



UNIVERSIDAD DE BUENOS AIRES

Departamento de Computación

Facultad de Ciencias Exactas y Naturales

## Base de Datos

### Trabajo Práctico

23 de septiembre de 2015

| Integrante                 | LU     | Correo electrónico          |
|----------------------------|--------|-----------------------------|
| Maurizio, Miguel Sebastián | 635/11 | miguelmaurizio.92@gmail.com |
| Prillo, Sebastián          | 616/11 | sebastianprillo@gmail.com   |
| Tagliavini Ponce, Guido    | 783/11 | guido.tag@gmail.com         |

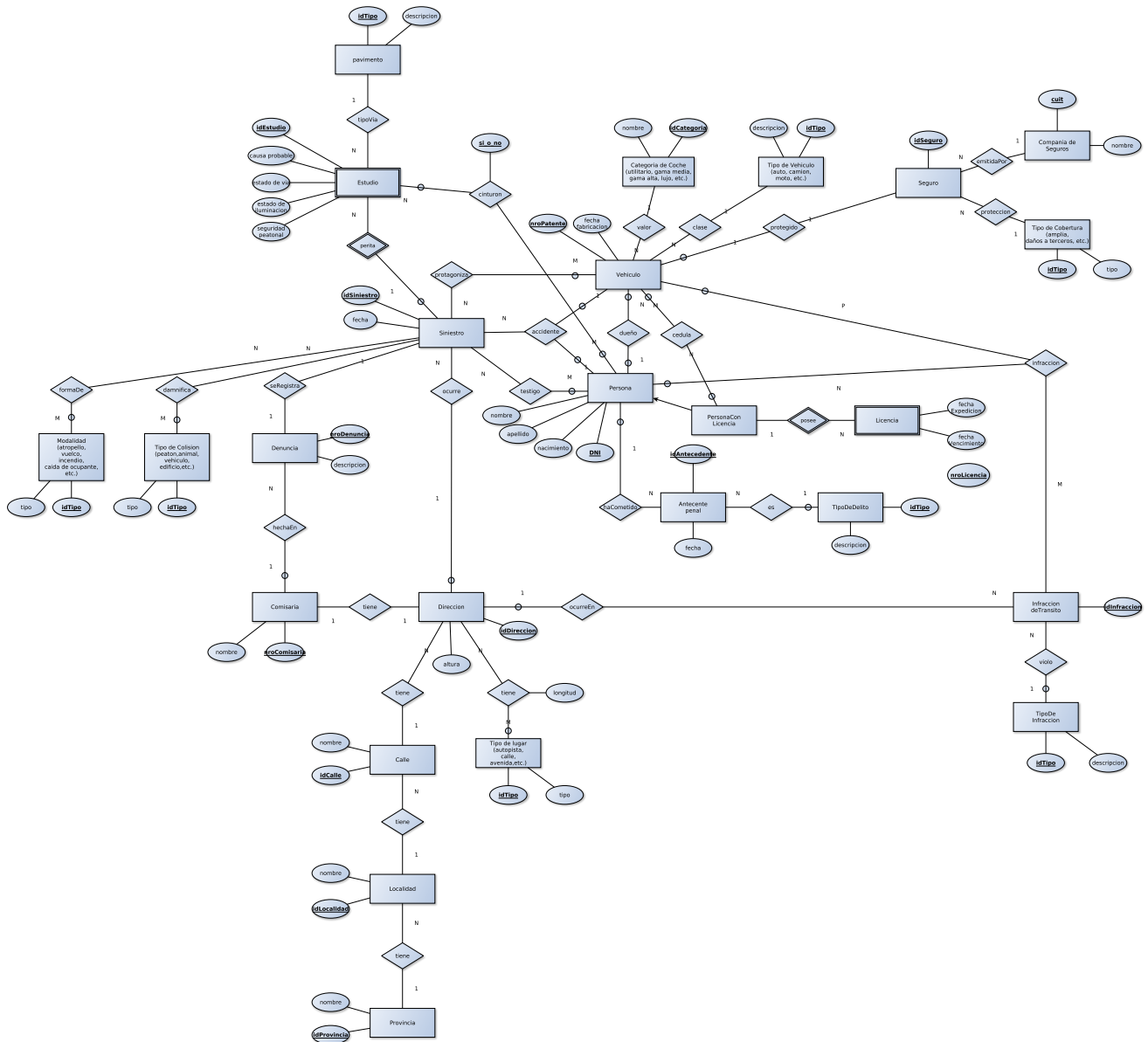
# Índice

|  |          |
|--|----------|
| <b>1. Introducción</b>                           | <b>1</b> |
| <b>2. Diagrama Entidad Relación</b>              | <b>1</b> |
| 2.1. Restricciones en lenguaje natural . . . . . | 1        |
| <b>3. Modelo Relacional</b>                      | <b>2</b> |
| <b>4. Implementación</b>                         | <b>7</b> |

## 1. Introducción

En este trabajo práctico usamos nuestros conocimientos sobre bases de datos para diseñar e implementar una base de datos - RUAT, por *Registro Único de Accidentes de Tránsito* - que unifica información de infracciones de tránsito y siniestros viales ocurridos en el país.

## 2. Diagrama Entidad Relación



## 2.1. Restricciones en lenguaje natural

- Los conductores que pertenecen a un estudio, deben ser conductores involucrados en el siniestro de ese estudio.
- Si un vehiculo aparece relacionado con siniestro sin conductor (en la binaria), entonces no aparece en la ternaria. (No tiene sentido que aparezca con y sin conductor).

- Las personas que aparecen relacionadas con el estudio en la relacion cinturon, deben ser personas que participaron del siniestro asociado al estudio, conduciendo uno de los vehiculos de ese siniestro.

### 3. Modelo Relacional

```
-- Tabla Provincia
CREATE TABLE provincia (
  idProvincia INTEGER NOT NULL,
  nombre VARCHAR(255) DEFAULT NULL,
  PRIMARY KEY (idProvincia)
);

-- Tabla Localidad
CREATE TABLE localidad (
  idLocalidad INTEGER NOT NULL PRIMARY KEY,
  nombre VARCHAR(255) DEFAULT NULL,
  tiene INTEGER NOT NULL,
  FOREIGN KEY(tiene) REFERENCES provincia(idProvincia)
);

-- Tabla Calle
CREATE TABLE calle (
  idCalle INTEGER NOT NULL PRIMARY KEY,
  nombre VARCHAR(255) DEFAULT NULL,
  tiene INTEGER NOT NULL,
  FOREIGN KEY(tiene) REFERENCES localidad(idLocalidad)
);

-- Tabla Direccion
CREATE TABLE direccion (
  idDireccion INTEGER NOT NULL PRIMARY KEY,
  altura INTEGER NOT NULL,
  tiene INTEGER NOT NULL,
  FOREIGN KEY(tiene) REFERENCES calle(idCalle)
);

-- Tabla Tipo de Lugar
CREATE TABLE tipo_lugar (
  idTipo INTEGER NOT NULL PRIMARY KEY,
  tipo VARCHAR(255) DEFAULT NULL
);

-- Tabla Tiene
CREATE TABLE tiene (
  idDireccion INTEGER NOT NULL,
  idTipo INTEGER NOT NULL,
  longitud INTEGER NOT NULL,
  PRIMARY KEY (idDireccion, idTipo),
  FOREIGN KEY(idDireccion) REFERENCES direccion(idDireccion),
  FOREIGN KEY(idTipo) REFERENCES tipo_lugar(idTipo)
```

```

);

-- Tabla Comisaria
CREATE TABLE comisaria (
    nroComisaria INTEGER NOT NULL PRIMARY KEY,
    nombre VARCHAR(255) DEFAULT NULL,
    tiene INTEGER NOT NULL,
    FOREIGN KEY(tiene) REFERENCES direccion(idDireccion)
);

-- Tabla Denuncia
CREATE TABLE denuncia (
    nroDenuncia INTEGER NOT NULL PRIMARY KEY,
    descripcion VARCHAR(255) DEFAULT NULL,
    hechaEn INTEGER NOT NULL,
    FOREIGN KEY(hechaEn) REFERENCES comisaria(nroComisaria)
);

-- Tabla Tipo Colision
CREATE TABLE tipo_colision (
    idTipo INTEGER NOT NULL PRIMARY KEY,
    tipo VARCHAR(255) DEFAULT NULL
);

-- Tabla Modalidad
CREATE TABLE modalidad (
    idTipo INTEGER NOT NULL PRIMARY KEY,
    tipo VARCHAR(255) DEFAULT NULL
);

-- Tabla Siniestro
CREATE TABLE siniestro (
    idSiniestro INTEGER NOT NULL PRIMARY KEY,
    fecha DATETIME DEFAULT NULL,
    seRegistra INTEGER NOT NULL,
    ocurre INTEGER NOT NULL,
    FOREIGN KEY(seRegistra) REFERENCES denuncia(nroDenuncia),
    FOREIGN KEY(ocurre) REFERENCES direccion(idDireccion)
);

-- Tabla Damnifica
CREATE TABLE damnifica (
    idSiniestro INTEGER NOT NULL,
    idTipo INTEGER NOT NULL,
    FOREIGN KEY(idSiniestro) REFERENCES siniestro(idSiniestro),
    FOREIGN KEY(idTipo) REFERENCES tipo_colision(idTipo),
    PRIMARY KEY (idSiniestro, idTipo)
);

-- Tabla Forma De
CREATE TABLE forma_de (

```

```

idSiniestro INTEGER NOT NULL,
idTipo INTEGER NOT NULL,
FOREIGN KEY(idSiniestro) REFERENCES siniestro(idSiniestro),
FOREIGN KEY(idTipo) REFERENCES modalidad(idTipo),
PRIMARY KEY (idSiniestro, idTipo)
);

-- Tabla Pavimento
CREATE TABLE pavimento (
    idPavimento INTEGER NOT NULL PRIMARY KEY,
    descripcion VARCHAR(255) DEFAULT NULL
);

-- Tabla Estudio
CREATE TABLE estudio (
    idEstudio INTEGER NOT NULL,
    causaProbable VARCHAR(255) DEFAULT NULL,
    estadoVia VARCHAR(255) DEFAULT NULL,
    estadoIluminacion VARCHAR(255) DEFAULT NULL,
    seguridadPeatonal BOOLEAN DEFAULT NULL,
    tipoVia INTEGER NOT NULL,
    perita INTEGER NOT NULL,
    FOREIGN KEY(tipoVia) REFERENCES modalidad(idTipo),
    FOREIGN KEY(perita) REFERENCES siniestro(idSiniestro) ,
    PRIMARY KEY (idEstudio, perita)
);

-- Tabla Persona
CREATE TABLE persona (
    dni INTEGER NOT NULL PRIMARY KEY,
    nombre VARCHAR(255) NOT NULL,
    apellido VARCHAR(255) NOT NULL,
    nacimiento DATE NOT NULL
);

-- Tabla Testigo
CREATE TABLE testigo (
    idSiniestro INTEGER NOT NULL,
    dni INTEGER NOT NULL,
    FOREIGN KEY(idSiniestro) REFERENCES siniestro(idSiniestro),
    FOREIGN KEY(dni) REFERENCES persona(dni),
    PRIMARY KEY (idSiniestro, dni)
);

-- Tabla Cinturon
CREATE TABLE cinturon (
    idEstudio INTEGER NOT NULL,
    dni INTEGER NOT NULL,
    tiene BOOLEAN NOT NULL,
    FOREIGN KEY(idEstudio) REFERENCES estudio(idEstudio),
    FOREIGN KEY(dni) REFERENCES persona(dni),

```

```

PRIMARY KEY (idEstudio, dni)
);

-- Tabla Tipo Delito
CREATE TABLE tipo_delito (
    idTipo INTEGER NOT NULL PRIMARY KEY,
    descripcion VARCHAR(255) DEFAULT NULL
);

-- Tabla Antecedente penal
CREATE TABLE antecedente_penal (
    idAntecedente INTEGER NOT NULL PRIMARY KEY,
    fecha Date NOT NULL,
    haCometido INTEGER NOT NULL,
    es INTEGER NOT NULL,
    FOREIGN KEY(haCometido) REFERENCES persona(dni),
    FOREIGN KEY(es) REFERENCES tipo_delito(idTipo)
);

-- Tabla Tipo Infraccion
CREATE TABLE tipo_infraccion (
    idTipo INTEGER NOT NULL PRIMARY KEY,
    descripcion VARCHAR(255) DEFAULT NULL
);

-- Tabla Antecedente penal
CREATE TABLE infraccion_transito (
    idInfraccion INTEGER NOT NULL PRIMARY KEY,
    ocurreEn INTEGER NOT NULL,
    violo INTEGER NOT NULL,
    FOREIGN KEY(ocurreEn) REFERENCES direccion(idDireccion),
    FOREIGN KEY(violo) REFERENCES tipo_infraccion(idTipo)
);

-- Tabla Persona con Licencia
CREATE TABLE persona_con_licencia (
    dni INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY(dni) REFERENCES persona(dni)
);

-- Licencia
CREATE TABLE licencia (
    nroLicencia INTEGER NOT NULL,
    dni INTEGER NOT NULL,
    expedicion DATE NOT NULL,
    expiracion DATE NOT NULL,
    FOREIGN KEY(dni) REFERENCES persona_con_licencia(dni),
    PRIMARY KEY (nroLicencia, dni)
);

-- Tabla Compania Seguros

```

```

CREATE TABLE compania_seguro (
    cuit INTEGER NOT NULL PRIMARY KEY,
    nombre VARCHAR(255) NOT NULL
);

-- Tabla Cobertura
CREATE TABLE cobertura (
    idTipo INTEGER NOT NULL PRIMARY KEY,
    descripcion VARCHAR(255) NOT NULL
);

-- Tabla Seguros
CREATE TABLE seguro (
    idSeguro INTEGER NOT NULL PRIMARY KEY,
    emitidaPor INTEGER NOT NULL,
    proteccion INTEGER NOT NULL,
    FOREIGN KEY(emitidaPor) REFERENCES compania_seguro(cuit),
    FOREIGN KEY(proteccion) REFERENCES cobertura(idTipo)
);

-- Tabla Tipo Vehiculo
CREATE TABLE tipo_vehiculo (
    idTipo INTEGER NOT NULL PRIMARY KEY,
    descripcion VARCHAR(255) NOT NULL
);

-- Tabla Categoria Vehiculo
CREATE TABLE categoria_vehiculo (
    idCategoria INTEGER NOT NULL PRIMARY KEY,
    descripcion VARCHAR(255) NOT NULL
);

-- Tabla Vehiculo
CREATE TABLE vehiculo (
    nroPatente CHARACTER(6) NOT NULL PRIMARY KEY,
    fechaFabricacion DATE NOT NULL,
    valor INTEGER NOT NULL,
    clase INTEGER NOT NULL,
    protegido INTEGER NOT NULL,
    dueno INTEGER NOT NULL,
    FOREIGN KEY(valor) REFERENCES categoria_vehiculo(idCategoria),
    FOREIGN KEY(clase) REFERENCES tipo_vehiculo(idTipo),
    FOREIGN KEY(protegido) REFERENCES seguro(idSeguro),
    FOREIGN KEY(dueno) REFERENCES persona(dni)
);

-- Tabla Cedula
CREATE TABLE cedula (
    nroPatente CHARACTER(6) NOT NULL,
    dni INTEGER NOT NULL,
    FOREIGN KEY(nroPatente) REFERENCES vehiculo(nroPatente),

```



```

FOREIGN KEY(dni) REFERENCES persona_con_licencia(dni),
PRIMARY KEY (nroPatente, dni)
);

-- Tabla Protagoniza
CREATE TABLE protagoniza (
  nroPatente CHARACTER(6) NOT NULL,
  idSiniestro INTEGER NOT NULL,
  FOREIGN KEY(nroPatente) REFERENCES vehiculo(nroPatente),
  FOREIGN KEY(idSiniestro) REFERENCES siniestro(idSiniestro),
  PRIMARY KEY (nroPatente, idSiniestro)
);

-- Tabla Accidente
CREATE TABLE accidente (
  nroPatente CHARACTER(6) NOT NULL,
  idSiniestro INTEGER NOT NULL,
  dni INTEGER NOT NULL,
  FOREIGN KEY(nroPatente) REFERENCES vehiculo(nroPatente),
  FOREIGN KEY(idSiniestro) REFERENCES siniestro(idSiniestro),
  FOREIGN KEY(dni) REFERENCES persona(dni),
  PRIMARY KEY (nroPatente, idSiniestro)
);

-- Tabla Infraccion
CREATE TABLE infraccion (
  nroPatente CHARACTER(6) NOT NULL,
  dni INTEGER NOT NULL,
  idInfraccion INTEGER NOT NULL,
  FOREIGN KEY(nroPatente) REFERENCES vehiculo(nroPatente),
  FOREIGN KEY(dni) REFERENCES persona(dni),
  FOREIGN KEY(idInfraccion) REFERENCES infraccion_transito(idInfraccion),
  PRIMARY KEY (nroPatente, dni, idInfraccion)
);

```

## 4. Implementación

Implementamos la base de datos usando la aplicacion SQLite. Las tablas mostradas en la seccion del modelo relacional son precisamente el codigo ddl utilizado para definir las tablas en SQLite.