

# PY518: Implementación del modelo físico

*Diseño funcional y Carga de Datos  
(DDL/DML)*

*Autores: Alexander, Sergio, Carlos*



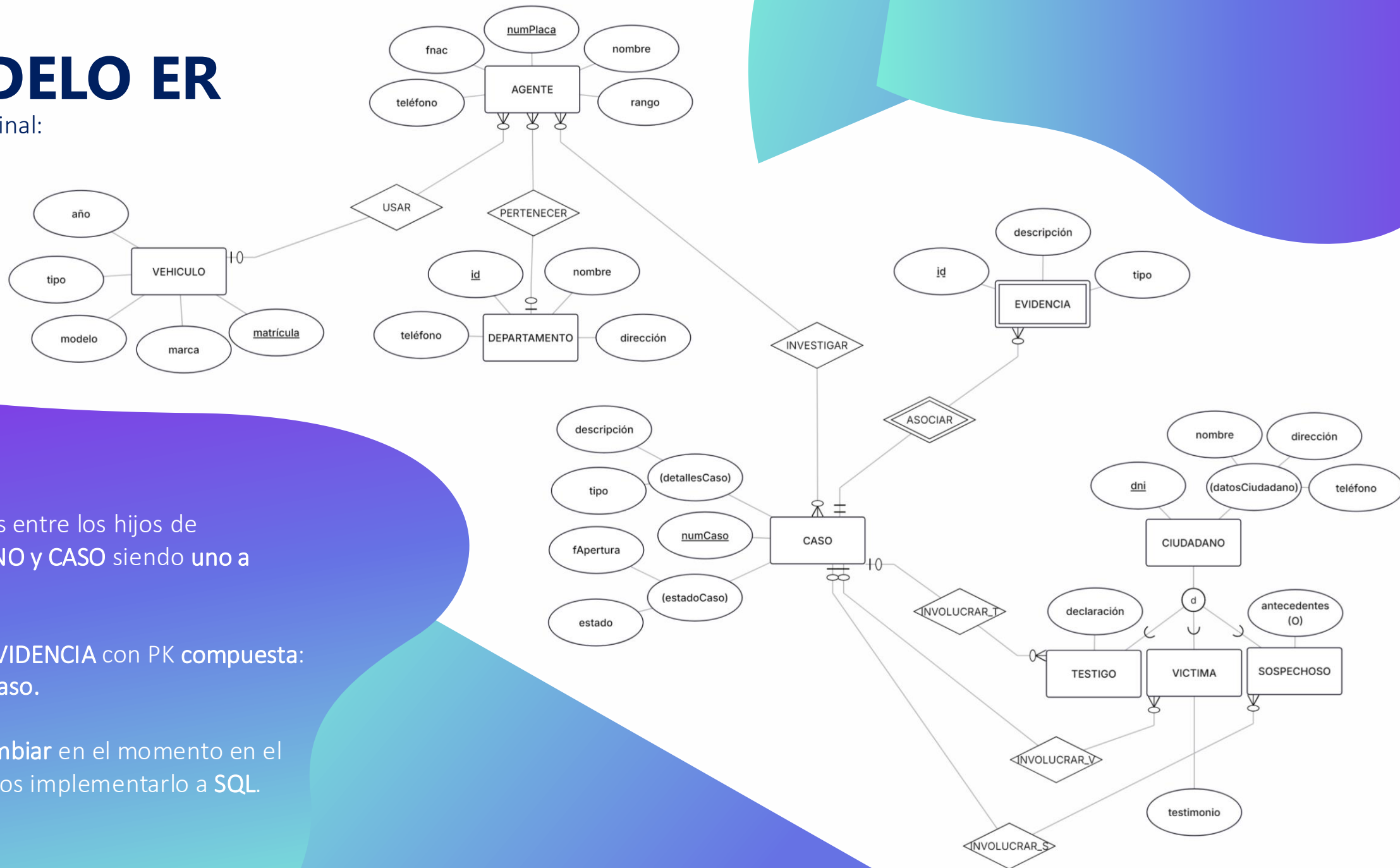
# ÍNDICE

- ☐ *MER: Diseño conceptual*
- ☐ *MR: Modelo Relacional y Adaptaciones*
- ☐ *DDL: Implementación, Claves y Restricciones*
- ☐ *DML: Carga de Datos y Validación*
- ☐ *TRUNCATE: Borrado de Datos*



# MODELO ER

Diseño original:



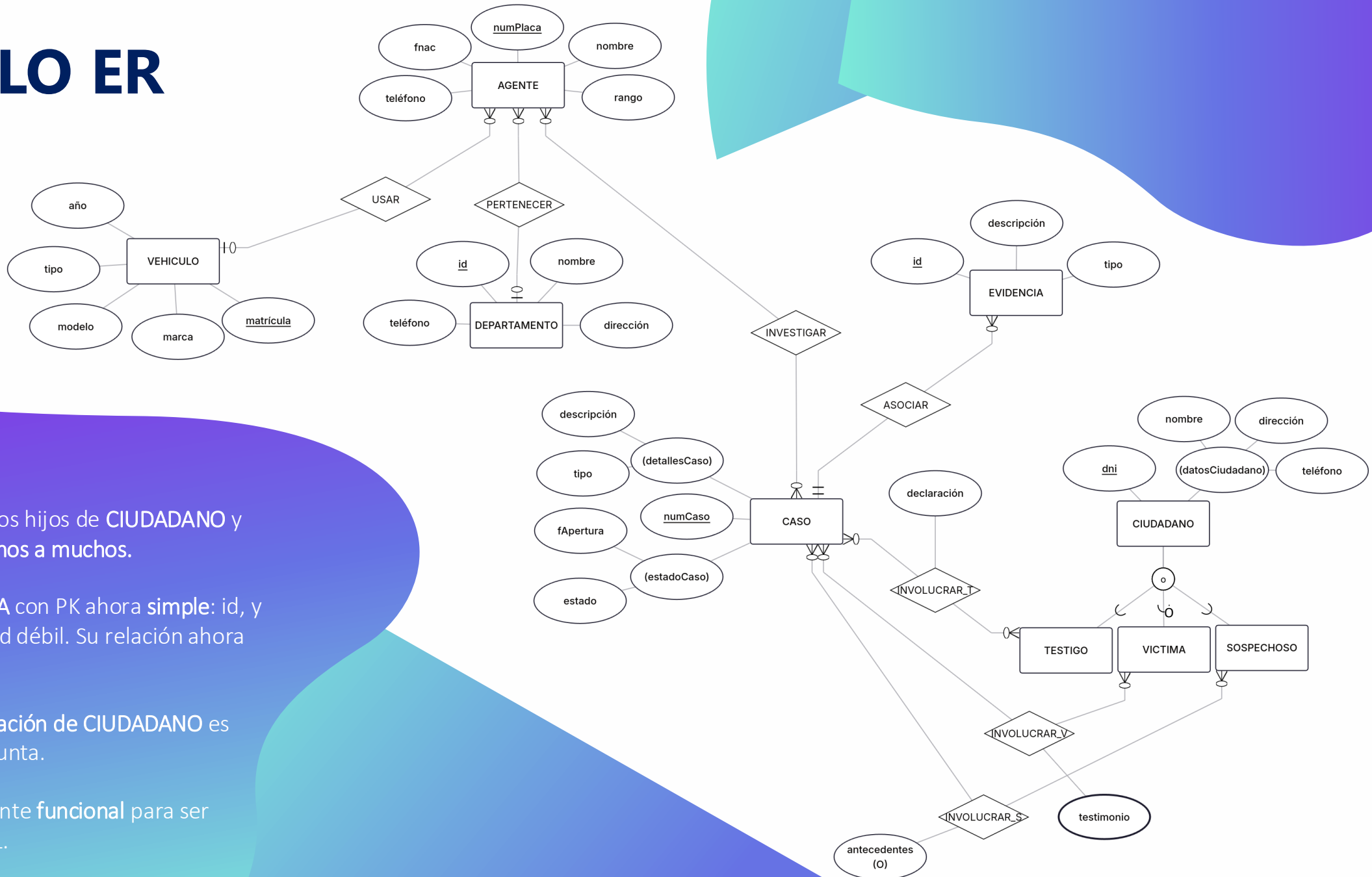
## Puntos clave:

- Relaciones entre los hijos de **CIUDADANO** y **CASO** siendo uno a muchos.
- Entidad **EVIDENCIA** con PK compuesta: id y numCaso.

Esto debe **cambiar** en el momento en el que intentamos implementarlo a **SQL**.

# MODELO ER

Diseño nuevo:



## Puntos clave:

- Relaciones entre los hijos de CIUDADANO y CASO siendo muchos a muchos.
- Entidad EVIDENCIA con PK ahora simple: id, y deja de ser entidad débil. Su relación ahora permite VNN.
- Ahora la generalización de CIUDADANO es solapada y no disjunta.

Ahora es completamente funcional para ser implementado en SQL.

# MODELO MR

Diseño original:

Puntos clave:

- FK de TESTIGO, VICTIMA, SOSPECHOSO siendo numCaso.

La relación no les permite pertenecer a más de un caso (1:N).

TESTIGO (dni\*, numCaso\*, declaración)

- PK (dni)
- FK (dni) → CIUDADANO
- FK (numCaso) → CASO

VICTIMA (dni\*, numCaso\*, testimonio)

- PK (dni)
- FK (dni) → CIUDADANO
- FK (numCaso) → CASO

SOSPECHOSO (dni\*, numCaso\*, antecedentes)

- PK (dni)
- FK (dni) → CIUDADANO
- FK (numCaso) → CASO

EVIDENCIA (id, numCaso\*, descripción, tipo)

- PK (id, numCaso)
- FK (numCaso) → CASO

Puntos clave:

- PK compuesta, con id y numCaso.

Correcto para una entidad débil en el modelo MR, pero no válido para SQL (id tiene AUTO\_INCREMENT).

# MODELO MR

Diseño nuevo:

Puntos clave:

- FK de TESTIGO, VICTIMA y SOSPECHOSO pasan a otras 3 tablas intermedias.

Correcto para SQL, ahora es una relación N:M que permite que un CIUDADANO pertenezca a más de un caso.

TESTIGO (dni\*)

- PK (dni)
- FK (dni) → CIUDADANO (dni)

INVOLUCRAR\_T (dniTestigo\*, numCaso\*, declaración)

- PK (dniTestigo, numCaso)
- FK (dniTestigo) → TESTIGO (dni)
- FK (numCaso) → CASO (numCaso)

VICTIMA (dni\*)

- PK (dni)
- FK (dni) → CIUDADANO (dni)

INVOLUCRAR\_V (dniVictima\*, numCaso\*, testimonio)

- PK (dniVictima, numCaso)
- FK (dniVictima) → VICTIMA (dni)
- FK (numCaso) → CASO (numCaso)

SOSPECHOSO (dni\*)

- PK (dni)
- FK (dni) → CIUDADANO (dni)

INVOLUCRAR\_S (dniSospechoso\*, numCaso\*, antecedentes)

- PK (dniSospechoso, numCaso)
- FK (dniSospechoso) → SOSPECHOSO (dni)
- FK (numCaso) → CASO (numCaso)

EVIDENCIA (id, numCaso\*, descripción, tipo)

- PK (id)
- FK (numCaso) → CASO (numCaso)
- VNN (numCaso)

Puntos clave:

- PK simple, con id.

Deja de ser entidad débil y ahora permite que id tenga AUTO\_INCREMENT.

# SCRIPT DDL

SQL más destacable:

```
CREATE TABLE CIUDADANO (  
    dni VARCHAR(9),  
    descripcionCaso VARCHAR(1024),  
    tipoCaso VARCHAR(64),  
    fApertura DATE DEFAULT CURRENT_DATE,  
    estadoCaso VARCHAR(32),  
  
    CONSTRAINT PK_CASO PRIMARY KEY (numCaso)  
);  
  
CREATE TABLE TESTIGO (  
    dni VARCHAR(9),  
    CONSTRAINT FK_TES_CIU FOREIGN KEY (dni) REFERENCES CIUDADANO(dni) ON DELETE CASCADE  
);  
  
CREATE TABLE INVOLUCRAR_T (  
    dniTestigo VARCHAR(9),  
    numCaso INT UNSIGNED,  
    declaracion VARCHAR(512),  
  
    CONSTRAINT PK_INVT PRIMARY KEY (dniTestigo, numCaso),  
    CONSTRAINT FK_INVT_TES FOREIGN KEY (dniTestigo) REFERENCES TESTIGO(dni) ON DELETE CASCADE,  
    CONSTRAINT FK_INVT_CAS FOREIGN KEY (numCaso) REFERENCES CASO (numCaso) ON DELETE CASCADE  
);
```

Vemos el uso de **ON DELETE CASCADE**; si se borra un **CIUDADANO** que era **TESTIGO**, se borra el registro de **ambos**.

Por otro lado, otro ejemplo es si se borra un **CASO**, todas las **declaraciones** de **TESTIGO** relacionadas con ese caso se **borran** también.





# SCRIPT DDL

SQL más destacable:

```
CREATE TABLE EVIDENCIA (  
    id INT UNSIGNED AUTO_INCREMENT,  
    numCaso INT UNSIGNED NOT NULL,  
    descripcion VARCHAR(512),  
    tipo VARCHAR(64),  
  
    CONSTRAINT PK_EVIDENCIA PRIMARY KEY (id),  
    CONSTRAINT FK_EVI_CAS FOREIGN KEY (numCaso) REFERENCES CASO(numCaso) ON DELETE CASCADE  
);  
  
CREATE TABLE DEPARTAMENTO (  
    id INT UNSIGNED AUTO_INCREMENT,  
    nombre VARCHAR(128),  
    telefono VARCHAR(16),  
    direccion VARCHAR(128),  
  
    CONSTRAINT PK_DEPARTAMENTO PRIMARY KEY (id)  
);  
  
ALTER TABLE DEPARTAMENTO AUTO_INCREMENT = 100;
```

Si un **CASO** se borra, todas sus **EVIDENCIAS** se borran automáticamente.

El id de **DEPARTAMENTO** comienza desde **100** y va incrementando automáticamente.



# SCRIPT DML

```
INSERT INTO VEHICULO (matricula, marca, modelo, tipo, anyo) VALUES
('ABC1234', 'Toyota', 'Corolla', 'Sedán', 2019);

INSERT INTO DEPARTAMENTO (nombre, telefono, direccion) VALUES
('Investigación Criminal', '+34 911 223 344', 'Calle de la Justicia, 1');

INSERT INTO CIUDADANO (dni, nombre, direccion, telefono) VALUES
('X4567890R', 'Alicia Pérez', 'Plaza Mayor 5, Madrid', '+34 600 111 222');

INSERT INTO CASO (descripcionCaso, tipoCaso, fApertura, estadoCaso) VALUES
('Robo a mano armada en joyería del centro.', 'Robo Agravado', '2025-10-20', 'Abierto');

INSERT INTO AGENTE (numPlaca, nombre, fnac, telefono, rango, matriculaVehiculo, idDepartamento) VALUES
('P001', 'David Sanz', '1980-05-15', '+34 611 999 000', 'Sargento', 'ABC1234', 100);

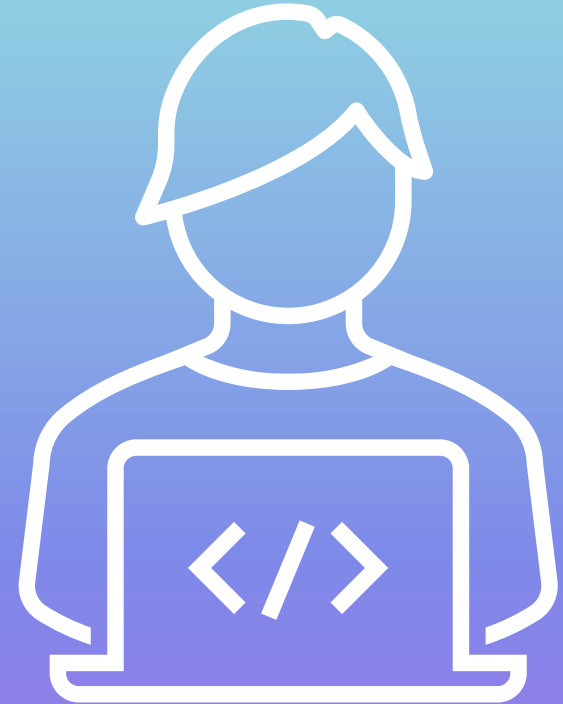
INSERT INTO INVESTIGAR (numPlaca, numCaso) VALUES
('P001', 1);

INSERT INTO TESTIGO (dni) VALUES
('X4567890R');

INSERT INTO INVOLUCRAR_T (dniTestigo, numCaso, declaracion) VALUES
('X4567890R', 1, 'Vi a dos hombres huir en una moto tras el robo.');
```

```
INSERT INTO EVIDENCIA (numCaso, descripcion, tipo) VALUES
(1, 'Casquillo de bala calibre 9mm encontrado en la escena.', 'Forense');
```

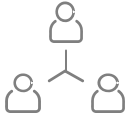
Las tablas **CASO**, **DEPARTAMENTO** y **EVIDENCIA** generan automáticamente sus **PK** al insertar registros (**AUTO\_INCREMENT**).



# VACIAR TABLAS

```
-- 1. Deshabilitar temporalmente la verificación de claves foráneas.  
-- Esto permite truncar tablas con relaciones sin error.  
SET FOREIGN_KEY_CHECKS = 0;  
  
-- 2. TRUNCAR LAS TABLAS  
-- TRUNCATE TABLE es la forma más eficiente de eliminar todos los datos  
  
-- Tablas de relaciones y entidades débiles (se pueden truncar primero)  
TRUNCATE TABLE INVESTIGAR;  
TRUNCATE TABLE INVOLUCRAR_T;  
TRUNCATE TABLE TESTIGO;  
TRUNCATE TABLE INVOLUCRAR_V;  
TRUNCATE TABLE VICTIMA;  
TRUNCATE TABLE INVOLUCRAR_S;  
TRUNCATE TABLE SOSPECHOSO;  
TRUNCATE TABLE EVIDENCIA;  
  
-- Tablas de entidades principales (en un orden seguro)  
TRUNCATE TABLE AGENTE;  
TRUNCATE TABLE CASO;  
TRUNCATE TABLE DEPARTAMENTO;  
TRUNCATE TABLE VEHICULO;  
TRUNCATE TABLE CIUDADANO;  
  
-- 3. Habilitar de nuevo la verificación de claves foráneas.  
  
SET FOREIGN_KEY_CHECKS = 1;
```





# Gracias