_Local (  
    Codigo_C NUMERIC NOT NULL PRIMARY KEY,  
    Nombre VARCHAR(100) NULL  
);
```

Esta tabla representa las compañías locales que manejan envíos internacionales. Tiene la llave primaria Codigo\_C.

---

```

CREATE TABLE Camion (
    Placa VARCHAR(20) NOT NULL PRIMARY KEY,
    Carga_max NUMERIC NULL,
    Ciudad_Resguardo VARCHAR(100) NULL,
    CONSTRAINT CH_Camion_Carga CHECK (Carga_max >= 250 AND
Carga_max <= 1250)
);

```

Esta tabla almacena información de los camiones usados.  
Tiene la llave primaria Placa y la restricción de chequeo donde Carga\_max debe estar entre 250 y 1250

---

```

CREATE TABLE Conduce (
    Fecha DATE NULL,
    RFC VARCHAR(20) NOT NULL, -- FK hacia Conductor
    Placa VARCHAR(20) NOT NULL, -- FK hacia Camion
    PRIMARY KEY (RFC, Placa),
    CONSTRAINT FK_Conduce_Conductor FOREIGN KEY (RFC)
    REFERENCES Conductor (RFC),
    CONSTRAINT FK_Conduce_Camion FOREIGN KEY (Placa)
    REFERENCES Camion (Placa)
);

```

Esta tabla conecta a los conductores con los camiones.  
Tiene la llave primaria (RFC, Placa), para que no haya duplicados en conductor y camión y las llaves foráneas RFC referencia a Conductor y Placa referencia a Camión.

---

```

CREATE TABLE Conductor (
    RFC VARCHAR(20) NOT NULL PRIMARY KEY,
    Nombre VARCHAR(100) NULL,
    Direccion VARCHAR(100) NULL
);

```

Esta tabla almacena información de los conductores.  
Tiene llave primaria RFC

---

```
CREATE TABLE ConteoEnvios (  
    CiudadDestino VARCHAR(100) NOT NULL PRIMARY KEY,  
    CantidadEnvios INT DEFAULT 0  
);
```

Esta tabla sirve para registrar la cantidad de envíos nacionales que se realizan hacia la ciudad de destino.  
Tiene la llave primaria CiudadDestino}

---

```
CREATE TABLE Internacional (  
   Codigo CHAR(18) NOT NULL PRIMARY KEY, -- FK hacia Paquete  
    Linea_aerea VARCHAR(100) NULL,  
    FechaEntrega DATETIME NULL,  
    Codigo_C NUMERIC NOT NULL, -- FK hacia C_Local  
    CONSTRAINT FK_Internacional_C_Local FOREIGN KEY (Codigo_C)  
        REFERENCES C_Local (Codigo_C),  
    CONSTRAINT FK_Internacional_Paquete FOREIGN KEY (Codigo)  
        REFERENCES Paquete (Codigo) ON DELETE CASCADE  
);
```

Esta tabla contiene información sobre los envíos internacionales.  
Tiene la llave primaria Codigo y las llaves foraneas Codigo referencia a Paquete y Codigo\_C referencia a C\_Local, el On Delete Cascade es por si un paquete se elimina, el envío internacional también se elimina.

---

```
CREATE TABLE Nacional (  
   Codigo CHAR(18) NOT NULL PRIMARY KEY, -- FK hacia Paquete  
    CiudadDestino VARCHAR(100) NULL,  
    RFC VARCHAR(20) NOT NULL, -- FK hacia Conductor  
    CONSTRAINT FK_Nacional_Conductor FOREIGN KEY (RFC)  
        REFERENCES Conductor (RFC),
```

```

        CONSTRAINT FK_Nacional_Paquete FOREIGN KEY (Codigo)
        REFERENCES Paquete (Codigo) ON DELETE CASCADE
    );

```

Esta tabla almacena información de los envíos nacionales.

Tiene la llave primaria: Código y Llaves foráneas Código referencia a Paquete y RFC referencia a Conductor, el On Delete Cascade es por si un paquete se elimina, el envío nacional también se elimina.

---

```

CREATE TABLE Paquete (
    Codigo CHAR(18) NOT NULL PRIMARY KEY,
    Direccion VARCHAR(100) NULL,
    Peso NUMERIC NULL,
    Destinatario VARCHAR(100) NULL
);

```

Esta tabla contiene información de los paquetes enviados.

Tiene la llave primaria Codigo

---

```

CREATE TABLE Rutas (
    Codigo CHAR(18) NOT NULL, -- FK hacia Nacional
    Nombre_rutas VARCHAR(50) NOT NULL, -- Ruta específica
    RFC VARCHAR(20) NOT NULL, -- FK hacia Conductor
    CONSTRAINT Rutas_PK PRIMARY KEY (Codigo, Nombre_rutas, RFC),
    CONSTRAINT FK_Rutas_Nacional FOREIGN KEY (Codigo)
    REFERENCES Nacional (Codigo),
    CONSTRAINT FK_Rutas_Conductor FOREIGN KEY (RFC) REFERENCES
    Conductor (RFC),
    CONSTRAINT CH_NombreRutas CHECK (Nombre_rutas IN ('Ruta A',
    'Ruta B', 'Ruta C', 'Ruta D', 'Ruta E', 'Ruta F', 'Ruta G', 'Ruta H', 'Ruta I', 'Ruta
    J'))
);

```

Esta tabla asocia las rutas con conductores y paquetes.

Tiene la llave primaria (Código,Nombre\_rutas, RFC) para evitar duplicados en las rutas y las llaves foráneas son Código referencia a Nacional y RFC

referencia a Conductor y constraint Check es para que nombre\_rutas solo tenga los valores ya definidos.

## 5.2 Descripción por columna de cada tabla

```
CREATE TABLE AsignacionPaquetes (  
    FechaAsignacion DATETIME NULL,  
   Codigo CHAR(18) NOT NULL, -- FK hacía Paquete  
    Placa VARCHAR(20) NOT NULL, -- FK hacía Camion  
    PRIMARY KEY (Codigo, Placa),  
    CONSTRAINT FK_AsignacionPaquetes_Paquete FOREIGN KEY (Codigo)  
        REFERENCES Paquete (Codigo),  
    CONSTRAINT FK_AsignacionPaquetes_Camion FOREIGN KEY (Placa)  
        REFERENCES Camion (Placa)  
);
```

- FechaAsignacion es de tipo DATETIME y es la fecha en la que se asigna un paquete a un camión.
- Código es de tipo CHAR y es el identificador único del paquete
- Placa es de tipo VARCHAR y da la placa del camión al que se asigna el paquete.

```
CREATE TABLE C_Local (  
    Codigo_C NUMERIC NOT NULL PRIMARY KEY,  
    Nombre VARCHAR(100) NULL  
);
```

- Codigo\_C es de tipo NUMERIC y es el codigo de identificacion unico
- Nombre es de tipo VARCHAR y es el nombre de la compañía local

```
CREATE TABLE Camion (  
    Placa VARCHAR(20) NOT NULL PRIMARY KEY,  
    Carga_max NUMERIC NULL,  
    Ciudad_Resguardo VARCHAR(100) NULL,  
    CONSTRAINT CH_Camion_Carga CHECK (Carga_max >= 250 AND  
    Carga_max <= 1250)  
);
```

- Placa es de tipo VARCHAR y es el identificador único del camión

- Carga\_max es de tipo NUMERIC y es la capacidad máxima de carga del camión que debería estar entre 250 y 1250
- Ciudad\_Resguardo es de tipo VARCHAR y es la ciudad donde se guarda el camión.

```
CREATE TABLE Conduce (
    Fecha DATE NULL,
    RFC VARCHAR(20) NOT NULL, -- FK hacia Conductor
    Placa VARCHAR(20) NOT NULL, -- FK hacia Camion
    PRIMARY KEY (RFC, Placa),
    CONSTRAINT FK_Conduce_Conductor FOREIGN KEY (RFC)
        REFERENCES Conductor (RFC),
    CONSTRAINT FK_Conduce_Camion FOREIGN KEY (Placa)
        REFERENCES Camion (Placa)
);
```

- Fecha es de tipo DATE y guarda la fecha en que el conductor maneja el camión.
- RFC es de tipo VARCHAR y es el identificador único del conductor
- Placa es de tipo VARCHAR y es la placa del camión del conductor

```
CREATE TABLE Conductor (
    RFC VARCHAR(20) NOT NULL PRIMARY KEY,
    Nombre VARCHAR(100) NULL,
    Direccion VARCHAR(100) NULL
);
```

- RFC es de tipo VARCHAR y es el identificador único del conductor
- Nombre es de tipo VARCHAR y es el nombre del conductor
- Dirección es de tipo VARCHAR y es la dirección del conductor

```
CREATE TABLE Rutas (
    Codigo CHAR(18),
    Nombre_rutas VARCHAR(50) CHECK (Nombre_rutas IN ('Ruta A', 'Ruta B',
'Ruta C', 'Ruta D')),
    RFC VARCHAR(20),
    PRIMARY KEY (Codigo, Nombre_rutas, RFC),
    FOREIGN KEY (Codigo) REFERENCES Nacional(Codigo),
    FOREIGN KEY (RFC) REFERENCES Conductor(RFC)
);
```

- Es el código único del paquete (relacionado con la tabla Nacional), para identificar qué paquete sigue esta ruta

- Nombre\_rutas: El nombre de la ruta asignada al paquete (por ejemplo, "Ruta A", "Ruta B").
- El RFC del conductor que está a cargo de la ruta.

```
CREATE TABLE AsignacionPaquetes (
    FechaAsignacion DATETIME,
   Codigo CHAR(18),
    Placa VARCHAR(20),
    PRIMARY KEY (Codigo, Placa),
    FOREIGN KEY (Codigo) REFERENCES Paquete(Codigo),
    FOREIGN KEY (Placa) REFERENCES Camion(Placa)
);
```

- FechaAsignacion: La fecha y hora en que se asignó el paquete a un camión.
- Codigo: El código del paquete (relacionado con la tabla Paquete).
- Placa: La placa del camión asignado para transportar el paquete.

```
CREATE TABLE Nacional (
    Codigo CHAR(18),
    CiudadDestino VARCHAR(100),
    RFC VARCHAR(20),
    PRIMARY KEY (Codigo),
    FOREIGN KEY (Codigo) REFERENCES Paquete(Codigo),
    FOREIGN KEY (RFC) REFERENCES Conductor(RFC)
);
```

- Codigo: El código único del paquete.
- CiudadDestino: La ciudad de destino para el paquete.
- RFC: El RFC del conductor responsable del paquete.

```
CREATE TABLE Internacional (
    Codigo CHAR(18),
    Linea_aerea VARCHAR(100),
    FechaEntrega DATETIME,
    Codigo_C NUMERIC,
    PRIMARY KEY (Codigo),
    FOREIGN KEY (Codigo) REFERENCES Paquete(Codigo),
    FOREIGN KEY (Codigo_C) REFERENCES C_Local(Codigo_C)
);
```



- **Codigo:** El código único del paquete.
- **Linea\_aerea:** La aerolínea encargada de transportar el paquete.
- **FechaEntrega:** La fecha en que se entrega el paquete.
- **Codigo\_C:** El código de la compañía encargada de la entrega en el país destino, relacionado con la tabla C\_Local.

```
CREATE TABLE ConteoEnvios (
    CiudadDestino VARCHAR(100),
    CantidadEnvios INT,
    PRIMARY KEY (CiudadDestino)
);
```

- **CiudadDestino:** La ciudad a la que se destinan los paquetes.
- **CantidadEnvios:** La cantidad total de envíos realizados a esa ciudad.

## 5.3 Descripción de Disparadores

### 5.3.1 Disparador para envíos nacionales:

```
CREATE OR ALTER TRIGGER TRG_Paquete_Exclusividad_Nacional
ON Nacional
INSTEAD OF INSERT
AS
BEGIN
    -- Verificar si algún código insertado existe en Internacional
    IF EXISTS (
        SELECT 1
        FROM INSERTED I
        WHERE EXISTS (
```

```

        SELECT 1
        FROM Internacional Int
        WHERE Int.Codigo = I.Codigo
    )
)
BEGIN
    -- Lanzar error si el paquete ya existe en Internacional
    THROW 50000, 'El paquete ya está registrado como Internacional y no puede
ser Nacional al mismo tiempo.', 1;
END;
-- Insertar los datos en la tabla Nacional si no hay conflicto
INSERT INTO Nacional (Codigo, CiudadDestino, RFC)
SELECT Codigo, CiudadDestino, RFC
FROM INSERTED;
END;

```

El trigger TRG\_Paquete\_Exclusividad\_Nacional INSTEAD OF INSERT, esta diseñado para la tabla Nacional donde el propósito es controlar que los códigos de los registros insertados en Nacional no existan ya en la tabla Internacional, garantizando una exclusividad de los códigos entre las tablas.

### 5.3.2 Disparador para envíos internacionales:

```

CREATE OR ALTER TRIGGER TRG_Paquete_Exclusividad_Internacional
ON Internacional
INSTEAD OF INSERT
AS
BEGIN
    -- Verificar si algún código insertado existe en Nacional
    IF EXISTS (
        SELECT 1

```

```

FROM INSERTED I
WHERE EXISTS (
    SELECT 1
    FROM Nacional N
    WHERE N.Codigo = I.Codigo
)
)
BEGIN
    -- Lanzar error si el paquete ya existe en Nacional
    THROW 50000, 'El paquete ya está registrado como Nacional y no puede ser
Internacional al mismo tiempo.', 1;
END;

```

```

-- Insertar los datos en la tabla Internacional si no hay conflicto
INSERT INTO Internacional (Codigo, Linea_aerea, FechaEntrega, Codigo_C)
SELECT Codigo, Linea_aerea, FechaEntrega, Codigo_C
FROM INSERTED;
END;

```

El trigger TRG\_Paquete\_Exclusividad\_Internacional INSTEAD OF INSERT está asociado a la tabla Internacional donde este trigger garantiza la exclusividad entre las tablas Internacional y Nacional, evitando que un código pueda existir en las 2 tablas a la vez.

#### Disparador para validacionCargaCamion :

```

CREATE or ALTER TRIGGER TRG_Validar_CargaCamion
ON Camion
INSTEAD OF INSERT
AS
BEGIN
    -- Verificar si algún camión rompe la regla de carga
    IF EXISTS (
        SELECT 1
        FROM INSERTED
        WHERE Carga_max < 250 OR Carga_max > 1250
    )

```

```

BEGIN
    -- Lanzar un error personalizado si la regla se rompe
    THROW 50000, 'La carga máxima debe estar entre 250 kg y 1250 kg.', 1;
END;
END;

El trigger TRG_Validar_CargaCamion INSTEAD OF INSERT está asociado a la tabla
Camión donde su objetivo es garantizar que los registros insertados cumplan con
una regla de validación relacionada con la capacidad de carga máxima del camión.

```

### 5.3.3 Disparador para eliminación de envíos nacionales:

```

CREATE OR ALTER TRIGGER TRG_Actualizar_CconteoEnvios ON Nacional
AFTER INSERT
AS
BEGIN
    -- Actualizar el conteo de envíos si la ciudad ya existe en ConteoEnvios
    UPDATE ConteoEnvios
    SET CantidadEnvios = CantidadEnvios + 1
    WHERE CiudadDestino IN (SELECT CiudadDestino FROM INSERTED);

    -- Insertar nuevas ciudades en ConteoEnvios si no existen
    INSERT INTO ConteoEnvios (CiudadDestino, CantidadEnvios)

```

```

SELECT I.CiudadDestino, 1
FROM INSERTED I
WHERE NOT EXISTS (
    SELECT 1
    FROM ConteoEnvios CE
    WHERE CE.CiudadDestino = I.CiudadDestino
);
END;

```

El trigger TRG\_Validar\_CargaCamion INSTEAD OF INSERT valida los datos de la tabla Camión antes de que se haga una inserción. Su objetivo es garantizar que el valor de la columna Carga\_max cumpla con el rango de 250 kg a 1250 kg. Si no se cumple, el trigger bloquea la operación y lanza un error.

## 5.4 Descripción de Procedimientos Almacenados

```

CREATE PROCEDURE ObtenerDetallesEnvio

```

```

    @CodigoPaquete CHAR(18) -- Parámetro de entrada

```

```

AS

```

```

BEGIN

```

```

    -- Declaración de variables para los resultados

```

```

    DECLARE @TipoEnvio VARCHAR(15);

```

```

    DECLARE @Direccion VARCHAR(100);

```

```

DECLARE @Peso NUMERIC(10, 2);

DECLARE @ConductorNombre VARCHAR(100);

DECLARE @RutaNombre VARCHAR(50);

DECLARE @PlacaCamion VARCHAR(20);

DECLARE @FechaConduce DATE;

DECLARE @LineaAerea VARCHAR(100);

DECLARE @CodigoCompania NUMERIC(10, 0);

-- Verificar si es Nacional

IF EXISTS (SELECT 1 FROM Nacional WHERE Codigo = @CodigoPaquete)

BEGIN

    SET @TipoEnvio = 'Nacional';

    SELECT

        @Direccion = P.Direccion,

        @Peso = P.Peso,

        @ConductorNombre = C.Nombre,

        @RutaNombre = R.Nombre_rutas,

        @PlacaCamion = CN.Placa,

        @FechaConduce = CN.Fecha

    FROM Paquete P

```

JOIN Nacional N ON P.Codigo = N.Codigo

JOIN Conductor C ON N.RFC = C.RFC

LEFT JOIN Rutas R ON N.Codigo = R.Codigo

LEFT JOIN Conduce CN ON C.RFC = CN.RFC

WHERE P.Codigo = @CodigoPaquete;

SELECT

@CodigoPaquete AS ID,

@TipoEnvio AS TipoEnvio,

@Direccion AS Direccion,

@Peso AS Peso,

@ConductorNombre AS ConductorNombre,

@RutaNombre AS RutaNombre,

@PlacaCamion AS PlacaCamion,

@FechaConduce AS FechaConduce;

END

-- Verificar si es Internacional

ELSE IF EXISTS (SELECT 1 FROM Internacional WHERE Codigo =  
@CodigoPaquete)

BEGIN

```

SET @TipoEnvio = 'Internacional';

SELECT

    @Direccion = P.Direccion,

    @LineaAerea = I.Linea_aerea,

    @CodigoCompania = I.Codigo_C

FROM Paquete P

JOIN Internacional I ON P.Codigo = I.Codigo

WHERE P.Codigo = @CodigoPaquete;

SELECT

    @CodigoPaquete AS ID,

    @TipoEnvio AS TipoEnvio,

    @Direccion AS Direccion,

    @LineaAerea AS LineaAerea,

    @CodigoCompania AS CodigoCompania;

END

-- Si no existe

ELSE

BEGIN

    THROW 50000, 'El código de paquete no existe.', 1;

```



END

END;

Este procedimiento está diseñado para consultar información detallada de un paquete, dependiendo de si el envío es nacional o internacional.

El procedimiento ObtenerDetallesEnvio empieza recibiendo un parámetro llamado @CodigoPaquete, que es un identificador único del paquete.

Luego, declaramos varias variables para ir guardando la información que se va obteniendo como por ejemplo en @TipoEnvio sirve para saber si el paquete es nacional o internacional, y @Direccion guarda la dirección asociada. Si el envío es nacional, también usó variables como @Peso para el peso, @ConductorNombre para el nombre del conductor, @RutaNombre para la ruta, @PlacaCamion para la placa del camión, y @FechaConduce para la fecha en que el conductor usó ese camión.

Por otro lado, si el envío es internacional, tengo @LineaAerea para la línea aérea y @CodigoCompania para el código de la compañía. Estas variables me ayudan a organizar y devolver los datos dependiendo del tipo de envío.