

Creación de una Base de Datos



Autor: Miguel Benayas Penas, Marzo 2021

Índice:

1. Breve introducción
2. Diseño conceptual (Modelo Entidad-Relación)
3. Diseño lógico (Modelo Relacional)
4. Consultas
5. Conclusiones

1. Breve Introducción

En este proyecto se procederá al diseño e implementación de una base de datos llamada 'España Relimpia' sobre la corrupción en España. En primer lugar, se tratará el diseño conceptual por medio del modelo entidad-relación. En este modelo entidad-relación se presentarán las entidades, sus relaciones y los atributos de ambos conjuntos de forma gráfica.

En segundo lugar, se aborda el diseño lógico a través del modelo relacional, donde se tabula el modelo entidad-relación añadiendo restricciones de integridad para la base de datos. Finalmente se insertarán los datos en la base y se realizarán diferentes consultas reflejando la variedad mostrada en las diapositivas..

2. Diseño conceptual

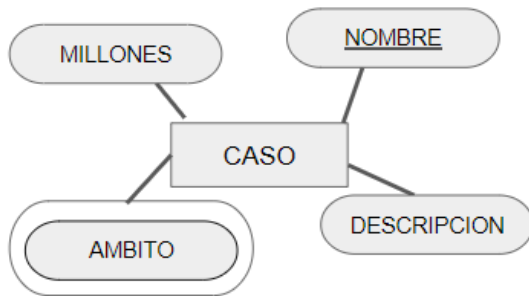
Las entidades y relaciones, junto con sus correspondientes atributos son ilustrados y comentados. En primer lugar se tratan las entidades y sus atributos. Después se presenta el diagrama entidad-relación de la base de datos sin mostrar los atributos de las entidades, ya discutidas en la primera parte de este apartado.

2.1 Entidades

2.1.1. Entidad **caso**

La entidad **caso** presenta cuatro atributos. De ellos, **nombre** es la clave primaria ya que solo se necesita esa información para determinar de forma única cualquier instancia **caso**. La designación de las clave primarias de las tablas se tratará más adelante en el diseño lógico.

El atributo **ambito** es simple y multivalorado ya que, como reza el enunciado, 'un caso puede tener varios ámbitos'. Los demás atributos son simples univalorados.



A continuación se expone el dominio para cada atributo de la entidad que luego servirán para definir el diseño lógico.

Dominio atributos:

- **nombre:** *varchar(50)*. Es decir, se asume nombres con máximo de 50 caracteres.
- **descripcion:** *varchar(100)*.
- **ambito:** uno a más valores del enunciado {banco,..., estado...}
- **millones:** números reales positivos

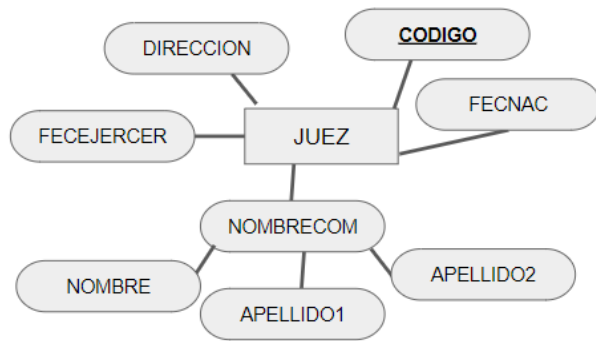
2.1.2 Entidad **juez**

La entidad **juez** presenta cuatro atributos procedentes del enunciado. No obstante, se ha decidido añadir un atributo extra (**codigo**) que será la clave primaria. **direccion** podría o no ser una clave primaria, en función de los valores que tomase. Por ejemplo, la dirección del domicilio particular sí podría ser una clave primaria, en cambio, si la dirección es de donde trabaja no ya que en esa dirección trabajarán más jueces probablemente.

Por otra parte, añadir a **direccion** (si es de juzgado) el nombre del juez o las fechas tampoco determinaría una clave primaria. Se podría dar el caso (improbable) de dos jueces con mismo nombre y apellidos, nacidos el mismo día y de la misma promoción. Se asume como **direccion** la del puesto de trabajo para la base de datos presentada en este trabajo.

Sin embargo, el número de magistrado (**codigo**) sí es un identificador único para cada juez (número de magistrado) y es una información que puede ser averiguada. Es por ello que se ha decidido agregar este atributo extra.

El nombre completo de una instancia **juez** (**nombrecom**) es un atributo compuesto univalorado, que se puede descomponer en tres atributos simples univalorados: **nombre**, **apellido1** y **apellido2**. Los demás atributos son simples univalorados.

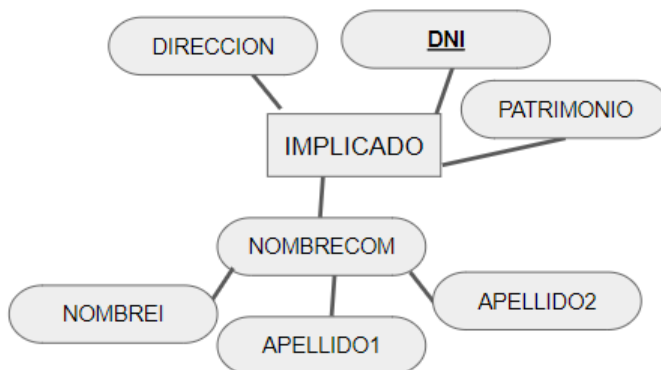


Dominio atributos:

- **codigo:** *smallint* ya que no se esperan tantos jueces en la base de datos como para ser *int*.
- **nombre, apellido1, apellido2:** *varchar(50)*
- **fecejercer:** fechas (*date* in MySQL).
- **fecejercer:** fechas
- **direccion:** *varchar(100)*

2.1.3. Entidad **implicado**

La entidad **implicado** presenta cuatro atributos, siendo **dni** la clave primaria.



Como con la entidad **juez**, **implicado** presenta un atributo simple compuesto, **nombrecom**, que se puede descomponer a su vez en tres atributos simples univalorados: **nombre**, **apellido1** y **apellido2**. Los demás atributos son simples univalorados.

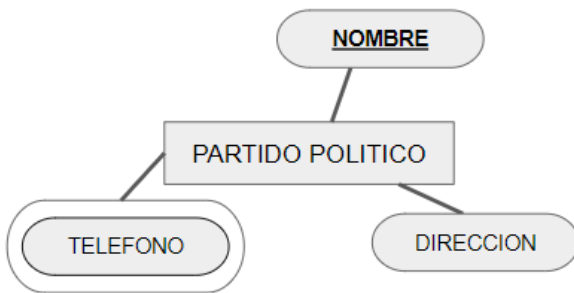
Dominio atributos:

- **dni:** *char(9)* ya que el dni es una cadena fija de 9 caracteres.
- **nombre, apellido1, apellido2:** *varchar(50)*
- **direccion:** *varchar(100)*

- **patrimonio:** números reales ya que un implicado podría tener deudas (valor negativo).

2.1.4. Entidad **partidopolitico**

La entidad **partidopolitico** presenta tres atributos siendo su **nombre** la clave primaria. Es importante destacar que el atributo **telefono** es simple multivalorado y formará una tabla diferente. Los otros dos son simples univalorados.

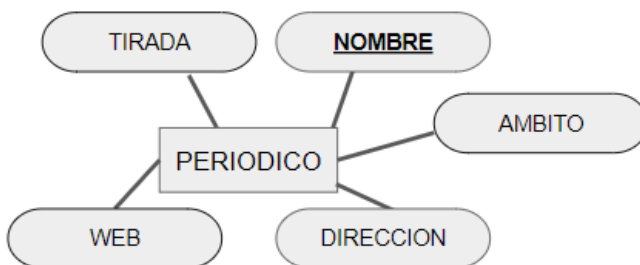


Dominio atributos:

- **nombre:** cadenas de caracteres de hasta 50 elementos *varchar(50)*.
- **telefono:** *char(9)* aunque también podría ser de tipo *numeric*.
- **direccion:** *varchar(100)*

2.1.5. Entidad **periodico**

La entidad **periodico** presenta cinco atributos siendo su nombre la clave principal. El atributo **tirada** representa si es en papel o digital. Todos los atributos son simples univalorados.

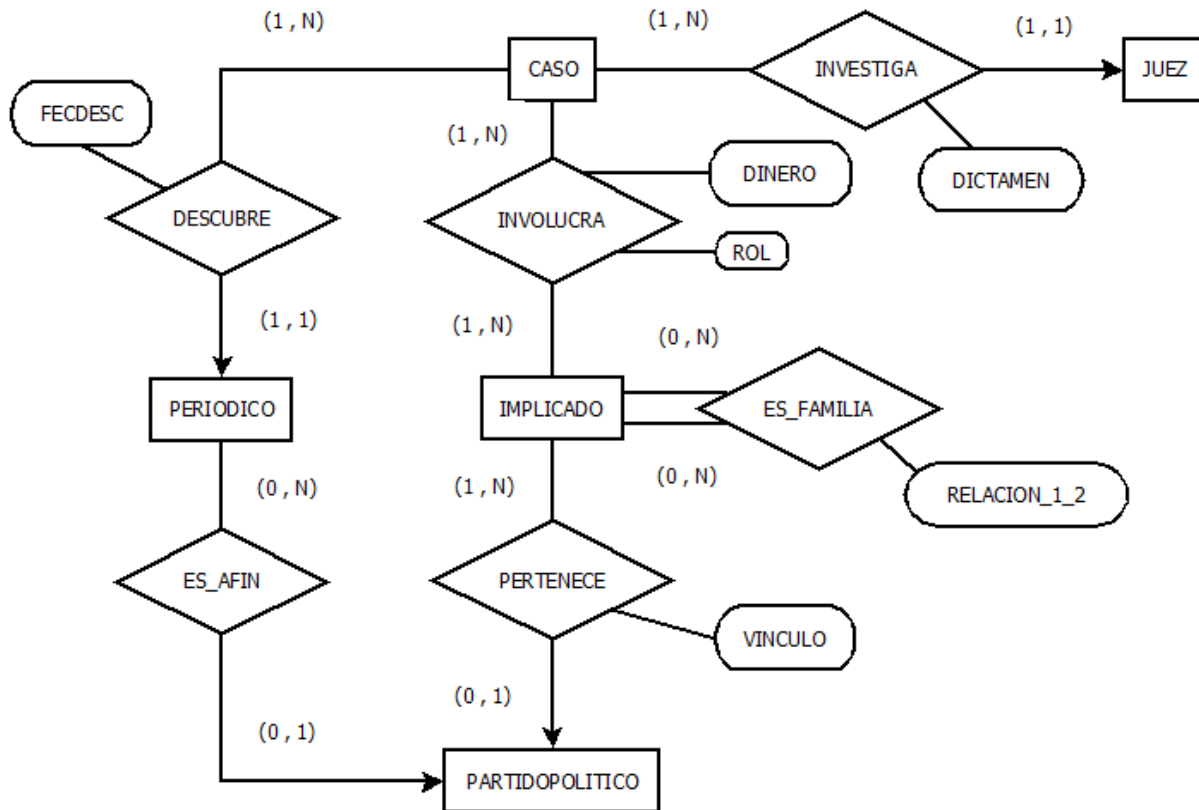


Dominio atributos:

- **nombre:** *varchar(50)*.
- **ambito:** un valor del conjunto del enunciado, {local..., internacional}
- **direccion:** *varchar(100)*
- **web:** *char(50)*
- **tirada:** un valor del conjunto {papel, digital}

2.2 Modelo Entidad-Relación

El diagrama entidad-relación es presentado a continuación con las entidades mencionadas.



Las relaciones existentes y sus correspondientes cardinalidades asumidas son:

1) **involucra**: relación N:N o 'varios con varios' ya que un caso puede involucrar a varios implicados y un implicado puede estar involucrado en varios casos.

- Cardinalidad **caso** (1,N). El número de casos en los que un implicado anda involucrado debe ser al menos uno para que se considere como implicado.
- Cardinalidad **implicado** (1,N). El número de implicados por caso debe ser al menos uno, de lo contrario no habría caso.

La relación **involucra** presenta dos atributos simples univalorados, rol del implicado en la instancia **caso** y dinero defraudado por el implicado, llamados **rol** y **dinero** respectivamente.

2) **investiga**: relación 1:N ya que un juez puede investigar (o haber investigado) varios casos pero un caso solo puede ser investigado por un juez. **caso** es el lado 'varios'.

- Cardinalidad juez (1,1). Se escoge (1,1) en vez de (0,1) ya que se asume que todo caso en la base de datos está siendo investigado (por un solo juez como ya se ha dicho).
- Cardinalidad caso (1,N). Se descarta (0,N) asumiendo que todo juez en la base de datos está activo, investigando algún caso.

La relación *investiga* presenta un atributo simple univalorado, **dictamen**, que se asume como un código alfanumérico de cuatro dígitos, *char*(4). El dictamen es la sentencia de la investigación.

3) **descubre**: relación 1:N ya que cada caso es descubierto por un periódico, pero un periódico puede descubrir varios casos. **caso** es el lado 'varios'.

- Cardinalidad **periodico** (1,1). Se escoge (1,1) en vez de (0,1) ya que se asume que todo caso en la base de datos se ha tenido que ser descubierto por un (solo) periódico.
- Cardinalidad **caso** (1,N). Se descarta (0,N) asumiendo que todo periódico en la base de datos ha descubierto un caso al menos.

La relación **descubre** tiene un atributo simple univalorado, **fecdesc**, de la fecha en el que **caso** es descubierto por **periodico**.

4) **es_afin**: relación 1:N ya que un periódico puede ser afín a un partido y un partido puede tener afinidad con varios periódicos. Periódico es el lado 'varios'.

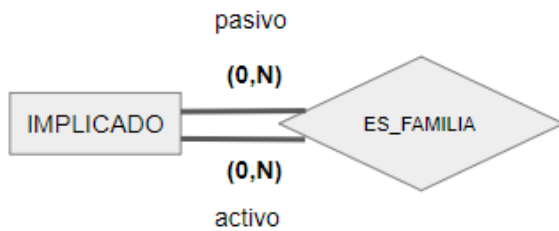
- Cardinalidad **partidopolitico** (0,1). Se escoge (0,1) ya que el enunciado dice explícitamente 'puede tener o no con un partido'.
- Cardinalidad **periodico** (0,N). Se escoge (0,N) ya que el enunciado dice explícitamente 'un partido político puede tener afinidad con varios periódicos o con ninguno.'

5) **pertenece**: relación 1:N ya que varios implicados pueden pertenecer solo a un partido. Se asume que un implicado no puede pertenecer a varios partidos.

- Cardinalidad **partidopolitico** (0,1). Se escoge (0,1) ya que el enunciado dice explícitamente 'pueden pertenecer o no a un partido'.
- Cardinalidad **implicado** (1,N). Se escoge (1,N) en vez de (0,N) ya que se asume que un partido tiene que tener a alguien implicado para aparecer 'señalado' como foco de corrupción y, por tanto, figurar en la base de datos.

La relación **pertenece** presenta un atributo simple univalorado, **vinculo**, que es la relación o no de una persona implicada con un partido político.

6) **es_familia**: relación N:N ya que varios implicados pueden ser a su vez familiares de varias personas. Es una relación reflexiva con la siguiente vista detalle, donde activo/pasivo se define como 'activo tiene una relación x con respecto a pasivo'.



Por ejemplo, en la frase 'María es la madre de Luisa' María es lo que se entiende como **activo** (el sujeto de la oración) mientras que Luisa sería el **pasivo** (el objeto de la oración).

La cardinalidad de tanto el lado activo como pasivo es $(0,N)$ ya que varios pueden o no ser parientes de varios.

El modelo entidad-relación sirve para identificar los agentes, las relaciones y los atributos de ambos conjuntos. Sin embargo, esto no es suficiente para el diseño de una base de datos ya que ciertas restricciones de integridad se han de imponer para evitar por ejemplo redundancia o inconsistencia de los datos. El paso a tablas del modelo entidad-relación así como las restricciones de integridad, algunas como la clave primaria ya comentadas, se tratan en el siguiente apartado.

3. Diseño lógico

3.1 Conceptualización de las tablas

Para el paso a tablas, se declaran las entidades y sus atributos. El atributo clave primaria se pone en primer lugar, subrayado. Después se añaden las tablas de relaciones (las que procedan) y se señalan su clave primaria(s) y ajena(s). A continuación se completan las tablas entidad con las claves ajenas. Finalmente, las tablas se complementan con otras restricciones de integridad.

Las tablas de la base de datos 'España Relimpia' son las siguientes. Las claves primarias son los atributos subrayados. El vínculo entre la clave primaria de una tabla con respecto a una ajena de otra viene expresada en colores.

- 1) caso(nombre,descripcion,millones,codjuez,dictamen,nombre_periodico,fecdesc)
- 2) ambito_caso(tipo_ambito,nombre_caso)
- 3) juez(codigo,nombre,apellido1,apellido2,fechnac,fecejercer,direccion)
- 4) implicado(dni,nombre,apellido1,apellido2,direccion,patrimonio,partido, vinculo)
- 5) partidopolitico(nombre,direccion)
- 6) telefono_partido(numero,partido)
- 7) periodico(nombre,tirada,web,ambito,direccion,partido)

Las siguientes relaciones del diseño conceptual son también pasadas a tablas:

- 8) involucra(**nombre caso**,**dni**,rol,dinero)
- 9) es_familia(**dni_implicado_1**,**dni_implicado_2**,relacion_1_con_2)

A continuación se explican los detalles de este modelo relacional.

3.1.1. Tabla 1 (**caso**)

Se añaden como claves ajenas los atributos **codjuez** y **nombre_periodico** procedentes de las claves primarias de las entidades **juez** y **periodico** al ser **descubre** e **investiga** relaciones 1:N, siendo **caso** el lado 'varios' en ambas relaciones. Además, se añaden los atributos procedentes de **descubre** e **investiga**, **dictamen** y **fecdesc** respectivamente, a la tabla **caso**.

No se hacen tablas **descubre** e **investiga** al ser relaciones 1:N, ambas con cardinalidad asumida para el lado 'varios' de (1,N), es decir, desdeñando la inclusión de valores nulos. En una relación 1:N, no es el caso aquí, si hubiese un número grande de nulos en el lado 'varios', se podría contemplar la creación de las tablas de relación por separado.

El atributo **ambito** se retira y pasa a formar la siguiente tabla.

3.1.2. Tabla 2 (**ambito_caso**)

Se crea la tabla **ambito_caso** a partir del atributo ambito de la entidad caso al ser un atributo multivalorado. La tabla ambito_caso presenta el tipo de ámbito y el nombre del caso como clave primaria, para poder distinguir una fila de otra de la tabla.

3.1.3. Tabla 3 (**juez**)

No se añaden atributos extra como claves ajenas.

3.1.4. Tabla 4 (**implicado**)

Se añade la clave primaria de **partidopolitico** (renombrada como **partido**) como clave ajena, procedente de la relación 1:N **pertenece** así como su atributo **vinculo**, al ser **implicado** el lado 'varios' de la relación. La justificación es idéntica a lo expuesto con la tabla 1 (**caso**), es decir, se asume que pocos implicados no van a pertenecer a un partido político.

3.1.5. Tabla 5 (**partidopolitico**)

No se añaden claves ajenas al conformar el lado 'uno' de la relación 1:N tanto para **es_afin** como para **pertenece**. El atributo **telefono**, al ser multivalorado, pasa a formar su propia tabla.

3.1.6. Tabla 6 (**telefono_partido**)

Tiene como clave primaria el número y como clave ajena el nombre del partido político, renombrado como **partido**.

3.1.7. Tabla 7 (**periodico**)

Se añade la clave primaria de **partidopolitico** como clave ajena por la relación 1:N **es_afin**. Se decide no crear una nueva tabla para **es_afin** ya que se asume que pocos periódicos no presentan afinidad con ningún partido.

3.1.8. Tabla 8 (**involucra**)

Se tabula esta relación al ser ‘varios a varios’, es decir, N:N. Las claves primarias y a su vez ajenas de **involucra** son las claves primarias de **caso** e **implicado**. Además, se añaden los dos atributos de la relación, el rol del implicado en el caso y el dinero defraudado por él/ella.

3.1.9. Tabla 9 (**es_familia**)

Se tabula esta relación al ser N:N. Las dos claves primarias y a su vez ajenas corresponden a la clave primaria **implicado**. Además tiene como atributo la relación de parentesco del sujeto (**dni_implicado_1**) con respecto al objeto (**dni_implicado_2**), es decir, ‘<sujeto> es <relacion_1_con_2> de <objeto>’.

3.2 Implementación de las tablas

Se procede a añadir otras restricciones de integralidad a las tablas del apartado anterior. Algunas restricciones de integridad ya han sido comentadas, estas son:

- Tipo de datos escogidos para los atributos de las entidades (apartado 2.1)
- Claves primarias y ajenas de las tablas (apartado 3.1)

Por tanto, en este apartado se comentan solo las otras restricciones de integridad implementadas en la base de datos.

El orden aquí expuesto no es el mismo al de creación de las tablas del código, donde primero se crearon las tablas que no tienen claves ajenas y luego las demás.

3.2.1. Tabla 1 (**caso**)

Se ha asumido como identificador para la sentencia de un caso, **dictamen**, un código alfanumérico de 10 caracteres.

descripcion. Se añade la restricción **unique** ya que cada caso tiene unas circunstancias diferentes y, por tanto, podría haber sido una clave primaria. Se ha dejado con la posibilidad de que tenga valores nulos, en el caso que no hubiese una descripción lista por ser un caso recientemente descubierto.

millones. tampoco puede tomar valores nulos ya que se asume que si hay un caso de corrupción es porque una cantidad ha sido defraudada y se asume que es una información fácil de averiguar. Se asumen unos valores máximos del orden de un billón hispano (euros). Se añade una restricción **check** para que las cantidades insertadas sean mayores que cero.

fecdesc. es una fecha y no puede tener valores nulos ya que para que sepa de la existencia de un caso, este debe haber sido descubierto antes por algún periódico, lo cual es fácil de constatar cuando ocurrió.

On delete restrict. Si se borra el juez o el periódico que no se borre el caso.

On delete cascade. Si se actualizan las claves primarias de las instancias **juez** y **periodico** que se modifique también en **caso**.

```
create table caso (  
  nombre varchar(50),  
  descripcion varchar(100) unique,  
  millones numeric(15,2) not null, -- orden de un billon hispano euros (2 decimales)  
  codjuez smallint,  
  dictamen varchar(10), -- codigo dictamen del juez para el caso  
  nombre_periodico varchar(50),  
  fecDesc date not null,  
  primary key (nombre),  
  foreign key (codjuez) references juez(codigo)  
  on delete restrict  
  on update cascade,  
  foreign key (nombre_periodico) references periodico(nombre)  
  on delete restrict  
  on update cascade  
);
```

```
alter table caso -- millones debe ser una cantidad positiva  
add constraint millones check (millones >0);
```

3.2.2. Tabla 2 (*ambito_caso*)

En este, al depender esta tabla de **caso**, si se borra una instancia **caso** también se ha de borrarse su ámbito. De ahí **on delete cascade**, a diferencia de la tabla anterior (**caso**).

```
create table ambito_caso( -- atributo multivalorado  
tipo_ambito varchar(50),  
nombre_caso varchar(50),  
primary key (tipo_ambito,nombre_caso),  
foreign key (nombre_caso) references caso(nombre)  
on delete cascade  
on update cascade);
```

3.2.3. Tabla 3 (*juez*)

Varios atributos se han restringido a no nulos ya que se asume como información que se puede obtener sin complicaciones. **direccion** no se ha definido como única por la duda expresada de entre si es del puesto de trabajo o la del domicilio. Como se expresó, se ha asumido que se refiere a la dirección del puesto de trabajo siendo, por tanto, no única.

```
create table juez(  
codigo smallint,-- no se esperan tantos jueces como para hacerlo int  
nombre varchar(50) not null,-- siguiendo ejemplos de clase, como para direcciones o  
apellidos  
apellido1 varchar(50) not null,  
apellido2 varchar(50) not null,  
fecNac date,  
fecEjercer date not null,  
direccion not null varchar(100),  
primary key (codigo)  
);
```

3.2.4. Tabla 4 (*implicado*)

Asumo que el nombre y apellidos se puede obtener de forma fácil, de ahí la restricción **not null**. La explicación de la configuración **on delete restrict** y **on update cascade** son idénticas a lo explicado en **caso**. En este caso el atributo **direccion** no se presupone como único ya que los implicados pueden ser familia y, por tanto, residir en el mismo domicilio. Se ha optado por permitir que **direccion** tome valores nulos por si el implicado estuviera en paradero desconocido.

```
create table implicado(  
dni char(9),  
nombre char(50) not null,
```

```

apellido1 char(50) not null,
apellido2 char(50) not null,
direccion varchar(100),
patrimonio numeric(15,2),
partido varchar(50),
vinculo_partido varchar(50),
primary key (dni),
foreign key(partido) references partidopolitico(nombre)
on delete restrict -- si borro partido, no borro implicados
on update cascade-- actualizo nombre partido
);

```

3.2.5. Tabla 5 (*partidopolitico*)

El atributo **telefono**, al ser multivalorado, se retira ya que pasará a formar una tabla aparte. El atributo **direccion** se ha restringido a no nulo debido a, de nuevo, se espera que se pueda obtener esa información de manera simple. También se ha asumido la dirección de cada partido como única y no nula, ya que puede ser fácilmente averiguada.

```

create table partidopolitico(
nombre varchar(50),
direccion varchar(100) not null unique,
primary key(nombre)
);

```

3.2.6. Tabla 6 (*telefono_partido*)

El razonamiento seguido es idéntico a la Tabla 2 (*ambito_caso*).

```

create table telefono_partido(
numero char(9),
partido varchar(50),
primary key (numero),
foreign key(partido) references partidopolitico(nombre)
on delete cascade -- borro partido, borro su telefono
on update cascade -- actualizo el nombre del partido si se cambia
);

```

3.2.7. Tabla 7 (*periodico*)

De nuevo, se impone **not null** a atributos asumiendo que su información es fácil de averiguar. Se impone además **on delete restrict** para no borrar un periódico si un partido con el que es afín se borra de la base de datos. Se añade un **check** para restringir los valores de tirada a 'digital' o 'papel'. El atributo **direccion** se ha vuelto a declarar como no nulo y único.

```

create table periodico(
nombre varchar(50),
tirada varchar(7) not null, -- papel o digital
web varchar(50) not null,
ambito varchar(13) not null,-- maximos caracteres: internacional (13)
direccion varchar(100) not null unique,
partido varchar(50),
primary key(nombre),
foreign key(partido) references partidopolitico(nombre)
on delete restrict -- si borro partido, no borro periodico
on update cascade);-- actualizo nombre partido

```

```

alter table periodico -- tirada solo puede adpotar dos valores
add constraint tirada check (tirada= 'papel' or tirada ='digital');

```

3.2.8. Tabla 8 (*involucra*)

La asunción de valores no nulos se ha realizado como en casos anteriores. En esta tabla sí se aplica **on delete cascade** ya que no tiene sentido borrar a algún (presunto) corrupto pero que continúe su implicación en la base de datos, lo mismo si se borra el caso.

```

create table involucra(
nombre_caso varchar(50),
dni char(9),
rol varchar(50) not null,
dinero numeric(9,2) not null, -- millones de euros
primary key (nombre_caso, dni),
foreign key (nombre_caso) references caso(nombre)
on delete cascade -- para involucrar a alguien debe existir
on update cascade
,
foreign key (dni) references implicado(dni)
on delete cascade
on update cascade
);

```

3.2.9. Tabla 9 (*es_familia*)

En esta tabla también se aplica **on delete cascade** ya que no tiene sentido hacer una referencia a una relación de personas que no existen para la base de datos. Finalmente, también se añade un disparador (ver el apartado 4) para impedir la redundancia de los datos.

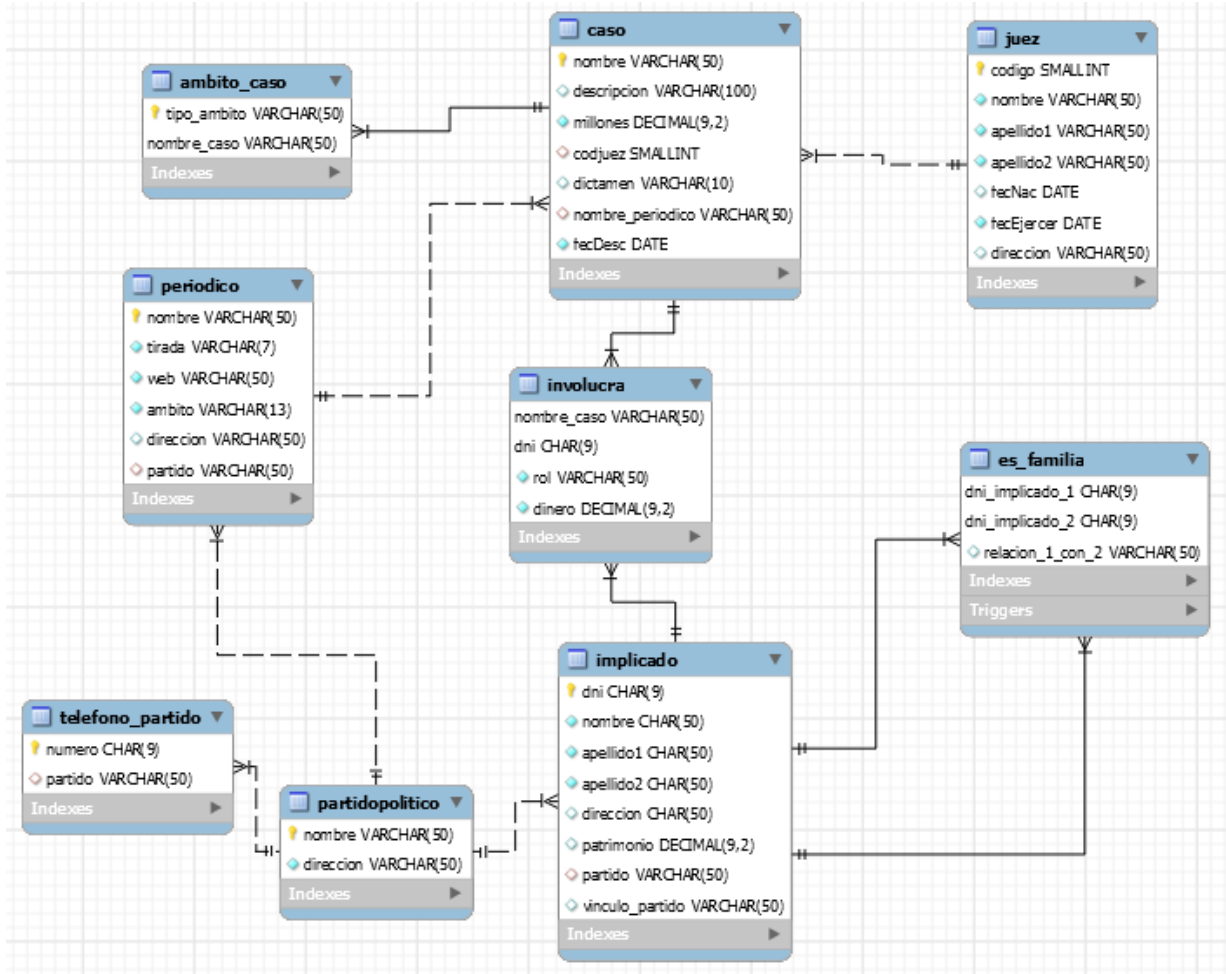
```
create table es_familia(
dni_implicado_1 char(9),
dni_implicado_2 char(9),
relacion_1_con_2 varchar(50),
primary key (dni_implicado_1, dni_implicado_2),
foreign key (dni_implicado_1) references implicado(dni)
on delete cascade
on update cascade,
foreign key (dni_implicado_2) references implicado(dni)
on delete cascade
on update cascade );
```

Con esta configuración de tabla no se pueden expresar valores nulos ya que los valores de las claves primarias no pueden ser nulas, es decir, sólo las relaciones de parentesco que existen figuran en esta tabla. Las cardinalidades asociadas a esta tabla eran (0, N) por tanto, relaciones que no aparecen se asumen nulas.

3.3 Representación modelo lógico

El modelo lógico es representado por medio de la herramienta 'Reverse Engineer' de MySQL. Esta representación sirve además como comprobación de la implementación del diagrama entidad-relación. Se observa en la representación del modelo lógico que las tablas tienen las claves primarias y el tipo de relación (1:N o N:N) deseados.

Como se esperaba **ambito_caso** muestra una 'identifying relationship' mientras que **telefono_partido** presenta una 'non-identifying relationship' a pesar de derivar a ambas tablas de atributos multivalorados. La razón es que, a diferencia de **telefono_partido**, en **ambito_caso** la clave ajena también forma parte de la clave primaria.



4. Implementación

En este apartado se detallan la implementación de la base de datos en MySQL. La sección comienza con un comentario sobre la inserción de datos en la propia base. Se prosigue tratando la creación del disparador para la tabla **es_familia**, la cual se podría considerar como la consulta número 0. Finalmente se muestran las consultas realizadas y su resultado como método de validación del código.

4.1 Inserción de datos

Se han cargado los datos siguiendo dos métodos: el código proporcionado en las diapositivas con una ligera modificación y el método **Insert into**.

Como se puede observar a continuación, el código usado para cargar la tabla **juez** es el de las diapositivas.


```
SELECT @@GLOBAL.secure_file_priv;
```

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/juez.csv'  
IGNORE INTO TABLE juez -- ignorar errores al insertar en la tabla 'juez'  
CHARACTER SET latin1  
FIELDS TERMINATED BY ';'   
LINES TERMINATED BY '\n'  
IGNORE 1 rows; -- ignoramos la línea con los nombres de las columnas
```

4.2 Disparador para la tabla **es_familia**

Para evitar redundancia en la información de la tabla del estilo 'activo tiene parentesco con pasivo' y 'pasivo tiene parentesco con activo' se recurre al siguiente disparador, el cual evalúa antes de insertar una nueva fila si ya está repetida esta información en la tabla. En el caso que la información esté repetida lanza un error.

```
DELIMITER $$  
create trigger noBA  
before insert on es_familia  
for each row  
begin  
    if exists      (select dni_implicado_1 as A, dni_implicado_2 as B  
                   from es_familia  
                   where dni_implicado_1=new.dni_implicado_2  
                   and dni_implicado_2=new.dni_implicado_1)  
  
    then  
        SIGNAL SQLSTATE '45000'  
        SET MESSAGE_TEXT = 'Este parentesco ya ha sido registrado a la inversa',  
        MYSQL_ERRNO = 1001;  
        END IF;  
  
END$$  
DELIMITER ;
```

Se prueba primero el disparador para el caso que la información no es redundante y no salta el error. Después se prueba con un caso que la información sí es redundante y en efecto el error salta en este segundo caso (dejado como comentario).

```
insert into es_familia values  
('71783331W','51183401C','pariente_lejano');
```

/*

-- Activando esta sentencia se ve que el disparador definido anteriormente para evitar
-- información redundante en es_familia funciona

insert into es_familia values

('91183101S','51183401C','pariente_cercano');

*/

4.3 Consultas realizadas

Antes de presentar las consultas y sus resultados, se muestran las tablas con los datos añadidos

1) **select * from juez;**

	codigo	nombre	apellido1	apellido2	fecNac	fecEjercer	direccion
►	1	Pablo	Ruz	Beitia	1972-06-09	1998-03-19	Ciudad de la Justicia, Madrid
	2	Juan	Romero	Rodriguez	1958-03-19	2000-05-19	Ciudad de la Justicia, Madrid
	3	Mercedes	Alaya	Perez	1967-06-25	1995-11-20	Juzgados de Barcelona, Barcelona
	4	Pablo	Ramirez	Lopez	1984-03-19	2005-08-24	Ciudad de la Justicia, Madrid
	5	Marta	Etayo	Garcia	1984-12-31	2000-05-19	Juzgados de Barcelona, Barcelona

2) **select * from partidopolitico;**

	nombre	direccion
►	Amarillo	Calle Laurel 34, 41007, Lugo
	Azul	Avenida La Paz 11, 28005, Madrid
	Rojo	Calle Pinto 54, 48005, Barcelona
	Verde	Calle Coimbra 54, 27009, Madrid

3) **select * from telefono_partido;**

	numero	partido
►	975684321	Amarillo
	918765499	Azul
	918947625	Azul
	956694625	Rojo
	939875462	Verde
	939995462	Verde

4) **select * from periodico;**

	nombre	tirada	web	ambito	direccion	partido
►	Diario Publico	digital	www.diariopublico.es	nacional	Via 21 28807, Madrid	Verde
	El Mundo	digital	www.elmundo.es	nacional	Via Constitución 28307, Zaragoza	NULL
	El Pais	papel	www.elpais.es	nacional	Avenia 12 29807, Teruel	Verde
	La Razon	papel	www.larazon.com	internacional	Calle Gernika 25807, Bibao	Rojo
	Vanguardia	digital	www.vanguardia.com	internacional	Via 1 28807, Madrid	Amarillo

5) ***select * from caso;***

	nombre	descripcion	millones	codjuez	dictamen	nombre_periodico	fecDesc
►	ERES	Descripcion ERES	400.25	1	0002	Vanguardia	2008-11-14
	Gurtel	Descripcion Gurtel	1000.00	1	0001	Vanguardia	2007-08-24
	Palau	Descripcion Palau	12567.10	5	0006	Diario Publico	2011-03-30
	Punica Malaya	Descripcion Punica Malaya	3000.00	3	0003	El Pais	2009-10-23
	Puyol	Descripcion Puyol	20000.00	4	0005	La Razon	2014-01-19
	Tarjetas Black	Descripcion Tarjetas Black	768.00	2	0004	El Mundo	2010-09-14

6) ***select * from ambito_caso;***

	dni	nombre	apellido1	apellido2	direccion	patrimonio	partido	vinculo_partido
►	41783001R	Isabel	Martinez	Heredia	Vigo	630.00	NULL	NULL
	51183401C	Juan	Molina	Perez	Madrid	34.00	Azul	afiliado
	51383601A	Carmen	Ferrero	Busquets	Madrid	250.00	Verde	afiliado
	61183401X	Gorka	Aguirre	Liceaga	Barcelona	5000.00	Amarillo	militante
	71783331W	Beatriz	Simon	Alegre	Valencia	500.00	Rojo	militante
	81684401T	Julia	Rodriguez	Velez	Madrid	24.00	Rojo	afiliado
	91183101S	Ramon	Martinez	Otero	Barcelona	12.00	NULL	NULL

7) ***select * from implicado;***

	dni	nombre	apellido1	apellido2	direccion	patrimonio	partido	vinculo_partido
►	41783001R	Isabel	Martinez	Heredia	Vigo	630.00	NULL	NULL
	51183401C	Juan	Molina	Perez	Madrid	34.00	Azul	afiliado
	51383601A	Carmen	Ferrero	Busquets	Madrid	250.00	Verde	afiliado
	61183401X	Gorka	Aguirre	Liceaga	Barcelona	5000.00	Amarillo	militante
	71783331W	Beatriz	Simon	Alegre	Valencia	500.00	Rojo	militante
	81684401T	Julia	Rodriguez	Velez	Madrid	24.00	Rojo	afiliado
	91183101S	Ramon	Martinez	Otero	Barcelona	12.00	NULL	NULL

8) ***select * from involucra;***

	nombre_caso	dni	rol	dinero
►	ERES	51183401C	colabora	5.00
	ERES	51383601A	jefe	4.60
	Gurtel	51183401C	jefe	20.00
	Gurtel	51383601A	colabora	13.00
	Gurtel	71783331W	colabora	2.45
	Palau	51383601A	colabora	10.00
	Punica Malaya	51183401C	colabora	7.50
	Punica Malaya	71783331W	jefe	50.00
	Puyol	81684401T	jefe	30.00
	Tarjetas Black	61183401X	jefe	6.00
	Tarjetas Black	91183101S	colabora	13.00

9) ***select * from es_familia;***

	dni_implicado_1	dni_implicado_2	relacion_1_con_2
▶	51183401C	91183101S	pariente_cercano
	71783331W	51183401C	pariente_lejano
	81684401T	51383601A	pariente_cercano

A continuación se muestran las consultas realizadas, tras haber poblado las tablas y creado el disparador de la tabla ***es_familia***.

-- 1) Nombre de los periódicos independientes, sin afinidad a ningún partido

select nombre

from periodico

where partido is null;

	nombre
▶	El Mundo

-- 2) Ciudad con el máximo número de corruptos

select direccion as ciudad_con_mas_corruptos, count(*) as numero_corruptos

from implicado

group by ciudad_con_mas_corruptos

order by numero_corruptos desc

limit 1; -- pide ciudad, no ciudades. Me centro en la del valor máximo

	ciudad_con_mas_corruptos	numero_corruptos
▶	Madrid	3

-- 3) Total dinero defraudado por partido si y solo si la persona está afiliada

-- Un implicado de un partido puede participar en X casos, toda la cantidad

-- de esos X casos será agregada al dinero total defraudado por partido

select implicado.partido as partido_politico, sum(involucra.dinero) as dinero_defraudado

from implicado, involucra

where implicado.dni=involucra.dni and implicado.partido is not null

group by partido_politico

order by dinero_defraudado desc;

	partido_politico	dinero_defraudado
▶	Rojos	82.45
	Azul	32.50
	Verde	27.60
	Amarillo	6.00

-- 4) Periódico que más casos ha descubierto

```
select nombre_periodico as periodico, count(*) as numero_casos
from caso
group by periodico
order by numero_casos desc
limit 1;
```

	periodico	numero_casos
▶	Vanguardia	2

-- 5) Número de familiares de 51183401C con casos de corrupcion

```
select count(*) as numero_familiares
from es_familia
where dni_implicado_1='51183401C' or dni_implicado_2='51183401C';
```

	numero_familiares
▶	2

-- 6) Partidos políticos con sede en Madrid

```
select nombre as partido, direccion
from partidopolitico
where direccion like '%Madrid%';-- contenga subcadena Madrid
```

	partido	direccion
▶	Azul	Avenida La Paz 11, 28005, Madrid
	Verde	Calle Coimbra 54, 27009, Madrid

-- 7) Jueces investigando casos con cuantías mayores de 900 millones de euros

```
select j.codigo as codigo_juez, c.nombre as nombre_caso, c.millones as cuantia_caso
from juez as j inner join caso as c on j.codigo=c.codjuez
where c.millones>900;
```

	codigo_juez	nombre_caso	cuantia_caso
▶	1	Gurtel	1000.00
	5	Palau	12567.10
	3	Punica Malaya	3000.00
	4	Puyol	20000.00

-- 8) Implicados que son parientes cercanos de otro implicado

```
select i.dni, f.relacion_1_con_2 as tiene_implicado_a
from implicado as i right join es_familia as f on (i.dni=f.dni_implicado_1 or
i.dni=f.dni_implicado_2)
where f.relacion_1_con_2='pariente_cercano';
```

	dni	tiene_implicado_a
▶	51183401C	pariente_cercano
	91183101S	pariente_cercano
	51383601A	pariente_cercano
	81684401T	pariente_cercano

-- 9) Partidos políticos con más de un implicado

select partido, count(*) as numero_implicados

from implicado

where partido is not null

group by partido

having numero_implicados>1;

	partido	numero_implicados
▶	Rojo	2

-- 10) Nombre y apellidos de implicados y jueces que tengan como nombre de pila 'Juan'

(SELECT nombre, apellido1,apellido2 FROM implicado where nombre = 'Juan')

union

(SELECT nombre, apellido1,apellido2 FROM juez where nombre = 'Juan');

	nombre	apellido1	apellido2
▶	Juan	Molina	Perez
	Juan	Romero	Rodriguez

-- 11) Partidos con implicados que no tienen ningún familiar implicado

select p.nombre as nombre_partido, i.dni as dni_sin_familiar

from partidopolitico as p inner join implicado as i on p.nombre=i.partido

where i.dni not in (select dni_implicado_1 from es_familia)

and i.dni not in (select dni_implicado_2 from es_familia);

	nombre_partido	dni_sin_familiar
▶	Amarillo	61183401X

-- 12) Todos los casos con al menos un implicado como colaborador

select distinct(nombre_caso)

from involucra

where rol='colabora';

	nombre_caso
▶	ERES
	Gurtel
	Palau
	Punica Malaya
	Tarjetas Black

```
-- 13) Jueces que nacieron entre 1950 y 1970
select codigo,fecNac
from juez
where fecNac between '1950-01-01' and '1970-01-01';
```

	codigo	fecNac
▶	2	1958-03-19
	3	1967-06-25

```
-- 14) Suma un millon al dinero defraudado de un caso si su ambito tiene como valor al
menos 'Estado'
select c.nombre as nombre_caso, c.millones as millones_antes, c.millones +1 as
millones_mas_1
from caso as c inner join ambito_caso as a on c.nombre=a.nombre_caso
where a.tipo_ambito = 'Estado';
```

	nombre_caso	millones_antes	millones_mas_1
▶	ERES	400.25	401.25
	Puyol	20000.00	20001.00
	Tarjetas Black	768.00	769.00

```
-- 15) Actualizar un numero de telefono de un partido politico
update telefono_partido
set numero= '999965499'
WHERE numero='918765499';
```

	numero	partido
	975684321	Amarillo
	918947625	Azul
▶	999965499	Azul
	956694625	Rojo
	939875462	Verde
	939995462	Verde

```
-- 16) Generar una vista con nombre de partido y su dirección junto
-- con el nombre de los caso(s) de corrupción del que algún militante suyo esté
involucrado
```

```
create view vista_partido_casos_asociados as
select p.nombre as nombre_partido, p.direccion as direccion, inv.nombre_caso as caso
from partidopolitico as p,involucra as inv,implicado as imp
where imp.partido=p.nombre and inv.dni=imp.dni and imp.vinculo_partido='militante';
```

```
select * from vista_partido_casos_asociados;
```

	nombre_partido	direccion	caso
▶	Amarillo	Calle Laurel 34, 41007, Lugo	Tarjetas Black
	Rojo	Calle Pinto 54, 48005, Barcelona	Gurtel
	Rojo	Calle Pinto 54, 48005, Barcelona	Punica Malaya

vista_partido_casos_asodados 34 x

-- 17) Generar una vista de los nombres de pila de los jueces y nombre de los periódicos de

-- ámbito 'internacional', sin duplicados

create view jueces_periodicos as

select distinct(j.nombre) as nombre_pila_juez, p.nombre as nombre_periodico

from juez as j, periodico as p

where p.ambito='internacional';

select * from jueces_periodicos;

	nombre_pila_juez	nombre_periodico
▶	Pablo	Vanguardia
	Pablo	La Razon
	Juan	Vanguardia
	Juan	La Razon
	Mercedes	Vanguardia
	Mercedes	La Razon
	Marta	Vanguardia
	Marta	La Razon

-- 18) Crear un trigger para impedir que un juez pueda llevar más de dos casos
DELIMITER \$\$

CREATE TRIGGER maxCasosPorJuez

before update ON caso

FOR EACH ROW

BEGIN

if exists (select codjuez, count(*) from caso group by codjuez

having count(*)=2 and codjuez=new.codjuez)

then

SIGNAL SQLSTATE '45000'

SET MESSAGE_TEXT = 'Este juez ya tiene dos casos', MYSQL_ERRNO = 1001;

-- update caso

-- set new.codjuez = old.codjuez;


```

END IF;
END$$
DELIMITER ;

```

```
select * from caso;
```

	nombre	descripcion	millones	codjuez	dictamen	nombre_periodico	fecDesc
▶	ERES	Descripcion ERES	400.25	1	0002	Vanguardia	2008-11-14
	Gurtel	Descripcion Gurtel	1000.00	1	0001	Vanguardia	2007-08-24
	Palau	Descripcion Palau	12567.10	5	0006	Diario Publico	2011-03-30
	Punica Malaya	Descripcion Punica Malaya	3000.00	3	0003	El Pais	2009-10-23
	Puyol	Descripcion Puyol	20000.00	4	0005	La Razon	2014-01-19
	Tarjetas Black	Descripcion Tarjetas Black	768.00	2	0004	El Mundo	2010-09-14
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```

update caso
set codjuez = 003
WHERE nombre='Puyol';

```

```
select * from caso;
```

	nombre	descripcion	millones	codjuez	dictamen	nombre_periodico	fecDesc
▶	ERES	Descripcion ERES	400.25	1	0002	Vanguardia	2008-11-14
	Gurtel	Descripcion Gurtel	1000.00	1	0001	Vanguardia	2007-08-24
	Palau	Descripcion Palau	12567.10	5	0006	Diario Publico	2011-03-30
	Punica Malaya	Descripcion Punica Malaya	3000.00	3	0003	El Pais	2009-10-23
	Puyol	Descripcion Puyol	20000.00	3	0005	La Razon	2014-01-19
	Tarjetas Black	Descripcion Tarjetas Black	768.00	2	0004	El Mundo	2010-09-14
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```

update caso
set codjuez = 001
WHERE nombre='Puyol';

```

Y el error que genera es el que se declaró, en el caso que se el juez en cuestión tenga ya dos casos.

✖ 43 14:16:21 update caso set codjuez = 001 WHERE nombre='Puyol'

Error Code: 1001. Este juez ya tiene dos casos