



DEPARTAMENTO  
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA



Departamento de Computación,  
Facultad de Ciencias Exactas y Naturales,  
Universidad de Buenos Aires

# Trabajo Práctico 1

## Sistema de Voto Electrónico

### Bases de Datos

Primer Cuatrimestre de 2015

### Grupo

Apellido y Nombre	LU	E-mail
Heredia, Martín	146/11	<a href="mailto:martin.herediaf@gmail.com">martin.herediaf@gmail.com</a>
Requeni, Gastón	400/11	<a href="mailto:grequeni@hotmail.com">grequeni@hotmail.com</a>
Vega, Leandro	698/11	<a href="mailto:leandrogvega@gmail.com">leandrogvega@gmail.com</a>
Vita, Sebastián	149/11	<a href="mailto:sebastian.vita@yahoo.com.ar">sebastian.vita@yahoo.com.ar</a>

# Índice

<b>1. Descripción del Problema</b>	<b>2</b>
1.1. Alcance . . . . .	2
1.2. Padrón electoral . . . . .	2
1.3. Datos históricos de las votaciones . . . . .	2
1.4. Otros datos tenidos en cuenta . . . . .	2
<b>2. Diagrama de Entidad Relación</b>	<b>3</b>
2.1. Restricciones . . . . .	4
<b>3. Modelo Relacional</b>	<b>5</b>
<b>4. Asumimos</b>	<b>10</b>
<b>5. Diseño Físico</b>	<b>11</b>
5.1. Esquema . . . . .	11
5.2. Tablas . . . . .	11
<b>6. Funcionalidades Implementadas</b>	<b>15</b>
<b>7. Conclusiones</b>	<b>17</b>

## 1. Descripción del Problema

El gobierno de una nación se acerca a nosotros con el objetivo de crear un sistema de voto electrónico para futuras elecciones.

### 1.1. Alcance

El alcance del sistema será todo tipo de votación nacional, provincial y municipal, ya sea para la elección de cargos públicos como la realización de consultas populares a la ciudadanía.

En esta primera etapa, en el caso de las elecciones de cargos públicos, sólo tendremos en cuenta un cargo por elección.

Cada votación abarcará ciertos municipios o provincias, pero no ambos a la vez. Esto significa que si una votación es a nivel provincial, abarcará las provincias correspondientes, pero no se establecerá una relación directa con los municipios que integran cada provincia; si una votación abarca una provincia, automáticamente abarcará todos los municipios.

En cuanto a los tipos de votación, en principio estableceremos unos tipos básicos, permitiendo a futuro nuevos tipos creados por el administrador del sistema.

### 1.2. Padrón electoral

El sistema aquí introducido incluirá la gestión de un padrón electoral, que para cada ciudadano indicará la mesa y el centro de votación donde debe votar. Se proveerá una interfaz de alta y modificación del padrón, que permitirá ingresar nuevos ciudadanos al mismo, como también determinar una fecha de defunción cuando corresponda. Los datos de los nuevos ciudadanos deberán ser provistos junto con el centro de votación y el número de mesa donde voten.

El padrón NO podrá ser modificado durante una votación. Esto se debe a que mientras haya una votación en curso, el sistema realizará validaciones en cuanto a que cada persona vote donde el padrón lo indique.

### 1.3. Datos históricos de las votaciones

Se almacenará la historia de todas las votaciones realizadas mediante este sistema, considerando fecha de inicio y fin de las mismas, así como también el alcance y el tipo. Si se tratare de una elección de un cargo público, se preservarán los datos de los candidatos (incluyendo el partido político) y los votos recibidos por cada uno en total y en cada mesa. En el caso de las consultas populares se mantendrán los datos de las distintas opciones y los votos en total y por cada mesa.

Además por cada mesa almacenaremos quiénes fueron las autoridades: presidente, vicepresidente y fiscales. De todos ellos guardaremos los datos personales y en el caso de los fiscales, el partido político que representaban.

Por último guardaremos la lista de ciudadanos que votó efectivamente en cada mesa en cada votación. Dado que sabemos la fecha en que un ciudadano ingresó al padrón y conocemos también las fechas de inicio de las votaciones, podemos determinar si efectivamente a un ciudadano le correspondía o no votar y si fue o no a votar cuando le correspondía.

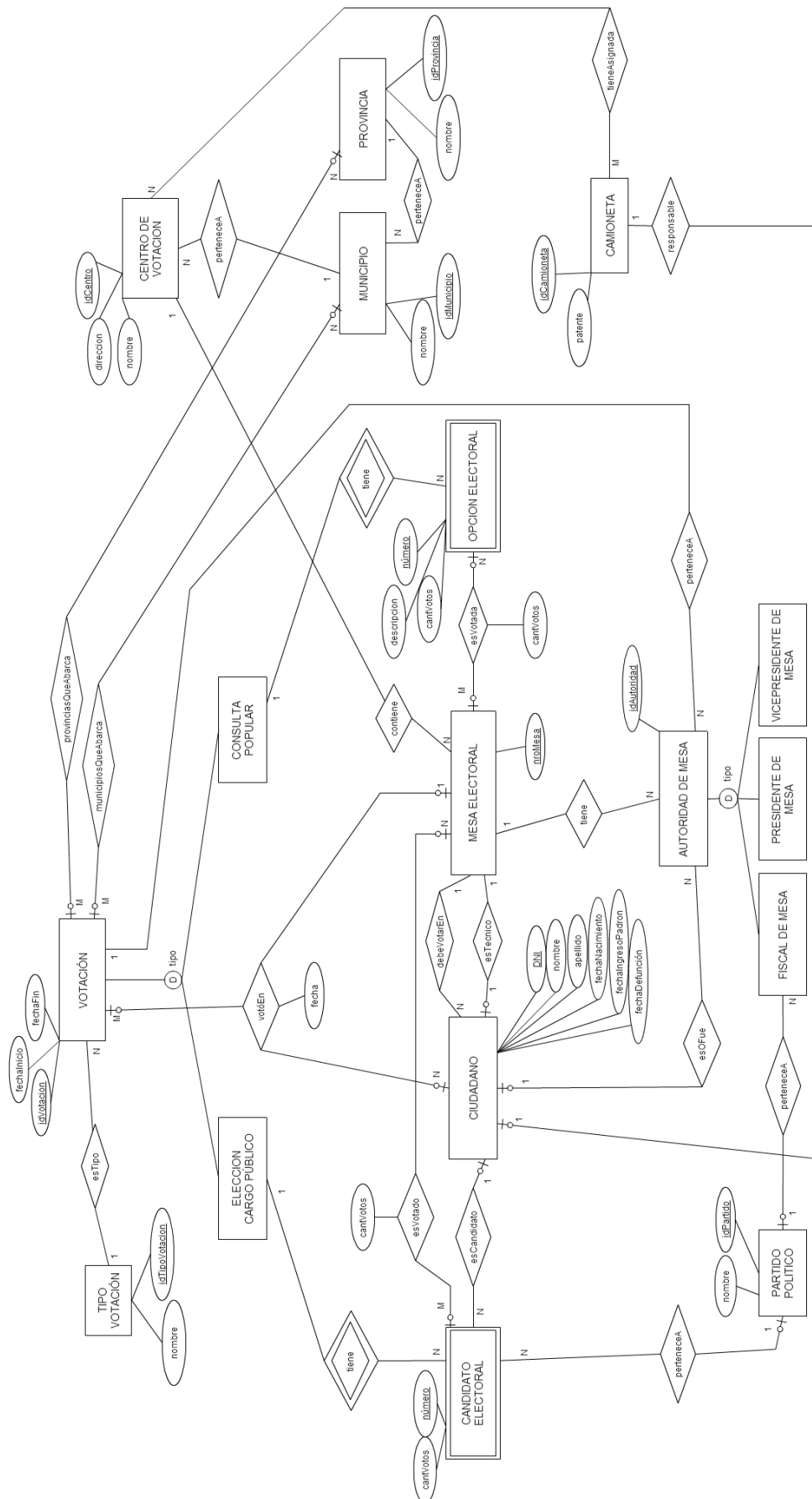
Una observación no menor sobre lo antedicho es que no guardamos a quién/qué votó cada ciudadano, sólo registramos el acto de votación. Esto permite preservar la confidencialidad del voto. La cantidad de votos por candidato o por opción electoral, se conocen por cada mesa (no por cada votante).

### 1.4. Otros datos tenidos en cuenta

Por cada mesa almacenamos los datos del técnico asignado a la máquina.

Por cada centro de votación almacenamos los datos de las camionetas asignadas al centro, que reemplazarán máquinas en caso de que se rompan, así como los datos del conductor asignado a cada una de ellas.

## 2. Diagrama de Entidad Relación



## 2.1. Restricciones

### ■ De las interrelaciones

- Los fiscales de una misma mesa son de distintos partidos políticos.
- Si un ciudadano es fiscal y candidato en una misma elección, entonces en ambas relaciones debe estar asociado al mismo partido político.
- Una votación se relaciona con al menos un municipio ó una provincia, pero no puede estar relacionado con un municipio y una provincia a la vez (es decir que los escenarios posibles son: votación que abarca de uno a muchos municipios o votación que abarca de una a muchas provincias, pero no puede abarcar municipios y provincias).
- Para cada elección, todo ciudadano es o bien fiscal, o bien presidente, o bien vicepresidente, o bien técnico, o ninguna de las anteriores, de una única mesa (en esa elección) (es decir que no puede haber un ciudadano que cumpla dos roles de esos en la misma votación).
- Dada una votación, participa en la relación `VotóEn` sólo las mesas que pertenecen a centros de votación cuyo municipio (o la provincia de éste) es abarcado por la votación.
- Un candidato o una opción electoral sólo pueden ser votados en mesas que pertenecen a algún centro de votación cuyo municipio (o la provincia de éste) es abarcado por la votación.
- Cada mesa tiene exactamente un presidente, un vicepresidente, y cero o más fiscales.
- Si un ciudadano tiene `fechaDeDefunción` distinto de null, entonces no puede participar en ninguna relación asociada a una votación con `fechaInicio` mayor a `fechaDeDefunción`.
- Un ciudadano no puede participar en ninguna relación asociada a una votación tal que `fechaInicio` sea anterior a `fechaIngresoPadrón` del ciudadano.

### ■ De los atributos

- La relación `VotóEn` no puede tener el atributo `fecha` mayor a `fechaFin` ni menor a `fechaInicio` de la votación a la que corresponde.
- Una elección cumple que el atributo `fechaFin` es menor a `fechaActual` y ninguno de ellos es nulo.
- La cantidad de votos de un candidato electoral debe ser igual a la suma de la cantidad de votos en cada tupla de la relación `esVotado` (asociadas a ese candidato).
- La cantidad de votos de una opción electoral debe ser igual a la suma de la cantidad de votos en cada tupla de la relación `esVotada` (asociadas a esa opción electoral).
- Los atributos `cantVotos` de las entidades opción electoral y candidato electoral, y de las interrelaciones `esVotado` y `esVotada`, son todos no nulos y mayores o iguales que cero.
- El atributo `fechaIngresoPadrón` de un ciudadano debe ser mayor que `fechaDeNacimiento` y menor que `fechaDeDefunción`.

### 3. Modelo Relacional

**Votación**(idVotación, fechaInicio, fechaFin, tipo, idTipoVotación)

PK = CK = {idVotación}

FK = {idTipoVotación}

- *Votación.idTipoVotación* debe estar en *TipoVotación.idTipoVotación*
- *Votación.idVotación* debe estar en *AutoridadDeMesa.idVotación*
- *Votación.idVotación* puede no estar en *ProvinciasQueAbarcaVotación.idVotación*
- *Votación.idVotación* puede no estar en *MunicipiosQueAbarcaVotación.idVotación*
- *Votación.idVotación* puede no estar en *VotoEn.idVotación*
- *Votación.tipo* debe ser “ELECCIÓN CARGO PÚBLICO” ó “CONSULTA POPULAR”
- *Votación.idVotación* debe estar en *ElecciónCargoPúblico.idVotación* ó (exclusivo) en *ConsultaPopular.idVotación*

**TipoVotación**(idTipoVotación, Nombre)

PK = CK = {idTipoVotación}

- *TipoVotación.idTipoVotación* debe estar en *Votación.idTipoVotación*

**ElecciónCargoPúblico**(idVotación)

PK = CK = FK = {idVotación}

- *ElecciónCargoPúblico.idVotación* debe estar en *Votación.idVotación*
- *ElecciónCargoPúblico.idVotación* debe estar en *CandidatoElectoral.idVotación*

**ConsultaPopular**(idVotación)

PK = CK = FK = {idVotación}

- *ConsultaPopular.idVotación* debe estar en *Votación.idVotación*
- *ConsultaPopular.idVotación* debe estar en *OpciónElectoral.idVotación*

**Ciudadano**(DNI, Nombre, Apellido, FechaNacimiento, FechaIngresoPadron, FechaDefunción, NúmeroMesaPadrón)

PK = CK = {DNI}

FK = {NúmeroMesaPadrón}

- *Ciudadano.DNI* puede no estar en *CandidatoElectoral.DNI*
- *Ciudadano.DNI* puede no estar en *VotoEn.DNI*
- *Ciudadano.DNI* puede no estar en *MesaElectoral.DNITécnico*
- *Ciudadano.NúmeroMesaPadrón* debe estar en *MesaElectoral.NúmeroMesa*

- *Ciudadano.DNI* puede no estar en *AutoridadDeMesa.DNI*
- *Ciudadano.DNI* puede no estar en *Camioneta.DNIResponsable*

### **MesaElectoral**(NúmeroMesa, DNITécnico, idCentro)

PK = CK = {NúmeroMesa}

FK = {DNITécnico, idCentro}

- *MesaElectoral.DNITécnico* debe estar en *Ciudadano.DNI*
- *MesaElectoral.NúmeroMesa* debe estar en *Ciudadano.NúmeroMesaPadrón*
- *MesaElectoral.NúmeroMesa* puede no estar en *VotoEn.NúmeroMesa*
- *MesaElectoral.NúmeroMesa* puede no estar en *CandidatoEsVotado.NúmeroMesa*
- *MesaElectoral.idCentro* debe estar en *CentroDeVotación.idCentro*
- *MesaElectoral.NúmeroMesa* puede no estar en *OpciónElectoralEsVotada.NúmeroMesa*
- *MesaElectoral.NúmeroMesa* debe estar en *AutoridadDeMesa.NúmeroMesa*

### **VotoEn**(DNI, idVotación, NúmeroMesa, fecha)

PK = CK = {(DNI, idVotación)}

FK = {DNI, idVotación, NúmeroMesa}

- *VotoEn.DNI* debe estar en *Ciudadano.DNI*
- *VotoEn.NúmeroMesa* debe estar en *MesaElectoral.NúmeroMesa*
- *VotoEn.idVotación* debe estar en *Votación.idVotación*

### **CandidatoElectoral**(idVotación, Número, CantVotos, DNI, idPartido)

PK = CK = {(idVotación, Número)}

FK = {idVotación, DNI, idPartido}

- (*CandidatoElectoral.idVotación*, *CandidatoElectoral.Número*) puede no estar en (*CandidatoEsVotado.idVotación*, *CandidatoEsVotado.NúmeroCandidato*)
- *CandidatoElectoral.idVotación* debe estar en *ElecciónCargoPúblico.idVotación*
- *CandidatoElectoral.idPartido* debe estar en *PartidoPolítico.idPartido*
- *CandidatoElectoral.DNI* debe estar en *Ciudadano.DNI*

### **CandidatoEsVotado**(idVotación, NúmeroCandidato, NúmeroMesa, cantVotos)

PK = CK = {(idVotación, NúmeroCandidato, NúmeroMesa)}

FK = {(idVotación, NúmeroCandidato), NúmeroMesa}

- (*CandidatoEsVotado.idVotación*, *CandidatoEsVotado.NúmeroCandidato*) debe estar en (*CandidatoElectoral.idVotación*, *CandidatoElectoral.Número*)
- *CandidatoEsVotado.NúmeroMesa* debe estar en *MesaElectoral.NúmeroMesa*

**OpciónElectoral**(idVotación, Número, descripción, cantVotos)

PK = CK = {idVotación, Número}

FK = {idVotación}

- (*OpciónElectoral.idVotación*, *OpciónElectoral.Número*) puede no estar en (*OpciónElectoralEsVotada.idVotación*, *OpciónElectoralEsVotada.NúmeroOpción*)
- *OpciónElectoral.idVotación* debe estar en *ConsultaPopular.idVotación*

**OpciónElectoralEsVotada**(idVotación, NúmeroOpción, NúmeroMesa, cantVotos)

PK = CK = {idVotación, NúmeroOpción, NúmeroMesa}

FK = {idVotación, NúmeroOpción, NúmeroMesa}

- (*OpciónElectoralEsVotada.idVotación*, *OpciónElectoralEsVotada.NúmeroOpción*) debe estar en (*OpciónElectoral.idVotación*, *OpciónElectoral.Número*)
- *OpciónElectoralEsVotada.NúmeroMesa* debe estar en *MesaElectoral.NúmeroMesa*

**AutoridadDeMesa**(idAutoridad, tipo, DNI, NúmeroMesa, idVotación)

PK = CK = {idAutoridad}

FK = {DNI, NúmeroMesa, idVotación}

- *AutoridadDeMesa.DNI* debe estar en *Ciudadano.DNI*
- *AutoridadDeMesa.NúmeroMesa* debe estar en *MesaElectoral.NúmeroMesa*
- *AutoridadDeMesa.idVotación* debe estar en *Votación.idVotación*
- *AutoridadDeMesa.tipo* debe ser “FISCAL DE MESA” ó “PRESIDENTE DE MESA” ó “VICEPRESIDENTE DE MESA”
- *AutoridadDeMesa.idAutoridad* puede no estar en *FiscalDeMesa.idAutoridad*

**FiscalDeMesa**(idAutoridad, idPartido)

PK = CK = {idAutoridad, idPartido}

FK = {idAutoridad}

- *FiscalDeMesa.idAutoridad* debe estar en *AutoridadDeMesa.idAutoridad*
- *FiscalDeMesa.idPartido* debe estar en *PartidoPolítico.idPartido*

**PartidoPolítico**(idPartido, Nombre)

PK = CK = {idPartido}

- *PartidoPolítico.idPartido* puede no estar en *CandidatoElectoral.idPartido*
- *PartidoPolítico.idPartido* puede no estar en *FiscalDeMesa.idPartido*

**CentroDeVotación**(idCentro, nombre, dirección, idMunicipio)

PK = CK = {idCentro}

FK = {idMunicipio}



- *CentroDeVotación.idCentro* debe estar en *MesaElectoral.idCentro*
- *CentroDeVotación.idMunicipio* debe estar en *Municipio.idMunicipio*
- *CentroDeVotación.idCentro* debe estar en *Camioneta\_X\_CentroDeVotación.idCentro*

#### **Camioneta**(idCamioneta, patente, DNIResponsable)

PK = {idCamioneta}

CK = {idCamioneta, patente}

FK = {DNIResponsable}

- *Camioneta.DNIResponsable* debe estar en *Ciudadano.DNI*
- *Camioneta.idCamioneta* debe estar en *Camioneta\_X\_CentroDeVotación.idCamioneta*

#### **Camioneta\_X\_CentroDeVotación**(idCentro, idCamioneta)

PK = CK = {(idCentro, idCamioneta)}

FK = {idCentro, idCamioneta}

- *Camioneta\_X\_CentroDeVotación.idCamioneta* debe estar en *Camioneta.idCamioneta*
- *Camioneta\_X\_CentroDeVotación.idCentro* debe estar en *CentroDeVotación.idCentro*

#### **Provincia**(idProvincia, nombre)

PK = CK = {idProvincia}

- *Provincia.idProvincia* puede no estar en *ProvinciasQueAbarcaVotación.idProvincia*
- *Provincia.idProvincia* debe estar en *Municipio.idProvincia*

#### **Municipio**(idMunicipio, nombre, idProvincia)

PK = CK = {idMunicipio}

FK = {idProvincia}

- *Municipio.idMunicipio* debe estar en *CentroDeVotación.idMunicipio*
- *Municipio.idProvincia* puede no estar en *MunicipiosQueAbarcaVotación.idProvincia*
- *Municipio.idProvincia* debe estar en *Provincia.idProvincia*

#### **ProvinciasQueAbarcaVotación**(idVotación, idProvincia)

PK = CK = {(idVotación, idProvincia)}

FK = {idVotación, idProvincia}

- *ProvinciasQueAbarcaVotación.idVotación* debe estar en *Votación.idVotación*
- *ProvinciasQueAbarcaVotación.idProvincia* debe estar en *Provincia.idProvincia*

#### **MunicipiosQueAbarcaVotación**(idVotación, idMunicipio)

PK = CK = {(idVotación, idMunicipio)}

FK = {idVotación, idMunicipio}

- *MunicipiosQueAbarcaVotación.idVotación* debe estar en *Votación.idVotación*
- *MunicipiosQueAbarcaVotación.idMunicipio* debe estar en *Municipio.idMunicipio*

## 4. Asumimos

Para realizar este trabajo, asumimos:

- La gestión del padrón de ciudadanos es correcta, en el sentido que se respetan las leyes de creación del padrón según corresponda (en cuanto a distancia al lugar de votación, distribución por apellido, etc.). Dado que estos datos son ingresados por un agente externo, el sistema se limita a confiar en esta información, y de hecho se basa en ella para realizar otras validaciones.
- Cualquier abarca o bien todas las provincias, o bien un subconjunto de provincias, o bien un subconjunto de municipios de una misma provincia. No tuvimos en cuenta mayor granularidad.
- El alcance de la historia que almacena nuestro sistema, mencionada en la sección inicial, fue determinado luego de un proceso iterativo de elicitación, y aceptado por el cliente.
- Las entidades `OpcionElectoral` y `CandidatoElectoral` se consideraron entidades débiles debido a que cada votación tendrá sus opciones y sus candidatos, y no tiene sentido hablar de opciones o candidatos cuando no existe una elección que los contemple.
- La entidad `MesaElectoral` representa la mesa física perteneciente a un centro de votación. A pesar de ser parte de este centro, el número de mesa es único en todo el sistema dado que actualmente las mesas de cualquier votación tienen un identificador independiente del centro.

## 5. Diseño Físico

Elegimos el motor de base de datos MySQL dado que la mayoría de los integrantes del grupo lo conocían y utilizaban previamente.

### 5.1. Esquema

```
CREATE DATABASE DB_GRUPO4.TP1
USE DB_GRUPO4.TP1
```

### 5.2. Tablas

```
CREATE TABLE Provincia (
  idProvincia bigint NOT NULL AUTO_INCREMENT,
  nombre char(40),
  CONSTRAINT PK_Provincia PRIMARY KEY(idProvincia)
)
```

```
CREATE TABLE Municipio (
  idMunicipio bigint NOT NULL AUTO_INCREMENT,
  nombre char(40),
  idProvincia bigint,
  CONSTRAINT PK_Municipio PRIMARY KEY(idMunicipio),
  CONSTRAINT FK_ProvinciaXMunicipio FOREIGN KEY (idProvincia) REFERENCES Provincia (id-
  Provincia)
)
```

```
CREATE TABLE CentroDeVotacion (
  idCentro bigint NOT NULL AUTO_INCREMENT,
  direccion char(40),
  nombre char(40),
  idMunicipio bigint,
  CONSTRAINT PK_CentroDeVotacion PRIMARY KEY(idCentro),
  CONSTRAINT FK_MunicipioXCentroDeVotacion FOREIGN KEY (idMunicipio) REFERENCES Mu-
  nicipio (idMunicipio)
)
```

```
CREATE TABLE Ciudadano (
  DNI bigint NOT NULL,
  Nombre char(40),
  Apellido char(20),
  FechaNacimiento datetime,
  FechaIngresoPadron datetime,
  FechaDefuncion datetime,
  NumeroMesaPadron int,
  CONSTRAINT PK_Ciudadano PRIMARY KEY(DNI)
)
```

```
CREATE TABLE MesaElectoral (
  NumeroMesa int NOT NULL AUTO_INCREMENT,
  DNITecnico bigint,
  idCentro bigint,
  CONSTRAINT PK_MesaElectoral PRIMARY KEY(NumeroMesa),
  CONSTRAINT FK_CiudadanoXMesaElectoral FOREIGN KEY (DNITecnico) REFERENCES Ciuda-
  dano (DNI),
)
```

```
CONSTRAINT FK_CentroDeVotacionXMesaElectoral FOREIGN KEY (idCentro) REFERENCES CentroDeVotacion (idCentro)
)
```

```
ALTER TABLE Ciudadano ADD FOREIGN KEY (NumeroMesaPadron) REFERENCES MesaElectoral (NumeroMesa)
```

```
CREATE TABLE TipoVotacion (
idTipoVotacion bigint NOT NULL AUTO_INCREMENT,
Nombre char(40),
CONSTRAINT PK_TipoVotacion PRIMARY KEY(idTipoVotacion)
)
```

```
CREATE TABLE Votacion (
idVotacion bigint NOT NULL AUTO_INCREMENT,
fechaInicio datetime,
fechaFin datetime,
tipo char(40),
idTipoVotacion bigint,
CONSTRAINT PK_Votacion PRIMARY KEY(idVotacion),
CONSTRAINT FK_TipoVotacionXVotacion FOREIGN KEY (idTipoVotacion) REFERENCES TipoVotacion (idTipoVotacion)
)
```

```
CREATE TABLE ProvinciaQueAbarcaVotacion (
idProvincia bigint NOT NULL,
idVotacion bigint NOT NULL,
CONSTRAINT PK_ProvinciaQueAbarcaVotacion PRIMARY KEY(idProvincia, idVotacion),
CONSTRAINT FK_ProvinciaXProvinciaQueAbarcaVotacion FOREIGN KEY (idProvincia) REFERENCES Provincia (idProvincia),
CONSTRAINT FK_VotacionXProvinciaQueAbarcaVotacion FOREIGN KEY (idVotacion) REFERENCES Votacion (idVotacion)
);
```

```
CREATE TABLE MunicipioQueAbarcaVotacion ( idMunicipio bigint NOT NULL,
idVotacion bigint NOT NULL,
CONSTRAINT PK_MunicipioQueAbarcaVotacion PRIMARY KEY(idMunicipio, idVotacion),
CONSTRAINT FK_MunicipioXMunicipioQueAbarcaVotacion FOREIGN KEY (idMunicipio) REFERENCES Municipio (idMunicipio),
CONSTRAINT FK_VotacionXMunicipioQueAbarcaVotacion FOREIGN KEY (idVotacion) REFERENCES Votacion (idVotacion)
)
```

```
CREATE TABLE PartidoPolitico (
idPartido bigint NOT NULL AUTO_INCREMENT,
Nombre char(40),
CONSTRAINT PK_PartidoPolitico PRIMARY KEY(idPartido)
)
```

```
CREATE TABLE EleccionCargoPublico (
idVotacion bigint NOT NULL,
CONSTRAINT PK_EleccionCargoPolitico PRIMARY KEY(idVotacion),
CONSTRAINT FK_VotacionXEleccionCargoPolitico FOREIGN KEY (idVotacion) REFERENCES Votacion (idVotacion)
)
```

```
CREATE TABLE ConsultaPopular (  
  idVotacion bigint NOT NULL,  
  CONSTRAINT PK_ConsultaPopular PRIMARY KEY(idVotacion),  
  CONSTRAINT FK_VotacionXConsutlaPopular FOREIGN KEY (idVotacion) REFERENCES Votacion  
  (idVotacion)  
)
```

```
CREATE TABLE CandidatoElectoral (  
  idVotacion bigint NOT NULL,  
  Numero int NOT NULL,  
  CantVotos int DEFAULT 0,  
  DNI bigint,  
  idPartido bigint,  
  CONSTRAINT PK_CandidatoElectoral PRIMARY KEY(idVotacion, Numero),  
  CONSTRAINT FK_PartidoPoliticoXCandidatoElectoral FOREIGN KEY (idPartido) REFERENCES  
  PartidoPolitico (idPartido),  
  CONSTRAINT FK_EleccionCargoPublicoXCandidatoElectoral FOREIGN KEY (idVotacion) REFE-  
  RENCES EleccionCargoPublico (idVotacion)  
)
```

```
CREATE TABLE OpcionElectoral (  
  idVotacion bigint NOT NULL,  
  Numero int NOT NULL,  
  descripcion char(40),  
  cantVotos int DEFAULT 0,  
  CONSTRAINT PK_OpcionElectoral PRIMARY KEY(idVotacion, Numero),  
  CONSTRAINT FK_ConsultaPopularXOpcionElectoral FOREIGN KEY (idVotacion) REFERENCES  
  ConsultaPopular (idVotacion)  
)
```

```
CREATE TABLE AutoridadDeMesa (  
  idAutoridad bigint NOT NULL AUTO_INCREMENT,  
  tipo char(40),  
  DNI bigint,  
  NumeroMesa int,  
  idVotacion bigint,  
  CONSTRAINT PK_AutoridadDeMesa PRIMARY KEY(idAutoridad),  
  CONSTRAINT FK_MesaElectoralXAutoridadDeMesa FOREIGN KEY (NumeroMesa) REFERENCES  
  MesaElectoral (NumeroMesa),  
  CONSTRAINT FK_VotacionXAutoridadDeMesa FOREIGN KEY (idVotacion) REFERENCES Vota-  
  cion (idVotacion),  
  CONSTRAINT FK_CiudadanoXAutoridadDeMesa FOREIGN KEY (DNI) REFERENCES Ciudadano  
  (DNI)  
)
```

```
CREATE TABLE FiscalDeMesa (  
  idAutoridad bigint NOT NULL,  
  idPartido bigint NOT NULL,  
  CONSTRAINT PK_FiscalDeMesaXFiscalDeMesa PRIMARY KEY(idAutoridad, idPartido),  
  CONSTRAINT FK_AutoridadDeMesaXFiscalDeMesa FOREIGN KEY (idAutoridad) REFERENCES  
  AutoridadDeMesa (idAutoridad),  
  CONSTRAINT FK_PartidoPoliticoXFiscalDeMesa FOREIGN KEY (idPartido) REFERENCES Parti-  
  doPolitico (idPartido)  
)
```

```
CREATE TABLE Camioneta (  

```

```
idCamioneta bigint NOT NULL AUTO_INCREMENT,  
patente char(7),  
DNIResponsable bigint,  
CONSTRAINT PK_Camioneta PRIMARY KEY(idCamioneta),  
CONSTRAINT FK_CiudadanoXCamioneta FOREIGN KEY (DNIResponsable) REFERENCES Ciudadano (DNI)  
)
```

```
CREATE TABLE Camioneta_X_CentroDeVotacion (  
idCentro bigint NOT NULL,  
idCamioneta bigint NOT NULL,  
CONSTRAINT PK_Camioneta_X_CentroDeVotacion PRIMARY KEY(idCentro, idCamioneta),  
CONSTRAINT FK_CentroDeVotacionXCamioneta_X_CentroDeVotacion FOREIGN KEY (idCentro) REFERENCES CentroDeVotacion (idCentro),  
CONSTRAINT FK_CamionetaXCamioneta_X_CentroDeVotacion FOREIGN KEY (idCamioneta) REFERENCES Camioneta (idCamioneta)  
)
```

```
CREATE TABLE VotoEn (  
DNI bigint NOT NULL,  
idVotacion bigint NOT NULL,  
NumeroMesa int,  
fecha datetime,  
CONSTRAINT PK_VotoEn PRIMARY KEY(DNI, idVotacion),  
CONSTRAINT FK_CiudadanoXVotoEn FOREIGN KEY (DNI) REFERENCES Ciudadano (DNI),  
CONSTRAINT FK_VotacionXVotoEn FOREIGN KEY (idVotacion) REFERENCES Votacion (idVotacion),  
CONSTRAINT FK_MesaElectoralXVotoEn FOREIGN KEY (NumeroMesa) REFERENCES MesaElectoral (NumeroMesa)  
)
```

```
CREATE TABLE CandidatoEsVotado (  
idVotacion bigint NOT NULL,  
NumeroCandidato int NOT NULL,  
NumeroMesa int NOT NULL,  
cantVotos int,  
CONSTRAINT PK_EsVotadoCandidato PRIMARY KEY(NumeroMesa, idVotacion, NumeroCandidato),  
CONSTRAINT FK_MesaElectoralXCandidatoEsVotado FOREIGN KEY (NumeroMesa) REFERENCES MesaElectoral (NumeroMesa),  
CONSTRAINT FK_CandidatoElectoralXCandidatoEsVotado FOREIGN KEY (idVotacion, NumeroCandidato) REFERENCES CandidatoElectoral (idVotacion, Numero)  
)
```

```
CREATE TABLE OpcionElectoralEsVotada (  
idVotacion bigint NOT NULL,  
NumeroOpcion int NOT NULL,  
NumeroMesa int NOT NULL,  
cantVotos int,  
CONSTRAINT PK_EsVotadaOpcion PRIMARY KEY(NumeroMesa, idVotacion, NumeroOpcion),  
CONSTRAINT FK_MesaElectoralXOpcionElectoralEsVotada FOREIGN KEY (NumeroMesa) REFERENCES MesaElectoral (NumeroMesa),  
CONSTRAINT FK_OpcionElectoralXOpcionElectoralEsVotada FOREIGN KEY (idVotacion, NumeroOpcion) REFERENCES OpcionElectoral (idVotacion, Numero)  
)
```

## 6. Funcionalidades Implementadas

- DELIMITER \$\$  
 CREATE PROCEDURE GanadoresDelÚltimoAño  
 BEGIN  
 select ce1.Numero  
 from CandidatoElectoral ce1  
 where  
 ((ce1.CantVotos, ce1.idVotacion) in (  
 select max(ce2.CantVotos), ce2.idVotacion  
 from CandidatoElectoral ce2  
 group by idVotacion))  
 and  
 (ce1.idVotacion =  
 (select v.idVotacion  
 from Votacion v  
 where  
 year(v.fechaInicio) = year(now()) and  
 v.fechaFin < now()));  
 END \$\$
  
- DELIMITER \$\$  
 CREATE PROCEDURE ultimos5votantesPorCentroPorVotacion  
 (IN idVot bigint)  
 BEGIN  
 SELECT DISTINCT ve\_final.DNI, me\_final.idCentro  
 FROM VotoEn ve\_final  
 JOIN MesaElectoral me\_final ON ve\_final.NumeroMesa = me\_final.NumeroMesa  
 WHERE ve\_final.idVotacion = idVot AND ve\_final.DNI NOT IN(  
 SELECT DISTINCT ve1.DNI  
 FROM VotoEn ve1  
 JOIN VotoEn ve2 ON ve1.fecha < ve2.fecha AND ve1.idVotacion = ve2.idvotacion  
 JOIN VotoEn ve3 ON ve2.fecha < ve3.fecha AND ve1.idVotacion = ve3.idvotacion  
 JOIN VotoEn ve4 ON ve3.fecha < ve4.fecha AND ve1.idVotacion = ve4.idvotacion  
 JOIN VotoEn ve5 ON ve4.fecha < ve5.fecha AND ve1.idVotacion = ve5.idvotacion  
 JOIN VotoEn ve6 ON ve5.fecha < ve6.fecha AND ve1.idVotacion = ve6.idvotacion  
 JOIN MesaElectoral me1 ON ve1.NumeroMesa = me1.NumeroMesa  
 JOIN MesaElectoral me2 ON ve2.NumeroMesa = me2.NumeroMesa AND me2.idCentro = me1.idCentro  
 JOIN MesaElectoral me3 ON ve3.NumeroMesa = me3.NumeroMesa AND me3.idCentro = me1.idCentro  
 JOIN MesaElectoral me4 ON ve4.NumeroMesa = me4.NumeroMesa AND me4.idCentro = me1.idCentro  
 JOIN MesaElectoral me5 ON ve5.NumeroMesa = me5.NumeroMesa AND me5.idCentro = me1.idCentro  
 JOIN MesaElectoral me6 ON ve6.NumeroMesa = me6.NumeroMesa AND me6.idCentro = me1.idCentro  
 WHERE ve1.idVotacion = idVot  
 );  
 END \$\$
  
- DELIMITER \$\$  
 CREATE PROCEDURE partidosPolíticosConMásDel20 %  
 BEGIN  
 SELECT vo.idVotacion, ce.DNI, sum(ce.CantVotos)  
 FROM Votacion vo  
 JOIN CandidatoElectoral ce ON vo.idVotacion = ce.idVotacion  
 JOIN ProvinciaQueAbarcaVotacion provXvot ON provXvot.idVotacion = vo.idVotacion  
 JOIN TipoVotacion tp ON vo.idTipoVotacion = tp.idTipoVotacion  
 WHERE tp.Nombre = 'Eleccion a Gobernador' AND vo.idVotacion not in (  
 SELECT v1.idVotacion  
 FROM  
 Votacion v1,



```
Votacion v2,
Votacion v3,
Votacion v4,
Votacion v5,
Votacion v6,
TipoVotacion tv
WHERE v1.idVotacion < v2.idVotacion
AND v2.idVotacion < v3.idVotacion
AND v3.idVotacion < v4.idVotacion
AND v4.idVotacion < v5.idVotacion
AND v5.idVotacion < v6.idVotacion
AND tv.Nombre='Eleccion a Gobernador'
AND v1.idTipoVotacion = tv.idTipoVotacion
AND v2.idTipoVotacion = tv.idTipoVotacion
AND v3.idTipoVotacion = tv.idTipoVotacion
AND v4.idTipoVotacion = tv.idTipoVotacion
AND v5.idTipoVotacion = tv.idTipoVotacion
AND v6.idTipoVotacion = tv.idTipoVotacion
)
GROUP BY ce.DNI
HAVING (sum(ce.CantVotos)*5) >= (SELECT sum(CantVotos) FROM Votacion v JOIN Candi-
datoElectoral ce2 ON v.idVotacion = ce2.idVotacion WHERE
v.idTipoVotacion = 1) ;
END $$
```

## 7. Conclusiones

El proceso de construcción de una base de datos es un proceso circular que comienza con un modelo de entidad relación, continúa con el pasaje correspondiente a modelo relacional, y sigue con un esquema físico. Esto se repite iterativamente hasta refinar el modelo y contribuir a la eficiencia de las consultas más solicitadas.

Consideramos que la etapa más importante del proceso es la creación del modelo de entidad relación dado que del mismo derivan las demás etapas. Además el DER y el MR permiten documentar el modelo de datos para su uso y modificación a futuro.

El motor elegido MySQL tiene las siguientes desventajas:

- No permite el uso de CHECKs (lo permite sintácticamente pero es ignorado por el motor). Hay que simularlos con TRIGGERS.
- No permite usar LIMIT en una subquery.