

Diseño e implementación de la base de datos para una aplicación de gestión de procesos electorales

Juan Javier Corrales Pérez

Grado de Ingeniería Informática
Bases de datos

Jordi Ferrer Duran
Xavier Baró Solé

10 de junio de 2021



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

© (Juan Javier Corrales Pérez)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Diseño e implementación de la base de datos para una aplicación de gestión de procesos electorales</i>
Nombre del autor:	<i>Juan Javier Corrales Pérez</i>
Nombre del consultor/a:	<i>Jordi Ferrer Duran</i>
Nombre del PRA:	<i>Xavier Baró Solé</i>
Fecha de entrega (mm/aaaa):	06/2021
Titulación:	<i>Grado de Ingeniería Informática</i>
Área del Trabajo Final:	<i>Bases de datos</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>Data warehouse, Oracle, Elecciones</i>
<p>Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i></p>	
<p>El presente trabajo fin de grado tiene como objetivo diseñar una base de datos que se ocupe de dar respuesta a un mayor control y gestión de los procesos electorales que se llevan a cabo en el país. Así mismo, se desea poder dar respuesta a la implementación del voto telemático en dichos procesos electorales, garantizando un registro seguro de estos. Otro de los objetivos a cubrir consiste en disponer de un almacén de datos, lo que nos permite consultar una serie de estadísticas en tiempo constante 1, ofreciendo un tiempo de respuesta reducido.</p> <p>Se ha empleado la metodología en cascada, lo que ha favorecido el desarrollo e implementación de todas las fases del proyecto así como un mínimo impacto ante cualquier desajuste detectado.</p> <p>Para cumplir con el objetivo del proyecto, se ha realizado el diseño de la base de datos, se han generados los procedimientos almacenados necesarios para realizar los requisitos funcionales descritos y se han generados los correspondientes juegos de pruebas para garantizar el correcto funcionamiento de estos. Además, la base de datos cuenta con un log de registro, que almacena el detalle de todos los procedimientos que se ejecutan, así como su resultado.</p>	

Abstract (in English, 250 words or less):

The objective of this final degree project is to design a database that deals with responding to greater control and management of the electoral processes that are carried out in the country. Likewise, it is desired to be able to respond to the implementation of the telematic vote in said electoral processes, guaranteeing a secure record of these. Another objective to be covered is to have a data warehouse, which allows us to consult a series of statistics in constant time 1, offering a reduced response time.

The cascade methodology has been used, which has favored the development and implementation of all phases of the project as well as a minimal impact on any detected mismatch.

To fulfill the objective of the project, the database design has been carried out, the stored procedures necessary to carry out the functional requirements described have been generated and the corresponding test sets have been generated to guarantee their correct operation. In addition, the database has a registration log, which stores the details of all the procedures that are executed, as well as their results.

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	1
1.3 Enfoque y método seguido.....	2
1.4 Planificación del Trabajo.....	2
1.5 Análisis de riesgos y plan de contingencia.....	4
1.6 Breve resumen de productos obtenidos.....	6
1.7 Breve descripción de los otros capítulos de la memoria.....	6
2. Resto de capítulos.....	8
2.1 Análisis de requisitos.....	8
2.1.1 Requisitos funcionales.....	8
2.1.2 Requisitos no funcionales.....	10
2.2 Diseño conceptual.....	10
2.2.1 Definición de entidades y atributos.....	10
2.2.2 Definición de relaciones.....	14
2.2.3 Restricciones de integridad.....	16
2.2.4 Diagrama conceptual.....	17
2.3 Diseño lógico.....	19
2.3.1 Esquema Lógico UML.....	22
2.3.2 Normalización.....	23
2.4 Diseño físico.....	24
2.4.1 Tablespaces.....	24
2.4.2 Usuarios.....	24
2.4.3 Tablas.....	25
2.4.4 Secuencias.....	30
2.4.5 Índices.....	30
2.4.6 Triggers.....	31
2.5 Diseño Data Warehouse.....	31
2.6 Implementación procedimientos.....	35
2.6.1 Procedimientos paquete PK_ABM.....	35

2.6.2 Procedimientos paquete PK_PRC	48
2.6.3 Procedimientos paquete PK_DWH	55
2.7 Ajustes de diseño	57
2.8 Elaboración de pruebas.....	58
2.8.1 Pruebas unitarias Pk_ABM	58
2.8.2 Pruebas unitarias Pk_PRC	64
2.8.3 Pruebas unitarias Pk_DHW	67
3. Conclusiones.....	69
3.1 Seguimiento de la planificación	69
3.1.1 PEC 1 – Plan de trabajo.....	69
3.1.2 PEC 2.....	69
3.1.3 PEC 3.....	69
3.1.4 PEC 4 – Entrega Final	70
3.2 Conclusiones.....	71
3.3 Acciones futuras.....	71
4. Glosario	72
5. Bibliografía	73

Lista de figuras

Ilustración 1: Metodología del proyecto.....	2
Ilustración 2: Diagrama de Gantt.....	4
Ilustración 3: Diagrama Conceptual	18
Ilustración 4: Diagrama lógico	22
Ilustración 5: Diagrama de Gantt ajustado	70
Tabla 1: Evaluación de riesgos	5
Tabla 2: Plan de contingencias	6
Tabla 3: Requisitos funcionales.....	10
Tabla 4: Requisitos no funcionales.....	10
Tabla 5: Definición de entidades y atributos.....	13
Tabla 6: Definición de entidades y atributos auxiliares.....	14
Tabla 7: Definición de relaciones	15
Tabla 8: Definición de tablespaces.....	24
Tabla 9: Definición de usuarios	25
Tabla 10: Definición de tablas	29
Tabla 11: Definición de tablas auxiliares	30
Tabla 12: Definición de secuencias.....	30
Tabla 13: Definición de índices	31
Tabla 14: Definición de triggers.....	31
Tabla 15: Definición de tablas DWH.....	34
Tabla 16: Definición de Índices DWH.....	34
Tabla 17: Definición de Secuencias DWH.....	34
Tabla 18: Definición de Triggers DWH	35
Tabla 19: Definición de paquetes de procedimientos.....	35
Tabla 20: Definición de los procedimientos del paquete PK_ABM.....	48
Tabla 21: Definición de los procedimientos del paquete PK_PRC	55
Tabla 22: Definición de los procedimientos del paquete PK_DWH	57
Tabla 23: Definición de pruebas unitarias pk_ABM.....	64
Tabla 24: Definición de pruebas unitarias pk_DWH	68
Tabla 25: Definición de pruebas unitarias pk_PRC	Error!
Marcador	no
definido.	

1. Introducción

1.1 Contexto y justificación del Trabajo

El objetivo de este proyecto de trabajo fin de grado pretende cubrir las necesidades marcadas por el departamento de organización administrativa del gobierno de nuestro país para la implementación de una base de datos para la gestión de procesos electorales. Este tendrá que cumplir con los requisitos marcados en la documentación aportada al comienzo del proyecto.

El departamento de organización administrativa necesita incrementar el control sobre los diferentes procesos electorales de manera que puedan garantizar su eficiencia y transparencia. Además de esto, otro de los objetivos que pretenden implementar es el voto telemático para adaptar estos procesos a la realidad tecnológica en la que nos encontramos.

Será muy importante generar mecanismos de seguridad en el acceso a los datos, dado que toda la información que se manejará es muy sensible y se debe de garantizar que el acceso se llevará a cabo cumpliendo todos los protocolos establecidos para ello.

Teniendo en cuenta que el volumen de datos que se va a manejar será bastante grande, será necesario la implementación y uso de almacenes de datos que nos permitan realizar consultas estadísticas de manera eficiente en tiempos de espera

En resumen, se debe de poder realizar un diseño ajustado a las necesidades concretas que han sido marcadas en el proyecto y debidamente documentada de manera que proporcione una base fiable para el desarrollo de la aplicación que deberá gestionar la base de datos.

1.2 Objetivos del Trabajo

El objetivo de este trabajo es la creación de una base de datos que se ocupe del control de los procesos electorales que se llevan a cabo en el país. Esta base de datos debe de ser escalable y debe poder permitir la integración con otras bases de datos accesibles desde el gobierno.

Toda la gestión y acceso a la información deberá realizarse mediante procedimientos de base de datos, ya que será la única manera de acceder a los datos.

El modelo debe de garantizar las normas de seguridad marcadas por el GDPR para garantizar la privacidad de los datos que se encuentren almacenados en el sistema.

El modelo deberá poder gestionar cualquier volumen de datos, por lo que se deberán de aplicar las técnicas de gestión de almacenes de datos con la restricción de que estos tendrán que ser obtenidos en tiempo constante 1.

El modelo también deberá de poder monitorizar las acciones que se realizan sobre este mediante logs, de manera que se pueda registrar los procedimientos ejecutados y también se deberá de dotar de mecanismos que nos permitan testear la funcionalidad de la base de datos.

1.3 Enfoque y método seguido

El proyecto se llevará a cabo siguiendo la metodología en cascada, debido a que conocemos las especificaciones del proyecto de antemano y estas no van a cambiar, además de tener definido un objetivo claro a desarrollar.

Este tipo de metodología nos permite la realización del proyecto mediante etapas que se irán retroalimentando cada una de ellas de las anteriores, lo que nos permitirá poder corregir los posibles errores que vayan surgiendo antes de avanzar al resto de etapas. Las etapas que se llevarán a cabo para abordar el proyecto son Análisis, Diseño, Implementación, Pruebas y Entrega Final.

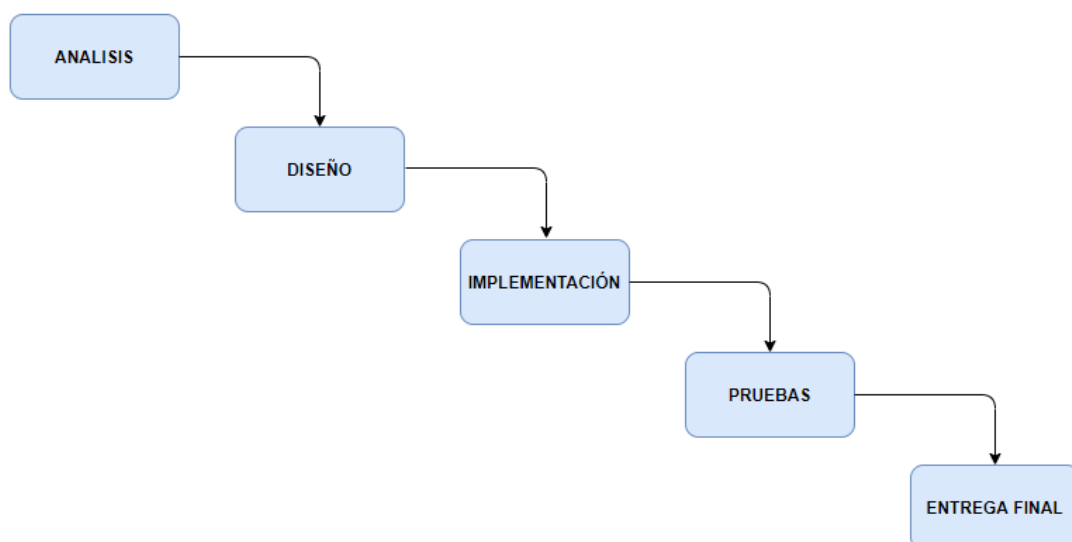


Ilustración 1: Metodología del proyecto

1.4 Planificación del Trabajo

Para la realización de este trabajo se va a emplear el tiempo y trabajo del estudiante con las indicaciones y sugerencias marcadas por el consultor. Como medios técnicos y materiales se empleará el hardware propio del estudiante y como software el disponible entre los recursos docentes.

El tiempo que ha sido definido para el proyecto considera un esfuerzo diario de 3 horas por día de lunes a sábado, lo que suponen 18 horas semanales. Ante cualquier imprevisto en el proyecto o necesidad de mayor dedicación a este se utilizará tanto el sábado como el domingo para poder afrontar esa demanda.

Los hitos parciales que han sido marcados para cada una de las entregas de las PEC son las siguientes:

PEC 1 – Plan de Trabajo

Comienzo: 18/02/21

Fin: 08/03/21

PEC 2

Comienzo: 09/03/21

Fin: 12/04/21

Tareas: Análisis de requisitos, Diseño conceptual, Diseño lógico

PEC 3

Comienzo: 13/04/21

Fin: 12/05/21

Tareas: Diseño físico, Diseño Data Warehouse, Implementación procedimientos.

Entrega Final

Comienzo: 13/05/21

Fin: 10/06/21

Tareas: Elaboración de pruebas, Depuración de errores, Elaboración documentación.

A continuación, se adjunta el diagrama de Gantt con la planificación definida para la duración completa del proyecto.

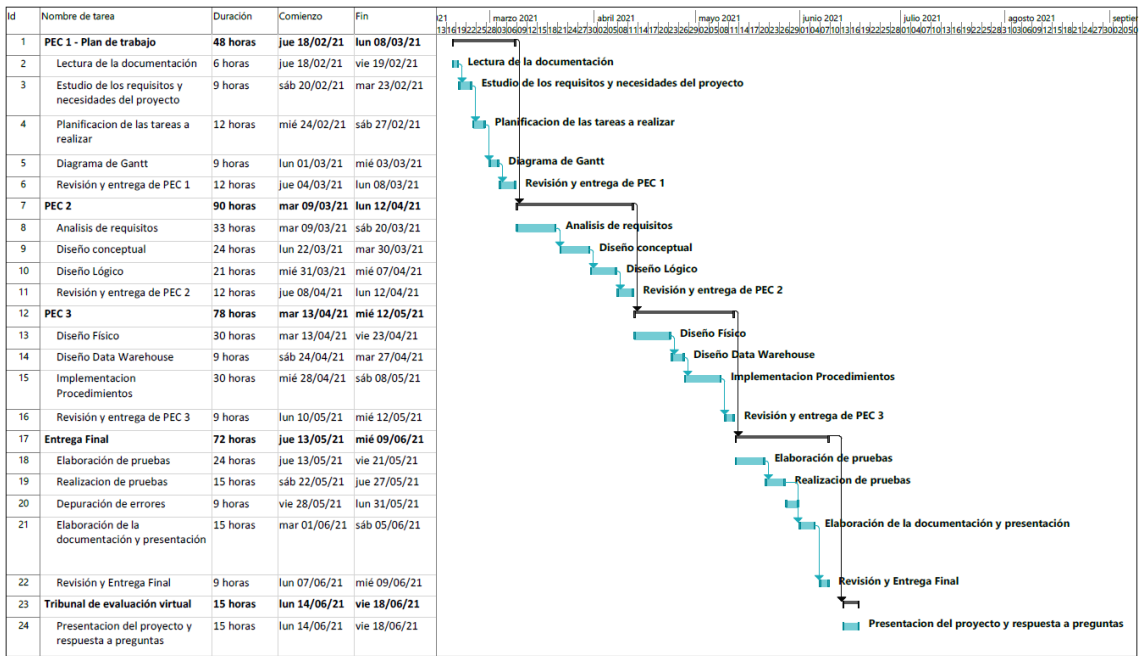


Ilustración 2: Diagrama de Gantt

1.5 Análisis de riesgos y plan de contingencia

En cualquier proyecto estamos expuestos a que puedan surgir cualquier tipo de imprevistos, por lo que deberemos de medir el impacto que estas pueden afectar en el proyecto y las medidas que tomaremos para reducir dicho impacto lo menos posible. Los riesgos más significativos identificados son:

Evaluación de riesgos				
Código	Nombre	Descripción	Impacto	
R01	Problemas de fuerza mayor	El estudiante adolece alguna enfermedad o tiene un pico de trabajo en su empresa.	Alto	
R02	Distorsión en planificación	La planificación definida inicialmente no cumple la realidad del desarrollo del proyecto.	Alto	
R03	Fallo de Hardware	La máquina donde se está desarrollando el proyecto queda inutilizada.	Bajo	
R04	Fallo de Software	Las herramientas que se utilizan para el desarrollo del proyecto presentan alguna anomalía.	Medio	
R05	Incomunicación con consultor	Perdida de la comunicación o respuesta con el consultor del proyecto.	Medio	
R06	Perdida de datos	Un problema en el equipo o un borrado accidental ocasionan una pérdida de datos del proyecto.	Medio	
R07	Perdida de conectividad	No se dispone de conexión a internet o no se tiene acceso a plataforma UOC.	Bajo	

R08	<i>Retraso en la entrega de hitos</i>	<i>No se consigue finalizar el trabajo planificado en los diferentes hitos.</i>	Alto
-----	---------------------------------------	---	-------------

Tabla 1: Evaluación de riesgos

A continuación, definimos el plan de contingencia a aplicar para cada uno de los riesgos identificados:

Plan de Contingencia				
1. Estrategia				
La estrategia que debe aplicarse para mitigar los riesgos del proyecto es proactiva. Se han de plantear aquellas acciones que, en primer lugar, nos permitan evitar la aparición de la incidencia y, en segundo lugar, en caso de que no se pueda evitar un riesgo, establecer los planes de contingencia que nos permitan minimizar su impacto.				
2. Medidas correctoras				
Código	Acción	Tipo	Riesgo Residual	Fecha límite
A1R01	Aumentar las horas de trabajo a las inicialmente planificadas.	Mitigadora	Muy bajo	30/05/21
A2R01	Consensuar con el consultor nuevas fechas de entrega del hito afectado.	Correctora	Medio	10/06/21
A1R02	Replanificación de las diferentes tareas y tiempos a emplear.	Correctora	Medio	12/05/21
A1R03	Adquirir una nueva máquina donde continuar el proyecto.	Correctora	Muy bajo	10/06/21
A1R04	Reinstalación de las herramientas o sustitución por otras de similares características.	Correctora	Muy bajo	10/06/21
A1R05	Establecer contacto con responsable asignatura o tutor para evaluar alternativas.	Correctora	Bajo	10/06/21
A1R06	Realizar copia de seguridad diaria y semanal de manera incremental.	Correctora	Bajo	10/06/21
A2R06	Estudiar la posibilidad de utilizar otro entorno como backup.	Correctora	Bajo	10/06/21
A1R07	Utilización de conexión móvil o de espacio público.	Correctora	Muy bajo	18/06/21
A2R07	Contactar con consultor vía email.	Correctora	Muy bajo	18/06/21

A1R08	Establecer unos mínimos funcionales a entregar en el hito afectado.	Mitigadora	Medio	10/06/21
-------	---	------------	-------	----------

Tabla 2: Plan de contingencias

1.6 Breve resumen de productos obtenidos

A la finalización del proyecto, los productos obtenidos serán los siguientes:

- Memoria: Documentación que contiene un detalle exhaustivo del desarrollo del proyecto, así como los detalles técnicos llevados a cabo para el diseño y la implementación de la base de datos.
- Presentación: Video resumen del proyecto donde se presentan las virtudes de la solución desarrollada.
- Autoinforme de evaluación: Registro de la evaluación de las competencias transversales.
- Producto: Este contendrá los scripts necesarios para la creación de la base de datos, el juego de datos inicial para la realización de las pruebas y los correspondientes scripts para la comprobación de los requerimientos que deben de cumplirse.

1.7 Breve descripción de los otros capítulos de la memoria

El capítulo análisis de requisitos se ocupará de recoger los requerimientos funcionales del proyecto que han de ser cumplidos además de los requerimientos no funcionales, sobre los que basaremos el diseño de la base de datos.

El capítulo del diseño conceptual se basará en identificar las entidades, relaciones y atributos que conformarán nuestra base de datos, así como crear el diagrama de entidad-relación.

El capítulo del diseño lógico se encargará de transformar el diseño conceptual definido en el apartado anterior en el modelo relacional que utilizaremos. También nos ocuparemos de realizar la normalización de esta para evitar futuros problemas.

El capítulo del diseño físico se ocupará de transformar el diseño lógico definido en un modelo que nos permita implementar sobre el sistema gestor de bases de datos que vayamos a utilizar.

El capítulo del diseño del DATA WAREHOUSE¹ se centrará en cubrir las necesidades definidas en el proyecto para el manejo de grandes volúmenes de datos y dar cabida al repositorio estadístico que manejará las consultas definidas a tal efecto.

El capítulo de implementación de procedimientos describe los procedimientos almacenados o funciones que van a ser incluidas en la base de datos para cumplir con la funcionalidad requerida en el proyecto.

El capítulo de la fase de pruebas se encargará de la definición de un plan de pruebas para la comprobación del correcto funcionamiento, así como la ejecución de dicho plan.

El resto de los capítulos, conclusiones, glosario, bibliografía y anexos contendrán las conclusiones del proyecto, el glosario de términos utilizados, la bibliografía utilizada en la elaboración de todo el proyecto y los anexos en caso de que sean necesarios respectivamente.

¹ *Almacén de datos* - *Wikipedia, la enciclopedia libre*. (s. f.). Recuperado 24 de febrero de 2021, de https://es.wikipedia.org/wiki/Almacén_de_datos

2. Resto de capítulos

2.1 Análisis de requisitos

En esta fase del proyecto nos ocuparemos de conocer y analizar las necesidades y objetivos que el enunciado de nuestro TFG describe, de manera que podamos cumplir con los objetivos marcados en estos. Para ello identificaremos tanto los requisitos funcionales como los requisitos no funcionales que deberá de cumplir el diseño de nuestra base de datos.

2.1.1 Requisitos funcionales

A continuación, enumeraremos la relación de funcionalidades necesarias para dar respuesta a las necesidades planteadas y que se desea almacenar en nuestra BD.

Código	Requisito
RF-01	Se desea poder almacenar la relación de personas existentes en el padrón municipal de los diferentes pueblos y ciudades del país.
RF-02	Almacenar los datos correspondientes a las mesas y locales electorales disponibles en cada uno de los municipios donde se lleve a cabo un proceso electoral.
RF-03	Registrar la publicación de las diferentes convocatorias de elecciones, así como las reglas que las rigen.
RF-04	Elaboración del censo electoral que podrá participar en cada una de las convocatorias publicadas según las características definidas en la convocatoria.
RF-05	Se gestionará el registro de partidos políticos según las reglas definidas para cada tipo de elecciones.
RF-06	Se llevará a cabo la gestión de los candidatos que se presentan a cada demarcación y se publicará la lista definitiva de candidatos por parte del gobierno.
RF-07	Gestionar las demarcaciones electorales para cada tipo de proceso electoral que se lleve a cabo con su número de escaños.
RF-08	Elaborar la composición de los miembros de las mesas electorales, que estarán formadas por 3 miembros + 3 suplentes.
RF-09	Se deberá poder almacenar las restricciones a considerar para formar parte de una mesa electoral.
RF-10	Se registrarán los votos obtenidos por cada partido político en cada una de las mesas electorales, tanto presencial como telemáticamente.
RF-11	El sistema debe de poder almacenar diferentes métodos de cálculo de resultados de cada proceso electoral, debiendo de implementarse como mínimo el método D'Hondt ² .

² *d'Hondt*. (s. f.). Recuperado 18 de marzo de 2021, de <https://alarcos.esi.uclm.es/per/fruiz/espaelec/ley/dhondt/dhondt.htm>

RF-12	Calcular los resultados de cada demarcación utilizando el método de cálculo definido y almacenar su resultado.
RF-13	Se mantendrá un registro de los usuarios que acceden a los datos. Este deberá almacenar el usuario que accede, la fecha y hora de acceso y la lista de registros consultados.
RF-14	<p>Se creará un almacén de datos estadísticos para dar respuesta a las siguientes consultas:</p> <ul style="list-style-type: none"> • De todos los datos existentes en el sistema, el proceso electoral en el que ha habido más participación. • Teniendo en cuenta todos los procesos electorales ya celebrados, municipio donde ha habido la mayor participación. • Dado un año cualquiera, porcentaje de votos telemáticos respecto al total de votos emitidos en todos los procesos electorales celebrados en ese año. • Porcentaje de incremento (o decremento) del total de votos del último proceso electoral celebrado respecte al inmediatamente anterior. • Teniendo en cuenta todos los datos de que se dispone, número total de personas que han formado parte más de 2 veces de una mesa electoral. • En un momento cualquiera, porcentaje de personas que han estado seleccionadas para participar en una mesa electoral en las próximas elecciones y que han presentado alegaciones para no formar parte de ella. • Teniendo en cuenta los últimos 10 años, número de partidos políticos que se han presentado en todas las elecciones celebradas en todas las demarcaciones electorales de cada proceso. • En un momento cualquiera, porcentaje de municipios que no han realizado todavía el sorteo de asignación de miembros de las mesas electorales. • Analizando todos los procesos electorales ya celebrados, Top20 de mesas electorales que han recibido menos votos (presenciales o telemáticos). Se deberá poder obtener las 20 mesas electorales con menos votos sea cual sea el proceso electoral. • Teniendo en cuenta el último proceso electoral comunicado por el Gobierno, porcentaje de personas que deberían formar parte de una mesa y a quien no se ha localizado para entregarle la comunicación de participación. • Teniendo en cuenta todos los datos de que se dispone, proceso electoral donde ha habido más candidatos en total presentados por los partidos políticos. • En las últimas elecciones celebradas, número total de escaños obtenido por cada partido político.
RF-15	Se creará un registro log de las operaciones realizadas con la BD, almacenando las llamadas a los procedimientos del sistema,

	identificando el nombre del procedimiento, los parámetros de entrada y los parámetros de salida.
--	--

Tabla 3: Requisitos funcionales

2.1.2 Requisitos no funcionales

Código	Requisito
RNF-01	La base de datos será relacional.
RNF-02	Se utilizará el motor de base de datos ORACLE ³ .
RNF-03	Toda la gestión y acceso a la información se hará mediante procedimientos de BD. Se implementarán los procedimientos de alta, baja y modificación de todas las entidades relevantes.
RNF-04	Los indicadores definidos en los requisitos funcionales (RF-014) deberán poder obtenerse en tiempo constante 1.
RNF-05	La base de datos tendrá un diseño escalable.
RNF-06	Incluir mecanismos de control para resolver potenciales problemas de integración y testear la funcionalidad de la base de datos.
RNF-07	El sistema almacenará información detallada de las llamadas a procedimientos de la base de datos mediante logs. RNF-03.01 Deberán disponer de un parámetro de salida RSP de tipo String que indicará si se ha finalizado correctamente ('OK') o si ha fracasado ('ERROR+TIPO ERROR') RNF-03.02 Se dispondrá de tratamiento de excepciones.

Tabla 4: Requisitos no funcionales

2.2 Diseño conceptual

El diseño conceptual lo que pretende es identificar las entidades que se encuentran involucradas en nuestro diseño, así como las relaciones existentes entre ellas y las restricciones de integridad que tienen. Dicho diseño nos deberá de permitir cumplir con todos los requisitos que han sido definidos. La metodología que emplearemos será la centralizada (one shot) con un único esquema conceptual, dado que no tenemos un número alto de requisitos que requieran emplear un modelo por áreas.

2.2.1 Definición de entidades y atributos

A continuación, detallamos la relación de entidades que hemos identificado, así como sus correspondientes atributos.

³ Oracle Corporation - Wikipedia, la enciclopedia libre. (s. f.). Recuperado 20 de marzo de 2021, de https://es.wikipedia.org/wiki/Oracle_Corporation

Entidad	Descripción	Atributos
Padron_Municipal	Almacena el conjunto de personas que se encuentran registradas como habitantes de un determinado municipio y que van a ser integrados desde una BBDD externa del gobierno. De esta tabla saldrá el censo electoral de la convocatoria.	<ul style="list-style-type: none"> • Código de padrón • Nombre de ciudadano • Primer apellido del ciudadano • Segundo apellido del ciudadano • Numero de documento del ciudadano • Fecha de nacimiento del ciudadano • Provincia de nacimiento del ciudadano • Municipio de nacimiento del ciudadano • País de nacimiento del ciudadano • Sexo del ciudadano • Provincia de la vivienda habitual del ciudadano • Municipio de la vivienda habitual del ciudadano • Dirección de la vivienda habitual del ciudadano
Convocatoria	Registra las convocatorias de elecciones que son publicadas desde el gobierno con sus diferentes características.	<ul style="list-style-type: none"> • Código de la Convocatoria • Descripción de la convocatoria • Fecha de la publicación de la convocatoria • Fecha de inicio de la votación de la convocatoria • Fecha de fin de la votación de la convocatoria • Ámbito geográfico de la convocatoria • Código de la autonomía • Número de escaños por demarcaciones de la convocatoria • Estado de la convocatoria
Censo_Electoral	Tendrá los datos de todas las personas con derecho a voto en una convocatoria. También indicará si dicha persona será miembro de una mesa electoral.	<ul style="list-style-type: none"> • Código de la convocatoria • Código del padrón municipal • Fecha de inscripción en el censo electoral • Booleano que indica si ha sido seleccionado

		<p>para formar parte de una mesa electoral</p> <ul style="list-style-type: none"> • Indicador de si ha sido seleccionado como titular o suplente de la mesa • Booleano que indica si ha sido recibida la notificación de la selección de la participación en la mesa • Booleano que indica si ha presentado una alegación para no formar parte de la mesa • Booleano que indica si el miembro de la mesa se ha presentado
Candidatura	Contendrá la relación de candidatos que participan en una demarcación a través de un partido político.	<ul style="list-style-type: none"> • Código de la demarcación • Código del partido político • Estado de la candidatura
Mesa	Contendrá los datos de las diferentes mesas electorales disponibles.	<ul style="list-style-type: none"> • Código de la mesa electoral • Descripción de la mesa electoral • Estado de la mesa
Centro_Votacion	Relación de centros de votación disponibles.	<ul style="list-style-type: none"> • Código de la demarcación • Código del municipio • Código de distrito • Código de sección • Dirección del centro de votación • Código postal del centro de votación • Booleano que indica si se ha realizado el sorteo de los miembros de la mesa • Booleano que indica si se han realizado las notificaciones a los seleccionados para las mesas.
Demarcacion	Esta entidad contendrá la lista de demarcaciones disponibles para una convocatoria de elecciones.	<ul style="list-style-type: none"> • Código de la demarcación • Número de escaños de la demarcación • Número de votantes en la demarcación
Resultados	Almacenará los resultados obtenidos por los partidos políticos en cada una de las demarcaciones de una convocatoria.	<ul style="list-style-type: none"> • Código de resultado • Código del partido • Escaños obtenidos • Votos Obtenidos

		<ul style="list-style-type: none"> • Porcentaje de votos
Partido_Politico	Relación de partidos políticos inscritos en el registro de partidos del país.	<ul style="list-style-type: none"> • Código del partido • Siglas del partido • Nombre del partido • Dirección física del partido • Fecha de inscripción en el registro.
Recuento	Almacenará el recuento de votos realizado en cada una de las mesas electorales.	<ul style="list-style-type: none"> • Código del partido • Votos obtenidos
Log_Procesos → Log	Se almacenarán el registro de logs de los procesos que se lanzan en la aplicación. Heredan atributos de la entidad genérica Log.	<ul style="list-style-type: none"> • Código del log • Fecha del log • Nombre procedimiento • Parámetros de entrada del procedimiento • Parámetros de salida del procedimiento
Log_Padron → Log	Se almacenarán el registro de logs de los accesos realizados a los datos del padrón municipal. Heredan atributos de la entidad genérica Log.	<ul style="list-style-type: none"> • Código del log • Fecha del log • Usuario aplicación • Consulta realizada
Usuarios	Esta entidad contendrá el registro de los usuarios de la aplicación.	<ul style="list-style-type: none"> • Código de usuario • Clave de usuario • Nivel de permisos • Estado del usuario
Votacion	Esta entidad contendrá el registro de las votaciones que se realizan en cada una de las mesas electorales.	<ul style="list-style-type: none"> • Tipo de voto realizado • Código del certificado digital

Tabla 5: Definición de entidades y atributos

Además de las entidades descritas anteriormente, también hemos definido una serie de entidades auxiliares que nos servirán de vínculo con las anteriores para realizar nuestro diseño.

Entidad	Descripción	Atributos
Titulacion	Contendrá las diferentes titulaciones académicas que puede poseer un ciudadano.	<ul style="list-style-type: none"> • Código de la titulación • Descripción de la titulación
Comunidad_Autonomas	Tendrá la lista de comunidades autónomas que existen en el país.	<ul style="list-style-type: none"> • Código de la comunidad autónoma • Descripción de la comunidad autónoma
Provincia	Recogerá la lista de las diferentes provincias que existen en el país.	<ul style="list-style-type: none"> • Código de la provincia • Descripción de la provincia
Municipio	Contendrá la lista de los diferentes municipios que existen en el país.	<ul style="list-style-type: none"> • Código del municipio • Descripción del municipio
Parametros	Relación de parámetros que definen una convocatoria. En esta se configurará el método de cálculo del	<ul style="list-style-type: none"> • Código del parámetro • Tipo de parámetro • Campo de filtro del

	escrutinio, las reglas de participación tanto de los partidos políticos como del censo y las restricciones para formar una mesa electoral.	parámetro <ul style="list-style-type: none"> • Valor a emplear
--	--	---

Tabla 6: Definición de entidades y atributos auxiliares

También se han definido unos tipos de datos específicos para algunas entidades con las siguientes agrupaciones:

- **Estado:** (Abierta, Cerrada, Finalizada, Anulada)
- **Prioridad:** (Titular, Suplente)
- **Sexo:** (Hombre, Mujer)
- **TipoConvocatoria:** (Estatal, Autonómica, Municipal)
- **EstadoCandidatura:** (Presentada, Aprobada, Rechazada, Anulada)
- **Voto:** (Presencial, Telemático)
- **TipoUsuario:** (Normal, Avanzado)
- **TipoParametro:** (Partido, Censo, Mesa, Escrutinio)
- **EstadoUsuario:** (Habilitado, Deshabilitado)

2.2.2 Definición de relaciones

En este apartado vamos a identificar las relaciones identificadas entre nuestras entidades y describiremos sus características.

Entidades	Etiquetas	Grado	Conectividad	Descripción
Censo_electoral— Padron_municipal-- Convocatoria	aspirantes confirmados	ternaria	M:N:1	Una convocatoria puede tener uno o varios posibles ciudadanos candidatos a participar en la convocatoria y a su vez esa convocatoria puede tener uno o varios ciudadanos confirmados a participar
Candidatura— Partido_politico-- Demarcacion	tiene se presentan	ternaria	M:N:1	Una demarcación puede tener uno o varios partidos políticos y a su vez, esa demarcación puede tener uno o varias candidaturas a participar.
Centro_votacion— Municipio-- Demarcacion	se compone tiene	ternaria	M:N:1	Una demarcación puede tener uno o varios municipios y a su vez, esa demarcación puede tener uno o varios centros de votación.
Censo_electoral— Candidatura	pertenece	binaria	0..1:N	Una candidatura puede contener varias personas del censo y una persona del censo puede pertenecer a una o ninguna candidaturas.
Convocatoria— Parametros	se define	binaria	M:N	Una convocatoria se define con uno o varios

				parámetros y un parámetro puede estar en varias convocatorias.
Usuarios--- Padron_municipal	consulta	binaria	1:N	Log_padron es una entidad asociada a la relación de consulta de los usuarios del sistema sobre los ciudadanos del padrón.
Padron_municipal— Titulacion	posee	binaria	1:N	Un usuario posee una titulación máxima, pero una titulación pueden tenerla muchos usuarios.
Mesa—Votacion	contiene	binaria	1:N	Una mesa contiene muchos votos, pero un voto solo puede depositarse en una mesa.
Mesa—Recuento	se realiza	binaria	1:N	En una mesa se realiza el recuento de diferentes partidos, pero un recuento pertenece a una mesa.
Centro_Votacion--Mesa	se compone	binaria	1:N	Un centro se compone de muchas mesas, pero una mesa solo puede estar en un centro.
Mesa— Censo_electoral	asignada	binaria	1:N	Una mesa tiene asignadas muchas personas del censo electoral, pero una persona solo tiene asignada una mesa.
Comunidad_autonoma- -Provincia	se compone	binaria	1:N	Una comunidad autónoma se compone de varias provincias, pero una provincia solo pertenece a una comunidad autónoma.
Provincia--Municipio	se compone	binaria	1:N	Una provincia se compone de varios municipios, pero un municipio solo pertenece a una provincia.
Demarcación— Resultados	contabiliza	binaria	1:N	Una demarcación registra los resultados obtenidos en dicha demarcación, pero un resultado solo se registra en una demarcación.
Convocatoria— Demarcación	engloba	binaria	1:N	Una convocatoria engloba a varias demarcaciones, pero una demarcación solo pertenece a una convocatoria.

Tabla 7: Definición de relaciones

2.2.3 Restricciones de integridad

Dadas las entidades identificadas en nuestro diseño, los aspectos más importantes que identificamos en el modelado de nuestro sistema son:

- No se puede borrar una titulación si alguna persona del padrón municipal la tiene asignada.
- La fecha de nacimiento de una persona del padrón municipal no puede ser mayor que la fecha actual.
- La fecha de inicio de una convocatoria debe de ser mayor que la fecha actual.
- La fecha de fin de una convocatoria debe de ser mayor que la fecha actual y mayor que la fecha de inicio.
- No se podrá borrar una mesa si tiene un censo electoral, una votación o un recuento asignado.
- Los datos de comunidad autónoma, provincia, municipio y padrón municipal solo serán accesibles en modo "solo lectura".
- Si se borra un partido político, se borrará en cascada todas las candidaturas si no se ha comenzado el periodo de votación en la demarcación asignada.
- El número de votos de resultados y recuento no podrá ser negativo.
- La suma de votos de resultados no puede ser mayor que el número de personas del censo electoral en esa demarcación.
- La suma de votos de recuento no puede ser mayor que el número de personas del censo electoral en esa mesa.
- El método de cálculo del escrutinio será único por convocatoria.
- Si el atributo 'tipo' de la entidad Convocatoria es Autónoma, el atributo 'autonomia' no puede ser NULL.
- Si el atributo 'esMesa' de la entidad Censo_Electoral es TRUE, el atributo 'tipo' no pueden ser NULL.
- Si el tipo de voto realizado es 'Telemático' el atributo 'certificado' no puede ser NULL.
- Si el estado de la mesa es 'Abierta', el atributo 'presentado' de la entidad Censo_Electoral no puede ser NULL.
- Si el estado de la mesa es 'Finalizada', los atributos 'partido' y 'votos' de la entidad Recuento no pueden ser NULL.
- Si el estado de la convocatoria es 'Finalizada', los atributos 'partido', 'escaños', 'votos' y 'porcentajeVotos' de la entidad Resultados no pueden ser NULL.
- El atributo 'votos' de la entidad Resultados se obtiene a partir de la suma de los votos de la entidad Recuento.
- Si un ciudadano es candidato por un partido político, este no puede formar parte de la mesa electoral.

2.2.4 Diagrama conceptual

En este apartado mostraremos el diagrama conceptual obtenido después de implementar el desarrollo anteriormente expuesto con las diferentes entidades, atributos, relaciones y cardinalidad de estas. Las claves primarias han sido marcadas con el símbolo <P>, mientras que para las claves candidatas hemos usado el símbolo <U>.

La representación de nuestro diseño conceptual se ha realizado utilizando el modelado UML⁴ mediante su correspondiente diagrama estructural.

⁴ *Lenguaje unificado de modelado - Wikipedia, la enciclopedia libre.* (s. f.). Recuperado 5 de junio de 2021, de https://es.wikipedia.org/wiki/Lenguaje_unificado_de_modelado

2.3 Diseño lógico

En esta etapa de lo que nos vamos a ocupar es de transformar nuestro modelo conceptual en un modelo lógico ajustado al modelo relacional en el que se basará nuestro SGBD⁵.

Para representar las relaciones del modelo conceptual emplearemos las siguientes conversiones:

- En las relaciones binarias donde la cardinalidad sea 1:N se añadirá una clave foránea referenciando a la otra relación en la N.
- En las relaciones binarias donde la cardinalidad sea 0..1:N será necesario la creación de una nueva entidad que contendrá como clave primaria la concatenación de las claves foráneas de la relación.
- En las relaciones binarias donde la cardinalidad sea M:N será necesario la creación de una nueva entidad que contendrá como clave primaria la concatenación de las claves foráneas de la relación.
- En las relaciones ternarias donde la cardinalidad sea M:N:1 se añadirán claves foráneas referenciando la relación en la M.

La notación que emplearemos para realizar nuestro diseño lógico será el siguiente:

- Indicaremos las relaciones a partir del nombre, seguido de la lista de atributos entre paréntesis y separados por comas.
- Indicaremos las claves primarias subrayando con una línea continua los atributos que las forman.
- Indicaremos las claves candidatas subrayando con una línea discontinua los atributos que las forman.
- Indicaremos las claves foráneas entre llaves y separadas por comas (si fuera necesario), indicando la tabla sobre la que esta referenciada.
- Utilizaremos el tipo de letra negrita en los nombres de atributo que queremos declarar NOT NULL.

Así, hemos obtenido la siguiente conversión del modelo conceptual:

Convocatoria(idConvocatoria, descripción, fechaPublicacion, fechaInicio, fechaFin, tipo, autonomía, numeroEscanos, estado)

⁵ Sistema de gestión de bases de datos - Wikipedia, la enciclopedia libre. (s. f.). Recuperado 5 de junio de 2021, de https://es.wikipedia.org/wiki/Sistema_de_gestión_de_bases_de_datos

Convocatoria_Parametros(idConvocatoria, idParametro)
{idConvocatoria} referencia CONVOCATORIA
{idParametro} referencia PARAMETROS

Parametros(idParametro, **tipo**, **campo**, **valor**)

Padron_Municipal(idPadron, **nombre**, **apellido1**, **apellido2**, **numDocumento**,
fechaNacimiento, **provinciaNacimiento**, **municipioNacimiento**, **país**, **sexo**,
provinciaDomicilio, **municipioDomicilio**, **direccion**)
{idTitulacion} referencia TITULACION

Titulación(idTitulacion, **descripción**)

Censo_Electoral(idConvocatoria, idPadron, **fechaInscripcion**, **esMesa**, tipo,
notificado, alega, presentado)
{idConvocatoria} referencia CONVOCATORIA
{idPadron} referencia PADRON_MUNICIPAL
{idMesa} referencia MESA

Candidatura(idDemarcacion, idPartido, **estado**)
{idDemarcacion} referencia DEMARCACION
{idPartido} referencia PARTIDO_POLITICO

Candidato(idConvocatoria, idPadron, idDemarcacion, idPartido, **orden**)
{idConvocatoria} referencia CONVOCATORIA
{idPadron} referencia PADRON_MUNICIPAL
{idDemarcacion} referencia DEMARCACION
{idPartido} referencia PARTIDO_POLITICO

Partido_Politico(idPartido, **siglas**, **nombre**, **direccion**, **fechaInscripcion**)

Demarcación(idDemarcacion, **escaños**, **votantes**)
{idConvocatoria} referencia CONVOCATORIA

Resultados(idResultado, **partido**, **escaños**, **votos**, **porcentajeVotos**)
{idDemarcacion} referencia DEMARCACION

Mesa(idMesa, **descripción**, **estado**)
{idDemarcacion} referencia DEMARCACION
{idMunicipio} referencia MUNICIPIO

Votación(**tipo**, certificado)
{idMesa} referencia MESA

Recuento(**partido**, **votos**)
{idMesa} referencia MESA

Centro_Votacion(idDemarcacion, idMunicipio, **distrito**, **sección**, **direccion**,
codigoPostal, **sorteado**, **notificado**)
{idDemarcacion} referencia DEMARCACION

{idMunicipio} referencia MUNICIPIO

Municipio(idMunicipio, **descripcion**)
{idProvincia} referencia PROVINCIA

Provincia(idProvincia, **descripcion**)
{idAutonomia} referencia COMUNIDAD_AUTONOMA

Comunidad_Autonoma(idAutonomia, **descripcion**)

Usuarios(idUsuario, **clave, nivel, estado**)

Log_Padron(idLog, **fecha**)
{idUsuario} referencia USUARIOS
{idPadron} referencia PADRON_MUNICIPAL

Log_Procesos(idLog, **fecha, procedimiento, parametrosEntrada, parametrosSalida**)

2.3.1 Esquema Lógico UML

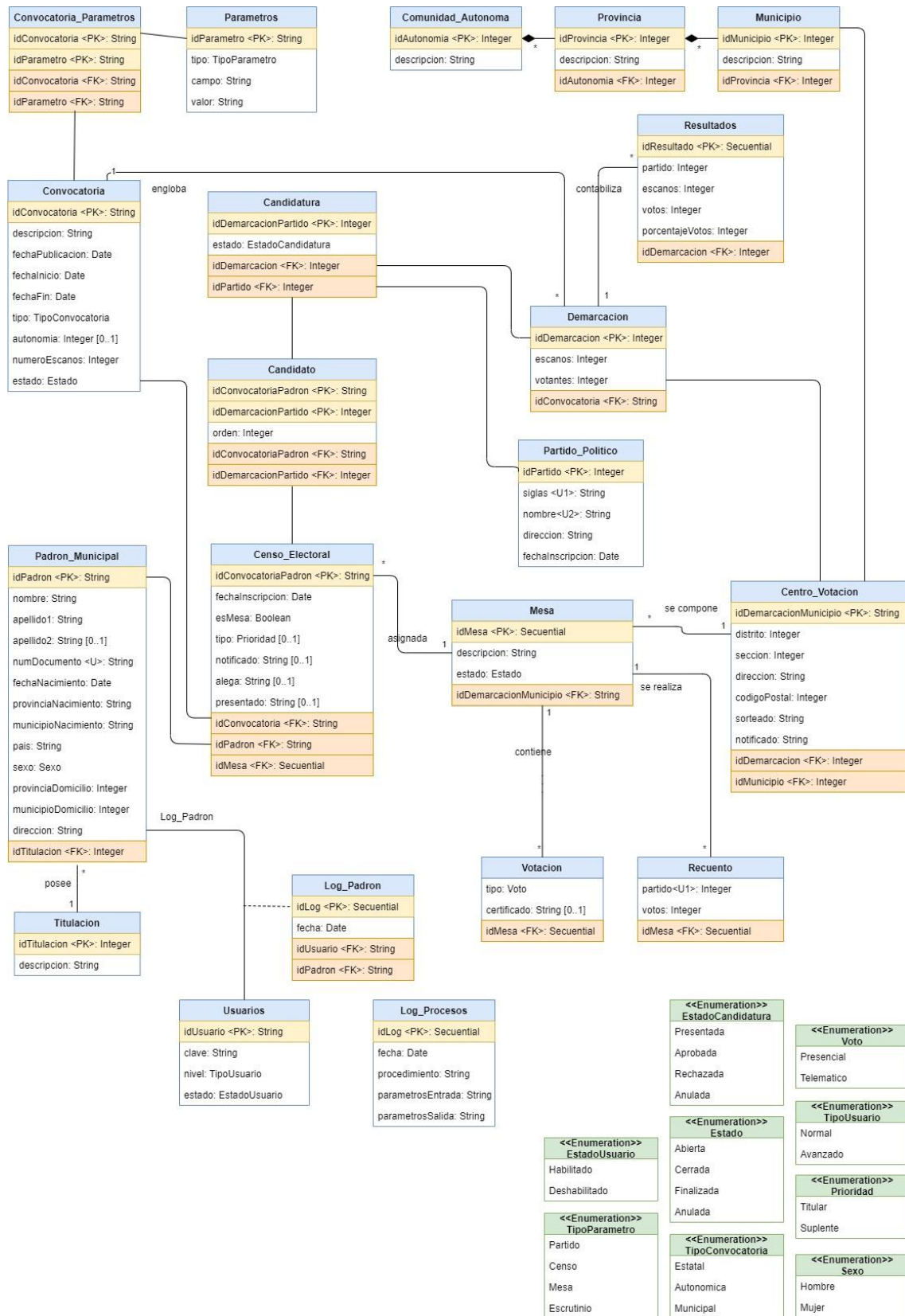


Ilustración 4: Diagrama lógico

2.3.2 Normalización

El objetivo de la teoría de la normalización⁶ es fijar unas condiciones que nos garanticen la separación de conceptos y la ausencia de redundancia para evitar las anomalías de actualización. Dichas anomalías tienen el origen en dependencias existentes entre los atributos de la relación. La teoría de la normalización tiene definido una serie niveles, los cuales nos permiten ir eliminando progresivamente determinadas dependencias que son las que nos producen dichas anomalías. Cuanto más alto es el grado de normalización, menos anomalías se pueden producir.

Para ello pasaremos las diferentes formas normales de las que se compone la teoría de la normalización para ver si nuestro diseño se ajusta a ella y evitamos posibles anomalías.

- **Primera forma normal (1FN):** Comprobamos que los atributos de las relaciones son atómicos y dado que se trata de un modelo relacional, podemos afirmar que nuestro diseño cumple con esta forma.
- **Segunda forma normal (2FN):** Las relaciones candidatas que disponen de un solo atributo y que cumplen **1FN** cumplen directamente con esta forma. En nuestro diseño se ven afectadas las relaciones con dos atributos como clave, estas son Convocatoria_Parametros, Candidatura, Candidato, Censo_Electoral y Centro_Votacion. Todas ellas se relacionan mediante una clave foránea de la parte de la clave compartida, por lo que para este caso también podemos afirmar que se cumple con la forma.
- **Tercera forma normal (3FN):** Una vez comprobado que ningún atributo que no forma parte de una clave candidata depende de un conjunto de atributos que no forma parte de una clave candidata y dado que cumplimos con la **2FN**, podemos afirmar que nuestro diseño cumple con esta forma.
- **Forma normal de Boyce-Codd (FNBC):** dado que en las relaciones con dos o más claves candidatas compuestas que tenemos en nuestro diseño ninguna de ellas tiene un atributo en común y como cumplimos con la forma **3FN**, podemos afirmar que cumplimos con la forma.
- **Cuarta forma normal (4FN):** Dado que nuestro diseño se encuentra en **FNBC** y que no tenemos atributos multivaluados, podemos afirmar que cumplimos con esta forma.

⁶ Normalización de bases de datos - Wikipedia, la enciclopedia libre. (s. f.). Recuperado 27 de marzo de 2021, de https://es.wikipedia.org/wiki/Normalización_de_bases_de_datos

- **Quinta forma normal (5FN):** Nuestro diseño no presenta dependencias de proyección-combinación que puedan producir redundancia en los datos, y dado que cumplimos con la forma **4FN**, podemos afirmar que cumplimos con esta forma también.

En resumen, nuestro diseño cumple con todas las formas normales según el teorema de la normalización, por lo que hemos eliminado todas las anomalías y redundancias que nuestro diseño pudiera tener.

2.4 Diseño físico

En esta etapa de lo que nos vamos a ocupar es de transformar nuestro modelo lógico en un modelo físico ajustado al modelo relacional en el que se basará nuestro SGBD, para el que utilizaremos ORACLE sobre la versión 11G Express Edition.

2.4.1 Tablespaces

El tablespace es la ubicación de almacenamiento que utilizaremos para almacenar los objetos de nuestra base de datos. Con el objetivo de prevenir posibles problemas de pérdida de datos, caída de la base de datos o problemas físicos en el disco, así como para dotar de una mayor eficiencia a nuestra base de datos, se definen los siguientes tablespace:

Tablespace	Descripción
TFG_Datos	En este tablespace se almacenará el diccionario de datos
TFG_Indices	Este tablespace será destinado al almacenamiento de los índices de las tablas.
TFG_DW	Destinaremos este tablespace al almacenamiento del datawarehouse definido.

Tabla 8: Definición de tablespaces

2.4.2 Usuarios

En este apartado describimos los diferentes usuarios que se crearán para darles acceso a la base de datos. Dado que uno de los requisitos es dar cumplimiento al GDPR⁷ para asegurar la privacidad en el acceso a datos sensibles, se crean dos tipos de roles, básico y avanzado, para poder probar dicha funcionalidad, negando el acceso a las tablas indicadas a los usuarios básicos.

Usuario	Permisos
jcorralesp	Administrador de la base de datos (SYSDBA).
usuario1	Usuario con asignación de rol avanzado, permitiendo conexión a la base de datos

⁷ Reglamento General de Protección de Datos - Wikipedia, la enciclopedia libre. (s. f.). Recuperado 5 de junio de 2021, de https://es.wikipedia.org/wiki/Reglamento_General_de_Protección_de_Datos

	y de selección, modificación y borrado de datos de todas las tablas.
usuario2	Usuario con asignación de rol básico, permitiendo conexión a la base de datos y de selección de todas las tablas salvo la tabla 'Padron_Municipal', solo disponible para usuarios con rol avanzado.

Tabla 9: Definición de usuarios

2.4.3 Tablas

A continuación se detallan las tablas creadas con sus correspondientes campos y restricciones según el diseño lógico.

Tabla: Comunidad_Autonomas				
Campo	Tipo	Clave	Nulo	Descripción
ID_AUTONOMIA	NUMBER(2)	PK	NO	Identificador único de la comunidad autónoma.
DESCRIPCION	VARCHAR2(30)	UN	NO	Nombre de la comunidad autónoma.

Tabla: Provincia				
Campo	Tipo	Clave	Nulo	Descripción
ID_PROVINCIA	NUMBER(2)	PK	NO	Identificador único de la provincia.
DESCRIPCION	VARCHAR2(25)	UN	NO	Nombre de la provincia.
ID_AUTONOMIA	NUMBER(2)	FK	NO	Identificador de la comunidad autónoma a la que pertenece la provincia.

Tabla: Municipio				
Campo	Tipo	Clave	Nulo	Descripción
ID_MUNICIPIO	NUMBER(3)	PK	NO	Identificador único del municipio.
DESCRIPCION	VARCHAR2(50)		NO	Nombre del municipio.
ID_PROVINCIA	NUMBER(2)	FK	NO	Identificador de la provincia a la que pertenece el municipio.

Tabla: Titulacion				
Campo	Tipo	Clave	Nulo	Descripción
ID_TITULACION	NUMBER(2)	PK	NO	Identificador único de la titulación académica.
DESCRIPCION	VARCHAR2(100)	UN	NO	Nombre de la titulación académica.

Tabla: Parametros				
Campo	Tipo	Clave	Nulo	Descripción
ID_PARAMETRO	VARCHAR2(20)	PK	NO	Identificador único del parámetro.
TIPO	VARCHAR2(10)		NO	Tipo de regla que se asigna al parámetro. Los valores posibles para utilizar son: - Partido - Censo - Mesa - Escrutinio
CAMPO	VARCHAR2(20)		NO	Campo que se utilizará para aplicar el parámetro.
VALOR	VARCHAR2(100)		NO	Valor que se asigna al parámetro.

Tabla: Partido_Politico				
Campo	Tipo	Clave	Nulo	Descripción
ID_PARTIDO	NUMBER(3)	PK	NO	Identificador único del partido político.
SIGLAS	VARCHAR2(10)	UN	NO	Siglas que identifican al partido político.
NOMBRE	VARCHAR2(100)	UN	NO	Nombre del partido político.
DIRECCION	VARCHAR2(100)		NO	Dirección fiscal del partido político.
FECHA_INSCRIPCION	DATE		NO	Fecha de inscripción en registro

				público de partidos políticos.
--	--	--	--	--------------------------------

Tabla: Usuarios				
Campo	Tipo	Clave	Nulo	Descripción
ID_USUARIO	VARCHAR2(20)	PK	NO	Identificador único del usuario.
CLAVE	VARCHAR2(15)		NO	Clave de acceso del usuario.
NIVEL	VARCHAR2(8)		NO	Perfil de acceso del usuario. Los valores posibles son: - Normal - Avanzado
ESTADO	VARCHAR2(13)		NO	Estado de la cuenta del usuario. Los valores posibles son: - Habilitado - Deshabilitado

Tabla: Log_Procesos				
Campo	Tipo	Clave	Nulo	Descripción
ID_LOG	NUMBER(15)	PK	NO	Identificador único del registro de los logs de procesos.
FECHA	DATE		NO	Fecha y hora del registro del log.
PROCEDIMIENTO	VARCHAR2(100)		NO	Nombre del procedimiento ejecutado.
PARAMETROS_ENTRADA	VARCHAR2(700)		NO	Parámetros y valores utilizados en la ejecución del procedimiento.
PARAMETROS_SALIDA	VARCHAR2(200)		NO	Parámetros de salida del procedimiento.

Tabla: Convocatoria				
Campo	Tipo	Clave	Nulo	Descripción
ID_CONVOCATORIA	VARCHAR2(20)	PK	NO	Identificador único de la convocatoria.
DESCRIPCION	VARCHAR2(100)		NO	Descripción de la convocatoria.
FECHA_PUBLICACION	DATE		NO	Fecha de publicación de la convocatoria.
FECHA_INICIO	DATE		NO	Fecha y hora de inicio de la votación.
FECHA_FIN	DATE		NO	Fecha y hora de finalización de la votación.
TIPO	VARCHAR2(10)		NO	Ámbito geográfico que abarca la convocatoria. Los valores posibles son: - Estatal - Autonómica - Municipal
AUTONOMIA	NUMBER(2)		SI	Código de la comunidad autónoma en el caso de que el tipo de convocatoria sea 'Autonómica'.
NUMERO_ESCANOS	NUMBER(3)		NO	Número de escaños que se reparten por demarcación en la convocatoria.
ESTADO	VARCHAR2(10)		NO	Estado de la convocatoria. Los valores posibles son: - Abierta - Cerrada - Finalizada - Anulada

Tabla: Convocatoria_Parametros				
Campo	Tipo	Clave	Nulo	Descripción
ID_CONVOCATORIA	VARCHAR2(20)	PK	NO	Identificador único de la convocatoria.
ID_PARAMETRO	VARCHAR2(20)	PK	NO	Identificador único del parámetro.

Tabla: Padron_Municipal				
Campo	Tipo	Clave	Nulo	Descripción
ID_PADRON	VARCHAR2(20)	PK	NO	Identificador único en el padrón municipal.
NOMBRE	VARCHAR2(100)		NO	Nombre de la persona física.

APELLIDO1	VARCHAR2(100)		NO	Primer apellido de la persona física.
APELLIDO2	VARCHAR2(100)		SI	Segundo apellido de la persona física.
NUM_DOCUMENTO	VARCHAR2(20)	UN	NO	Documento de identificación de la persona física.
FECHA_NACIMIENTO	DATE		NO	Fecha de nacimiento de la persona física.
PROVINCIA_NACIMIENTO	VARCHAR2(50)		NO	Provincia de nacimiento de la persona física.
MUNICIPIO_NACIMIENTO	VARCHAR2(85)		NO	Municipio de nacimiento de la persona física.
PAIS	VARCHAR2(60)		NO	Provincia de nacimiento
SEXO	VARCHAR2(6)		NO	Sexo de la persona física. Los valores posibles son: - Hombre - Mujer
PROVINCIA_DOMICILIO	NUMBER(2)		NO	Código de la provincia del domicilio habitual de la persona física.
MUNICIPIO_DOMICILIO	NUMBER(3)		NO	Código del municipio del domicilio habitual de la persona física.
DIRECCION	VARCHAR2(100)		NO	Dirección del domicilio habitual de la persona física.
TITULACION	NUMBER(2)	FK	NO	Código de la mayor titulación obtenida por la persona física.

Tabla: Demarcacion

Campo	Tipo	Clave	Nulo	Descripción
ID_DEMARCACION	NUMBER(15)	PK	NO	Identificador único de la demarcación de una convocatoria.
ESCANOS	NUMBER(3)		NO	Número de escaños que se reparten en la demarcación.
VOTANTES	NUMBER(9)		NO	Número de votantes potenciales que contiene la demarcación.
ID_CONVOCATORIA	VARCHAR2(20)	FK	NO	Identificador único de la convocatoria.

Tabla: Resultados

Campo	Tipo	Clave	Nulo	Descripción
ID_RESULTADO	NUMBER(6)	PK	NO	Identificador único del resultado calculado.
PARTIDO	NUMBER(3)		NO	Código identificador del partido político.
ESCANOS	NUMBER(3)		NO	Número de escaños obtenidos.
VOTOS	NUMBER(9)		NO	Número de votos obtenidos.
PORCENTAJE_VOTOS	NUMBER(5,2)		NO	Porcentaje de votos obtenido.
ID_DEMARCACION	NUMBER(15)	FK	NO	Identificador único de la demarcación donde se obtienen los resultados.

Tabla: Centro_Votacion

Campo	Tipo	Clave	Nulo	Descripción
ID_DEMARCACION	NUMBER(15)	PK	NO	Identificador único de la demarcación.
ID_MUNICIPIO	NUMBER(3)	PK	NO	Identificador único del municipio.
DISTRITO	NUMBER(3)		NO	Código de distrito del centro de votación.
SECCION	NUMBER(3)		NO	Código de sección del centro de votación.
DIRECCION	VARCHAR2(100)		NO	Dirección fiscal del centro de votación.
CODIGO_POSTAL	NUMBER(5)		NO	Código postal del centro de votación.
SORTEADO	CHAR(1)		NO	Indicador de si ha sido sorteada la formación de los miembros de las mesas del centro de votación. Los valores posibles para utilizar son: - S -> 'Si' - N -> 'No'
NOTIFICADO	CHAR(1)		NO	Indicador de si han sido notificados las

				<p>personas elegidas en el sorteo para la formación de las mesas del centro de votación. Los valores posibles para utilizar son:</p> <ul style="list-style-type: none"> - S -> 'Sí' - N -> 'No'
--	--	--	--	---

Tabla: Mesa				
Campo	Tipo	Clave	Nulo	Descripción
ID_MESA	NUMBER(9)	PK	NO	Identificador único de la mesa electoral.
DESCRIPCION	VARCHAR2(100)		NO	Descripción de la mesa electoral.
ESTADO	VARCHAR2(10)		NO	Estado de la mesa electoral. Los valores posibles son: - Abierta - Cerrada - Finalizada - Anulada
ID_DEMARCACION	NUMBER(15)	FK	NO	Identificador único de la demarcación de una convocatoria.
ID_MUNICIPIO	NUMBER(3)	FK	NO	Identificador único del municipio.

Tabla: Votacion				
Campo	Tipo	Clave	Nulo	Descripción
TIPO	VARCHAR2(10)		NO	Indicador del tipo de voto emitido. Los valores posibles para utilizar son: - Presencial - Telemático
CERTIFICADO	VARCHAR2(100)		SI	Código del certificado empleado para realizar el voto. Solo aplica si el valor de 'Tipo' es 'Telemático'.
ID_MESA	NUMBER(9)	FK	NO	Identificador único de la mesa electoral.

Tabla: Recuento				
Campo	Tipo	Clave	Nulo	Descripción
PARTIDO	NUMBER(3)	UN	NO	Código identificador del partido político.
VOTOS	NUMBER(9)		NO	Número de votos contabilizados.
ID_MESA	NUMBER(9)	UN,FK	NO	Identificador único de la mesa electoral.

Tabla: Censo Electoral				
Campo	Tipo	Clave	Nulo	Descripción
ID_CONVOCATORIA	VARCHAR2(20)	PK	NO	Identificador único de la convocatoria.
ID_PADRON	VARCHAR2(20)	PK	NO	Identificador único en el padrón municipal.
FECHA_INSCRIPCION	DATE		NO	Fecha de inscripción en el censo electoral.
ES_MESA	CHAR(1)		NO	Indicador de si ha sido designado para formar parte de una mesa electoral. Los valores posibles para utilizar son: - S -> 'Sí' - N -> 'No'
TIPO	VARCHAR2(8)		SI	Indicador del puesto a ocupar en la mesa electoral. Solo aplica si el valor de 'Es_Mesa' vale 'S'. Los valores posibles para utilizar son: - Titular - Suplente
NOTIFICADO	CHAR(1)		SI	Indicador de si ha sido notificado de la asignación a una mesa electoral. Solo aplica si el valor de 'Es_Mesa' vale 'S'. Los valores posibles para utilizar son: - S -> 'Sí se notifica' - N -> 'No se notifica'
ALEGA	CHAR(1)		SI	Indicador de si se ha presentado una alegación para no formar parte de la

				mesa electoral. Solo aplica si el valor de 'Es_Mesa' vale 'S'. Los valores posibles para utilizar son: - S -> 'Si se presenta' - N -> 'No se presenta'
PRESENTADO	CHAR(1)		SI	Indicador de si se ha presentado a ocupar el puesto designado en la mesa electoral. Solo aplica si el valor de 'Es_Mesa' vale 'S'. Los valores posibles para utilizar son: - S -> 'Si se presenta' - N -> 'No se presenta'
ID_MESA	NUMBER(9)	FK	NO	Identificador único de la mesa electoral asignada.

Tabla: Candidatura				
Campo	Tipo	Clave	Nulo	Descripción
ID_DEMARCACION	NUMBER(15)	PK	NO	Identificador único de la demarcación de una convocatoria.
ID_PARTIDO	NUMBER(3)	PK	NO	Código identificador del partido político.
ESTADO	VARCHAR2(10)		NO	Estado de la candidatura. Los valores posibles para utilizar son: - Presentada - Aprobada - Rechazada - Anulada

Tabla: Candidato				
Campo	Tipo	Clave	Nulo	Descripción
ID_CONVOCATORIA	VARCHAR2(20)	PK	NO	Identificador único de la convocatoria.
ID_PADRON	VARCHAR2(20)	PK	NO	Identificador único en el padrón municipal.
ID_DEMARCACION	NUMBER(15)	PK	NO	Identificador único de la demarcación de una convocatoria.
ID_PARTIDO	NUMBER(3)	PK	NO	Código identificador del partido político.
ORDEN	NUMBER(3)		NO	Número de puesto a ocupar en la candidatura.

Tabla: Log_Padron				
Campo	Tipo	Clave	Nulo	Descripción
ID_LOG	NUMBER(15)	PK	NO	Identificador único del registro de los logs de acceso a los datos del padrón municipal.
FECHA	DATE		NO	Fecha y hora de registro del log.
ID_USUARIO	VARCHAR2(20)	FK	NO	Identificador del usuario que accede a los datos.
ID_PADRON	VARCHAR2(20)	FK	NO	Identificador de la persona del padrón al que se han accedido a sus datos.

Tabla 10: Definición de tablas

También generamos una tabla temporal que nos servirá de apoyo a la hora de generar los escrutinios de las diferentes demarcaciones.

Tabla: Temp_Resultados				
Campo	Tipo	Clave	Nulo	Descripción
PARTIDO	NUMBER(3)	PK	NO	Identificador único de la convocatoria.
ESCANOS	NUMBER(3)	PK	NO	Identificador único en el padrón municipal.
VOTOS	NUMBER(9)	PK	NO	Identificador único de la demarcación de una convocatoria.
VOTOS_ESCANO	NUMBER(9)	PK	NO	Código identificador del partido político.
PORCENTAJE_VOTOS	NUMBER(5,2)		NO	Número de puesto a ocupar en la

				candidatura.
--	--	--	--	--------------

Tabla 11: Definición de tablas auxiliares

2.4.4 Secuencias

Tal y como se definió en el modelo lógico, tenemos una serie de tablas para las que se han definidos campos con valores secuenciales. Estos valores secuenciales van a ser usados como identificadores de estas tablas sin realizar conversión alguna sobre el valor que genere dicha secuencia. Las secuencias empleadas comenzarán por 1 y su incremento será de 1 en 1. La definición de estas es la siguiente:

Secuencia	Tabla	Campo
SEC_RESULTADOS	RESULTADOS	ID_RESULTADO
SEC_MESA	MESA	ID_MESA
SEC_LOG_PADRON	LOG_PADRON	ID_LOG
SEC_LOG_PROCESOS	LOG_PROCESOS	ID_LOG
SEC_DEMARCACION	DEMARCACION	ID_DEMARCACION

Tabla 12: Definición de secuencias

2.4.5 Índices

Los índices son estructuras de datos que mejoran la velocidad de las operaciones permitiendo un acceso rápido a los registros de una tabla. Oracle se encarga de generar un índice automáticamente cuando creamos una restricción 'primary key' o 'unique' sobre una tabla. Dado que en la creación de nuestras tablas hemos creado dichas restricciones, Oracle ya ha generado los índices por nosotros con el mismo nombre que la restricción, pero los ha alojado en el tablespace por defecto del usuario. Dado que queremos ubicar los índices en el tablespace que hemos creado para tal fin, nos ayudaremos del comando Alter Index para mover dichos índices a nuestro tablespace.

La relación de índices que han sido generados es la siguiente:

Índice	Tabla	Campo
PK_CANDIDATO	CANDIDATO	ID_CONVOCATORIA, ID_PADRON, ID_DEMARCACION, ID_PARTIDO
PK_CANDIDATURA	CANDIDATURA	ID_DEMARCACION, ID_PARTIDO
PK_CENSO_ELECTORAL	CENSO_ELECTORAL	ID_CONVOCATORIA, ID_PADRON
PK_CENTRO_VOTACION	CENTRO_VOTACION	ID_DEMARCACION, ID_MUNICIPIO
PK_AUTONOMIA	COMUNIDAD_AUTONOMA	ID_AUTONOMIA
PK_CONVOCATORIA	CONVOCATORIA	ID_CONVOCATORIA
PK_CONVOCATORIA_PARAM	CONVOCATORIA_PARAMETROS	ID_CONVOCATORIA, ID_PARAMETRO
PK_DEMARCACION	DEMARCACION	ID_DEMARCACION
PK_LOG_PADRON	LOG_PADRON	ID_LOG
PK_LOG_PROCESOS	LOG_PROCESOS	ID_LOG
PK_MESA	MESA	ID_MESA
PK_MUNICIPIO	MUNICIPIO	ID_MUNICIPIO
PK_PADRON	PADRON_MUNICIPAL	ID_PADRON
PK_PARAMETROS	PARAMETROS	ID_PARAMETRO
PK_PARTIDO	PARTIDO_POLITICO	ID_PARTIDO
PK_PROVINCIA	PROVINCIA	ID_PROVINCIA
PK_RESULTADOS	RESULTADOS	ID_RESULTADO
PK_TITULACION	TITULACION	ID_TITULACION
PK_USUARIO	USUARIOS	ID_USUARIO

UN_AUTONOMIA	COMUNIDAD_AUTONOMA	DESCRIPCION
UN_PROVINCIA	PROVINCIA	DESCRIPCION
UN_TITULACION	TITULACION	DESCRIPCION
UN1_PARTIDO	PARTIDO_POLITICO	SIGLAS
UN2_PARTIDO	PARTIDO_POLITICO	NOMBRE
UN_PADRON	PADRON_MUNICIPAL	NUM_DOCUMENTO
UN_RECUENTO	RECUENTO	PARTIDO, ID_MESA

Tabla 13: Definición de índices

2.4.6 Triggers

Los triggers son objetos de una base de datos que se asocian a las tablas para que sean ejecutados cuando sucede algún evento sobre dicha tabla. En este caso, hemos diseñado una serie de triggers para poder utilizar las secuencias y para dar respuesta a los procedimientos definidos. La relación de triggers definidos es la siguiente:

Trigger	Tabla	Campo
INSERT_SEC_RESULTADOS	RESULTADOS	ID_RESULTADO
INSERT_SEC_MESA	MESA	ID_MESA
INSERT_SEC_LOG_PADRON	LOG_PADRON	ID_LOG
INSERT_SEC_LOG_PROCESOS	LOG_PROCESOS	ID_LOG
INSERT_SEC_DEMARCACION	DEMARCACION	ID_DEMARCACION

Tabla 14: Definición de triggers

2.5 Diseño Data Warehouse

Según los requisitos definidos **RF-14** y **RNF-04**, se nos pide implementar un Data Warehouse para dar respuesta a una serie de consultas planteadas y que estas tengan un tiempo de respuesta constante 1, que garanticen una respuesta óptima y eficiente en la consulta de dichos datos.

Dado que los datos estadísticos requeridos no requieren de mucha complejidad, se ha optado por la generación de una tabla por cada una de las consultas a definir en nuestro Data Warehouse, puesto que entendemos que un diseño más compacto no nos hubiera aportado grandes ventajas sobre el modelo adoptado y sí que nos hubiera complicado el desarrollo de este.

A continuación, se desglosa la definición de las tablas para cada una de las consultas definidas. Todas estas se alojarán en el tablespace 'Tfg_Dw' definido para tal fin.

Consulta 1. De todos los datos existentes en el sistema, proceso electoral en el que ha habido más participación. Entendemos como participación el porcentaje de personas incluidas en el censo electoral que han votado.

Tabla: DWH_Participacion_Convocatoria				
Campo	Tipo	Clave	Nulo	Descripción
ID_PART_CONV	NUMBER(15)	PK	NO	Identificador único del registro de la participación por convocatoria.
ID_CONVOCATORIA	VARCHAR2(20)		NO	Identificador único de la convocatoria.
PORCENT_PARTICIPACION	NUMBER(5,2)		NO	Porcentaje de participación de las personas del censo.

Consulta 2. Teniendo en cuenta todos los procesos electorales ya celebrados, municipio donde ha habido la mayor participación.

Tabla: DWH_Participacion_Municipio				
Campo	Tipo	Clave	Nulo	Descripción
ID_PART_MUN	NUMBER(15)	PK	NO	Identificador único del registro de la participación por municipio.
ID_CONVOCATORIA	VARCHAR2(20)		NO	Identificador único de la convocatoria.
ID_MUNICIPIO	NUMBER(3)		NO	Identificador único del municipio.
PORCENT_MUNICIPIO	NUMBER(5,2)		NO	Porcentaje de participación del municipio.

Consulta 3. Dado un año cualquiera, porcentaje de votos telemáticos respecto al total de votos emitidos en todos los procesos electorales celebrados en ese año.

Tabla: DWH_Participacion_Telematica				
Campo	Tipo	Clave	Nulo	Descripción
ID_PART_TEL	NUMBER(15)	PK	NO	Identificador único del registro de la participación telemática.
ANO_CONVOCATORIA	VARCHAR2(4)		NO	Año de celebración de la convocatoria.
PORCENTAJE_VOTOS	NUMBER(5,2)		NO	Porcentaje de votos telemáticos.

Consulta 4. Porcentaje de incremento (o decremento) del total de votos del último proceso electoral celebrado respecto al inmediatamente anterior.

Tabla: DWH_Variacion_Votos				
Campo	Tipo	Clave	Nulo	Descripción
ID_VAR_VOTOS	NUMBER(15)	PK	NO	Identificador único del registro de la variación de votos.
ID_CONVOCATORIA	VARCHAR2(20)		NO	Identificador único de la convocatoria.
FECHA	DATE		NO	Fecha de celebración de la convocatoria.
PORCENT_VARIACION	NUMBER(5,2)		NO	Porcentaje de variación del voto.

Consulta 5. Teniendo en cuenta todos los datos de que se dispone, número total de personas que han formado parte más de 2 veces de una mesa electoral.

Tabla: DWH_Fijos_Mesa				
Campo	Tipo	Clave	Nulo	Descripción
ID_FIJOS_MESA	NUMBER(15)	PK	NO	Identificador único del registro de la variación de votos.
PERSONAS	NUMBER(10)		NO	Número de personas

Consulta 6. En un momento cualquiera, porcentaje de personas que han estado seleccionadas para participar en una mesa electoral en las próximas elecciones y que han presentado alegaciones para no formar parte de ella.

Tabla: DWH_Alegacion_Mesa				
Campo	Tipo	Clave	Nulo	Descripción
ID_ALEG_MESA	NUMBER(15)	PK	NO	Identificador único del registro de las alegaciones a mesa.
ID_CONVOCATORIA	VARCHAR2(20)		NO	Identificador único de la convocatoria.
PORCENT_ALEGACION	NUMBER(5,2)		NO	Porcentaje de alegaciones presentadas.

Consulta 7. Teniendo en cuenta los últimos 10 años, número de partidos políticos que se han presentado en todas las elecciones celebradas en todas las demarcaciones electorales de cada proceso.

Tabla: DWH_Partidos_Presentacion				
Campo	Tipo	Clave	Nulo	Descripción
ID_PAR_PRE	NUMBER(15)	PK	NO	Identificador único del registro de los partidos presentados.
ID_PARTIDO	NUMBER(3)		NO	Código identificador del partido político.

Consulta 8. En un momento cualquiera, porcentaje de municipios que no han realizado todavía el sorteo de asignación de miembros de las mesas electorales.

Tabla: DWH_Sorteo_Mesas				
Campo	Tipo	Clave	Nulo	Descripción
ID_SOR_MESA	NUMBER(15)	PK	NO	Identificador único del registro de la no realización del sorteo de mesas.
ID_CONVOCATORIA	VARCHAR2(20)		NO	Identificador único de la convocatoria.
PORCENT_MUNICIPIO	NUMBER(5,2)		NO	Porcentaje de municipios sin asignar miembros a las mesas.

Consulta 9. Analizando todos los procesos electorales ya celebrados, Top20 de mesas electorales que han recibido menos votos (presenciales o telemáticos). Se deberá poder obtener las 20 mesas electorales con menos votos sea cual sea el proceso electoral.

Tabla: DWH_Mesas_Menos_Votos				
Campo	Tipo	Clave	Nulo	Descripción
ID_MESAS_VOTOS	NUMBER(15)	PK	NO	Identificador único del registro de las mesas con menos votos.
ID_CONVOCATORIA	VARCHAR2(20)		NO	Identificador único de la convocatoria.
ID_MESA	NUMBER(9)		NO	Identificador único de la mesa electoral asignada.
NUM_VOTOS	NUMBER(9)		NO	Total del número de votos registrados.

Consulta 10. Teniendo en cuenta el último proceso electoral comunicado por el Gobierno, porcentaje de personas que deberían formar parte de una mesa y a quien no se ha localizado para entregarle la comunicación de participación.

Tabla: DWH_Notificacion_Mesa				
Campo	Tipo	Clave	Nulo	Descripción
ID_NOTIF_MESA	NUMBER(15)	PK	NO	Identificador único del registro de las personas no notificadas.
ID_CONVOCATORIA	VARCHAR2(20)		NO	Identificador único de la convocatoria.
PORCENTAJE_NOTIF	NUMBER(5,2)		NO	Porcentaje de personas no localizadas a notificar.

Consulta 11. Teniendo en cuenta todos los datos de que se dispone, proceso electoral donde ha habido más candidatos en total presentados por los partidos políticos.

Tabla: DWH_Max_Candidatos				
Campo	Tipo	Clave	Nulo	Descripción
ID_MAX_CANDIDATOS	NUMBER(15)	PK	NO	Identificador único del registro del máximo número de candidatos.

ID_CONVOCATORIA	VARCHAR2(20)		NO	Identificador único de la convocatoria.
NUM_CANDIDATOS	NUMBER(5)		NO	Número total de candidatos presentados.

Consulta 12. En las últimas elecciones celebradas, número total de escaños obtenido por cada partido político.

Tabla: DWH_Reparto_Escanos				
Campo	Tipo	Clave	Nulo	Descripción
ID_REP_ESCANOS	NUMBER(15)	PK	NO	Identificador único del registro del reparto de escaños entre partidos.
ID_CONVOCATORIA	VARCHAR2(20)		NO	Identificador único de la convocatoria.
ID_PARTIDO	NUMBER(3)		NO	Código identificador del partido político.
NUM_ESCANOS	NUMBER(3)		NO	Número total de escaños obtenido.

Tabla 15: Definición de tablas DWH

Así mismo, han sido necesarios también la generación de una serie de índices, secuencias y triggers para el data warehouse que se describen a continuación:

Índice	Tabla	Campo
PK_DWH_PARTCONV	DWH_PARTICIPACION_CONVOCATORIA	ID_PART_CONV
PK_DWH_PARTMUN	DWH_PARTICIPACION_MUNICIPIO	ID_PART_MUN
PK_DWH_PARTTEL	DWH_PARTICIPACION_TELEMATICA	ID_PART_TEL
PK_DWH_VARVOT	DWH_VARIACION_VOTOS	ID_VAR_VOTOS
PK_DWH_FIJMES	DWH_FIJOS_MESA	ID_FIJOS_MESA
PK_DWH_ALEMES	DWH_ALEGACION_MESA	ID_ALEG_MESA
PK_DWH_PARPRE	DWH_PARTIDOS_PRESENTACION	ID_PAR_PRE
PK_DWH_SORMES	DWH_SORTEO_MESAS	ID_SOR_MESA
PK_DWH_MESVOT	DWH_MESAS_MENOS_VOTOS	ID_MESAS_VOTOS
PK_DWH_NOTMES	DWH_NOTIFICACION_MESA	ID_NOTIF_MESA
PK_DWH_MAXCAN	DWH_MAX_CANDIDATOS	ID_MAX_CANDIDATOS
PK_DWH_REPESC	DWH_REPARTO_ESCANOS	ID_REP_ESCANOS

Tabla 16: Definición de Índices DWH

Secuencia	Tabla	Campo
SEC_DWH_PART_CONV	DWH_PARTICIPACION_CONVOCATORIA	ID_PART_CONV
SEC_DWH_PART_MUN	DWH_PARTICIPACION_MUNICIPIO	ID_PART_MUN
SEC_DWH_PART_TELEMATICA	DWH_PARTICIPACION_TELEMATICA	ID_PART_TEL
SEC_DWH_VARIACION_VOTOS	DWH_VARIACION_VOTOS	ID_VAR_VOTOS
SEC_DWH_FIJOS_MESA	DWH_FIJOS_MESA	ID_FIJOS_MESA
SEC_DWH_ALEGACION_MESA	DWH_ALEGACION_MESA	ID_ALEG_MESA
SEC_DWH_PARTIDOS_PRES	DWH_PARTIDOS_PRESENTACION	ID_PAR_PRE
SEC_DWH_SORTEO_MESAS	DWH_SORTEO_MESAS	ID_SOR_MESA
SEC_DWH_MESAS_MENOS_VOTOS	DWH_MESAS_MENOS_VOTOS	ID_MESAS_VOTOS
SEC_DWH_NOTIFICACION_MESA	DWH_NOTIFICACION_MESA	ID_NOTIF_MESA
SEC_DWH_MAX_CANDIDATOS	DWH_MAX_CANDIDATOS	ID_MAX_CANDIDATOS
SEC_DWH_REPARTO_ESCANOS	DWH_REPARTO_ESCANOS	ID_REP_ESCANOS

Tabla 17: Definición de Secuencias DWH

Trigger	Tabla	Campo
INSERT_DWH_PART_CONV	DWH_PARTICIPACION_CONVOCATORIA	ID_PART_CONV
INSERT_DWH_PART_MUN	DWH_PARTICIPACION_MUNICIPIO	ID_PART_MUN
INSERT_DWH_PART_TELEMATICA	DWH_PARTICIPACION_TELEMATICA	ID_PART_TEL
INSERT_DWH_VARIACION_VOTOS	DWH_VARIACION_VOTOS	ID_VAR_VOTOS
INSERT_DWH_FIJOS_MESA	DWH_FIJOS_MESA	ID_FIJOS_MESA
INSERT_DWH_ALEGACION_MESA	DWH_ALEGACION_MESA	ID_ALEG_MESA
INSERT_DWH_PARTIDOS_PRES	DWH_PARTIDOS_PRESENTACION	ID_PAR_PRE
INSERT_DWH_SORTEO_MESAS	DWH_SORTEO_MESAS	ID_SOR_MESA
INSERT_DWH_MESAS_MENOS_VOTOS	DWH_MESAS_MENOS_VOTOS	ID_MESAS_VOTOS

INSERT_DWH_NOTIFICACION_MESA	DWH_NOTIFICACION_MESA	ID_NOTIF_MESA
INSERT_DWH_MAX_CANDIDATOS	DWH_MAX_CANDIDATOS	ID_MAX_CANDIDATOS
INSERT_DWH_REPARTO_ESCANOS	DWH_REPARTO_ESCANOS	ID_REP_ESCANOS

Tabla 18: Definición de Triggers DWH

2.6 Implementación procedimientos

En este apartado nos ocuparemos de la creación de los diferentes procedimientos almacenados que se ocuparan de implementar la lógica descrita en el enunciado del proyecto para hacer cumplir todos los requisitos marcados.

Se ha optado por generar diferentes paquetes para poder organizar los diferentes tipos de procedimientos que vamos a generar. La agrupación definida es la siguiente:

Paquete	Descripción
PK_ABM	Este paquete contiene todos los procedimientos almacenados que se ocuparan de la gestión de la ALTA, BAJA y MODIFICACIÓN de las entidades más relevantes.
PK_DWH	Este paquete contiene los procedimientos encargados de la carga de las tablas del Datawarehouse.
PK_PRC	Este paquete contiene los procedimientos de la lógica y funcionalidad del proyecto para hacer cumplir los requerimientos descritos.

Tabla 19: Definición de paquetes de procedimientos

2.6.1 Procedimientos paquete PK_ABM

Procedimiento Almacenado	ALTA_COMUNIDAD_AUTONOMA
Descripción	Dar de alta una nueva comunidad autónoma
Condiciones	La comunidad autónoma no puede existir en la base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pldautonomia: NUMBER(2) pDescripcion: VARCHAR(30)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la Comunidad Autónoma no puede ser nulo.” “ERROR: La descripción de la Comunidad Autónoma no puede ser nula.” “ERROR: La Comunidad Autónoma ya ha sido dado de alta.” “ERROR: SQLERRM”

Procedimiento Almacenado	BAJA_COMUNIDAD_AUTONOMA
Descripción	Eliminar una comunidad autónoma
Condiciones	La comunidad autónoma debe de existir en la base de datos. La comunidad autónoma no puede tener dependencias de foreign key. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pldautonomia: NUMBER(2)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la comunidad autónoma no puede ser nulo.” “ERROR: La comunidad autónoma no existe.” “ERROR: SQLERRM”

Procedimiento Almacenado	MODIF_COMUNIDAD_AUTONOMA
Descripción	Modifica la descripción de una comunidad autónoma
Condiciones	La comunidad autónoma debe de existir en la base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pldautonomia: NUMBER(2)

	pDescripcion: VARCHAR(30)
Parámetros de salida	<p>“OK”, si el proceso finaliza correctamente.</p> <p>“ERROR: El Código de la comunidad autónoma no puede ser nulo.”</p> <p>“ERROR: La Descripción de la comunidad autónoma no puede ser nula.”</p> <p>“ERROR: La comunidad autónoma no existe.”</p> <p>“ERROR: SQLERRM”</p>

Procedimiento Almacenado	ALTA_PROVINCIA
Descripción	Dar de alta una nueva provincia
Condiciones	<p>La provincia no puede existir en base de datos.</p> <p>La comunidad autónoma debe de existir en base de datos.</p> <p>Los parámetros de entrada no pueden ser nulos.</p>
Parámetros de entrada	<p>pIdautonomia: NUMBER(2)</p> <p>pIdprovincia: NUMBER(2)</p> <p>pDescripcion: VARCHAR(25)</p>
Parámetros de salida	<p>“OK”, si el proceso finaliza correctamente.</p> <p>“ERROR: El código de la comunidad autónoma no puede ser nulo.”</p> <p>“ERROR: El código de la provincia no puede ser nulo.”</p> <p>“ERROR: La descripción de la provincia no puede ser nula.”</p> <p>“ERROR: La comunidad autónoma no existe.”</p> <p>“ERROR: La provincia ya ha sido dado de alta para la comunidad autónoma.”</p> <p>“ERROR: SQLERRM”</p>

Procedimiento Almacenado	BAJA_PROVINCIA
Descripción	Eliminar una provincia
Condiciones	<p>La provincia debe de existir en la base de datos.</p> <p>La provincia no puede tener dependencias de foreign key.</p> <p>Los parámetros de entrada no pueden ser nulos.</p>
Parámetros de entrada	pIdprovincia: NUMBER(2)
Parámetros de salida	<p>“OK”, si el proceso finaliza correctamente.</p> <p>“ERROR: El código de la provincia no puede ser nulo.”</p> <p>“ERROR: La provincia no existe.”</p> <p>“ERROR: SQLERRM”</p>

Procedimiento Almacenado	MODIFICACION_PROVINCIA
Descripción	Modifica la descripción de una provincia
Condiciones	<p>La provincia debe de existir en la base de datos.</p> <p>Los parámetros de entrada no pueden ser nulos.</p>
Parámetros de entrada	<p>pIdprovincia: NUMBER(2)</p> <p>pDescripcion: VARCHAR(25)</p>
Parámetros de salida	<p>“OK”, si el proceso finaliza correctamente.</p> <p>“ERROR: El código de la provincia no puede ser nulo.”</p> <p>“ERROR: La descripción de la provincia no puede ser nula.”</p> <p>“ERROR: La provincia no existe.”</p> <p>“ERROR: SQLERRM”</p>

Procedimiento Almacenado	ALTA_MUNICIPIO
Descripción	Dar de alta un nuevo municipio
Condiciones	<p>El municipio no puede existir en base de datos.</p> <p>La provincia debe de existir en base de datos.</p> <p>Los parámetros de entrada no pueden ser nulos.</p>
Parámetros de entrada	<p>pIdprovincia: NUMBER(2)</p> <p>pIdmunicipio: NUMBER(3)</p> <p>pDescripcion: VARCHAR(50)</p>
Parámetros de salida	<p>“OK”, si el proceso finaliza correctamente.</p> <p>“ERROR: El código de la provincia no puede ser nulo.”</p> <p>“ERROR: El código del municipio no puede ser nulo.”</p> <p>“ERROR: La descripción del municipio no puede ser nulo.”</p> <p>“ERROR: La provincia no existe.”</p>

	“ERROR: El municipio ya ha sido dado de alta para la provincia dada.” “ERROR: SQLERRM”
--	---

Procedimiento Almacenado	BAJA_MUNICIPIO
Descripción	Eliminar un municipio
Condiciones	El municipio debe de existir en la base de datos. El municipio no puede tener dependencias de foreign key. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIdmunicipio: NUMBER(3)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código del municipio no puede ser nulo.” “ERROR: El municipio no existe.” “ERROR: SQLERRM”

Procedimiento Almacenado	MODIFICACION_MUNICIPIO
Descripción	Modifica la descripción de un municipio
Condiciones	El municipio debe de existir en la base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIdmunicipio: NUMBER(3) pDescripcion: VARCHAR(50)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código del municipio no puede ser nulo.” “ERROR: La descripción del municipio no puede ser nulo.” “ERROR: El municipio no existe.” “ERROR: SQLERRM”

Procedimiento Almacenado	ALTA_TITULACION
Descripción	Dar de alta una nueva titulación
Condiciones	La titulación no puede existir en base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIdtitulacion: NUMBER(2) pDescripcion: VARCHAR(100)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la titulación no puede ser nulo.” “ERROR: La descripción de la titulación no puede ser nulo.” “ERROR: La titulación ya ha sido dado de alta.” “ERROR: SQLERRM”

Procedimiento Almacenado	BAJA_TITULACION
Descripción	Eliminar una titulación
Condiciones	La titulación debe de existir en la base de datos. La titulación no puede tener dependencias de foreign key. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIdtitulacion: NUMBER(2)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la titulación no puede ser nulo.” “ERROR: La titulación no existe.” “ERROR: SQLERRM”

Procedimiento Almacenado	MODIFICACION_TITULACION
Descripción	Modifica la descripción de una titulación
Condiciones	La titulación debe de existir en la base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIdtitulacion: NUMBER(2) pDescripcion: VARCHAR(100)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la titulación no puede ser nulo.” “ERROR: La descripción de la titulación no puede ser nulo.”

	<p>“ERROR: La titulación no existe.”</p> <p>“ERROR: SQLERRM”</p>
--	--

Procedimiento Almacenado	ALTA_PARAMETROS
Descripción	Dar de alta un nuevo parámetro
Condiciones	El parámetro no puede existir en base de datos. El tipo de parámetro solo puede ser partido, censo, mesa o escrutinio. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIdparametro: VARCHAR (20) pTipo: VARCHAR(10) pCampo: VARCHAR (20) pValor: VARCHAR (100)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código del parámetro no puede ser nulo.” “ERROR: El tipo de parámetro no puede ser nulo.” “ERROR: El valor del tipo de parámetro es incorrecto. Valores correctos: PARTIDO, CENSO, MESA, ESCRUTINIO.” “ERROR: El campo del parámetro no puede ser nulo.” “ERROR: El valor del parámetro no puede ser nulo.” “ERROR: El parámetro ya ha sido dado de alta.” “ERROR: SQLERRM”

Procedimiento Almacenado	BAJA_PARAMETROS
Descripción	Eliminar un parámetro
Condiciones	El parámetro debe de existir en la base de datos. El parámetro no puede tener dependencias de foreign key. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIdparametro: VARCHAR (20)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código del parámetro no puede ser nulo.” “ERROR: El parámetro no existe.” “ERROR: SQLERRM”

Procedimiento Almacenado	MODIFICACION_PARAMETROS
Descripción	Modifica el valor de un parámetro
Condiciones	La titulación debe de existir en la base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIdparametro: VARCHAR (20) pValor: VARCHAR (100)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código del parámetro no puede ser nulo.” “ERROR: El valor del parámetro no puede ser nulo.” “ERROR: El parámetro no existe.” “ERROR: SQLERRM”

Procedimiento Almacenado	ALTA_PARTIDO_POLITICO
Descripción	Dar de alta un nuevo partido político
Condiciones	El partido político no puede existir en base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIdpartido: NUMBER (3) pSiglas: VARCHAR(10) pNombre: VARCHAR (100) pDireccion: VARCHAR (100) pFecinscripcion: DATE
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código del partido político no puede ser nulo.” “ERROR: Las siglas del partido político no puede ser nulo.” “ERROR: El nombre del partido político no puede ser nulo.” “ERROR: La dirección del partido político no puede ser nulo.” “ERROR: La fecha de inscripción del partido político no puede ser nulo.”

	<p>“ERROR: El partido político ya ha sido dado de alta.” “ERROR: SQLERRM”</p>
--	--

Procedimiento Almacenado	BAJA_PARTIDO_POLITICO
Descripción	Eliminar un partido político
Condiciones	El partido político debe de existir en la base de datos. El partido político no puede tener dependencias de foreign key. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	idpartido: NUMBER (3)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código del partido político no puede ser nulo.” “ERROR: El partido político no existe.” “ERROR: SQLERRM”

Procedimiento Almacenado	MODIFICACION_PARTIDO_POLITICO
Descripción	Modifica los datos de un partido político
Condiciones	El partido político debe de existir en la base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIdpartido: NUMBER (3) pSiglas: VARCHAR(10) pNombre: VARCHAR (100) pDireccion: VARCHAR (100) pFecinscripcion: DATE
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código del partido político no puede ser nulo.” “ERROR: Las siglas del partido político no puede ser nulo.” “ERROR: El nombre del partido político no puede ser nulo.” “ERROR: La dirección del partido político no puede ser nulo.” “ERROR: La fecha de inscripción del partido político no puede ser nulo.” “ERROR: El partido político no existe.” “ERROR: SQLERRM”

Procedimiento Almacenado	ALTA_USUARIOS
Descripción	Dar de alta un nuevo usuario
Condiciones	El usuario no puede existir en base de datos. El nivel solo puede ser normal o avanzado. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIdusuario: VARCHAR(20) pClave: VARCHAR(15) pNivel: VARCHAR (8)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código del usuario no puede ser nulo.” “ERROR: La clave del usuario no puede ser nula.” “ERROR: El nivel del usuario no puede ser nulo.” “ERROR: El valor del nivel del usuario es incorrecto. Valores correctos: NORMAL, AVANZADO.” “ERROR: El usuario ya ha sido dado de alta.” “ERROR: SQLERRM”

Procedimiento Almacenado	BAJA_USUARIOS
Descripción	Cambia el estado del usuario a Deshabilitado.
Condiciones	El usuario debe de existir en la base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIdusuario: VARCHAR(20)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código del usuario no puede ser nulo.” “ERROR: El usuario no existe.” “ERROR: SQLERRM”

Procedimiento Almacenado	MODIFICACION_ USUARIOS
Descripción	Modifica los datos de un usuario
Condiciones	El usuario debe de existir en la base de datos. El nivel solo puede ser normal o avanzado. El estado solo puede ser habilitado o deshabilitado. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIdusuario: VARCHAR(20) pClave: VARCHAR(15) pNivel: VARCHAR (8) pEstado: VARCHAR (13)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código del usuario no puede ser nulo.” “ERROR: La clave del usuario no puede ser nula.” “ERROR: El nivel del usuario no puede ser nulo.” “ERROR: El valor del nivel del usuario es incorrecto. Valores correctos: NORMAL, AVANZADO.” “ERROR: El estado del usuario no puede ser nulo.” “ERROR: El valor del estado del usuario es incorrecto. Valores correctos: HABILITADO, DESHABILITADO.” “ERROR: El usuario no existe.” “ERROR: SQLERRM”

Procedimiento Almacenado	ALTA_ CONVOCATORIA
Descripción	Dar de alta una convocatoria
Condiciones	La convocatoria no puede existir en base de datos. El número de escaños no puede ser menor que cero. La fecha de finalización debe de ser mayor que la fecha de inicio. La fecha de publicación debe de ser menor que la fecha de inicio y de finalización. El tipo de convocatoria debe de ser estatal, autonómica o municipal. Si el tipo de convocatoria es autonómica, el valor de la autonomía no puede ser nulo. Si el tipo de convocatoria es autonómica, la autonomía debe de existir en base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIdconvocatoria: VARCHAR(20) pDescripcion: VARCHAR(100) pPublicacion: DATE pFinicio: DATE pFfin: DATE pTipo: VARCHAR (10) pAutonomia: NUMBER(2) pNumescanos: NUMBER(3)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la convocatoria no puede ser nulo.” “ERROR: La descripción de la convocatoria no puede ser nula.” “ERROR: La fecha de publicación de la convocatoria no puede ser nula.” “ERROR: La fecha de inicio de la convocatoria no puede ser nula.” “ERROR: La fecha de finalización de la convocatoria no puede ser nula.” “ERROR: El tipo de convocatoria no puede ser nulo.” “ERROR: El valor del tipo de convocatoria es incorrecto. Valores correctos: ESTATAL, AUTONOMICA, MUNICIPAL.” “ERROR: El valor de la autonomía no puede ser nulo si el tipo de convocatoria es autonómica.” “ERROR: El número de escaños de la convocatoria no puede ser nulo o menor que CERO.” “ERROR: La comunidad autónoma no existe.” “ERROR: La convocatoria ya ha sido dado de alta.” “ERROR: SQLERRM”

Procedimiento Almacenado	BAJA_ CONVOCATORIA
Descripción	Cambia el estado de la convocatoria a ANULADA.

Condiciones	La convocatoria debe de existir en la base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIdconvocatoria: VARCHAR(20)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la convocatoria no puede ser nulo.” “ERROR: La convocatoria no existe.” “ERROR: SQLERRM”

Procedimiento Almacenado	MODIFICACION_CONVOCATORIA
Descripción	Modifica los datos de una convocatoria
Condiciones	La convocatoria debe de existir en la base de datos. El número de escaños no puede ser menor que cero. La fecha de finalización debe de ser mayor que la fecha de inicio. La fecha de publicación debe de ser menor que la fecha de inicio y de finalización. El tipo de convocatoria debe de ser estatal, autonómica o municipal. Si el tipo de convocatoria es autonómica, el valor de la autonomía no puede ser nulo. Si el tipo de convocatoria es autonómica, la autonomía debe de existir en base de datos. El estado de la convocatoria solo puede ser abierta, cerrada, finalizada o anulada. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIdconvocatoria: VARCHAR(20) pDescripcion: VARCHAR(100) pPublicacion: DATE pFinicio: DATE pFfin: DATE pTipo: VARCHAR (10) pAutonomia: NUMBER(2) pNumescanos: NUMBER(3) pEstado: VARCHAR (10)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la convocatoria no puede ser nulo.” “ERROR: La descripción de la convocatoria no puede ser nula.” “ERROR: La fecha de publicación de la convocatoria no puede ser nula.” “ERROR: La fecha de inicio de la convocatoria no puede ser nula.” “ERROR: La fecha de finalización de la convocatoria no puede ser nula.” “ERROR: El tipo de convocatoria no puede ser nulo.” “ERROR: El valor del tipo de convocatoria es incorrecto. Valores correctos: ESTATAL, AUTONOMICA, MUNICIPAL.” “ERROR: El valor de la autonomía no puede ser nulo si el tipo de convocatoria es autonómica.” “ERROR: El número de escaños de la convocatoria no puede ser nulo o menor que CERO.” “ERROR: El estado de la convocatoria no puede ser nulo.” “ERROR: El estado de la convocatoria es incorrecto. Valores correctos: ABIERTA, CERRADA, FINALIZADA, ANULADA.” “ERROR: La comunidad autónoma no existe.” “ERROR: La convocatoria no existe.” “ERROR: SQLERRM”

Procedimiento Almacenado	ALTA_CONVOCATORIA_PARAMETROS
Descripción	Dar de alta parámetros para una convocatoria
Condiciones	La convocatoria debe de existir en la base de datos. El parámetro debe de existir en la base de datos. Los parámetros de la convocatoria no pueden existir en base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIdconvocatoria: VARCHAR(20) pIdparametro: VARCHAR(20)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la convocatoria no puede ser nulo.” “ERROR: El código del parámetro no puede ser nulo.”

	<p>“ERROR: La convocatoria no existe.”</p> <p>“ERROR: El parámetro no existe.”</p> <p>“ERROR: El parámetro ya ha sido dado de alta para esa convocatoria.”</p> <p>“ERROR: SQLERRM”</p>
--	--

Procedimiento Almacenado	BAJA_CONVOCATORIA_PARAMETROS
Descripción	Eliminar un parámetro para una convocatoria
Condiciones	<p>La convocatoria debe de existir en la base de datos.</p> <p>El parámetro debe de existir en la base de datos.</p> <p>Los parámetros de la convocatoria deben de existir en la base de datos.</p> <p>Los parámetros de entrada no pueden ser nulos.</p>
Parámetros de entrada	<p>pldconvocatoria: VARCHAR(20)</p> <p>plddparametro: VARCHAR(20)</p>
Parámetros de salida	<p>“OK”, si el proceso finaliza correctamente.</p> <p>“ERROR: El código de la convocatoria no puede ser nulo.”</p> <p>“ERROR: El código del parámetro no puede ser nulo.”</p> <p>“ERROR: La convocatoria no existe.”</p> <p>“ERROR: El parámetro no existe.”</p> <p>“ERROR: El parámetro no existe para esa convocatoria.”</p> <p>“ERROR: SQLERRM”</p>

Procedimiento Almacenado	ALTA_PADRON_MUNICIPAL
Descripción	Dar de alta una persona en el padrón municipal
Condiciones	<p>La persona no puede existir en base de datos.</p> <p>La provincia del domicilio debe existir en base de datos.</p> <p>El municipio del domicilio debe existir en base de datos.</p> <p>La titulación debe existir en base de datos.</p> <p>El sexo de la persona solo puede contener el valor Hombre o Mujer.</p> <p>Los parámetros de entrada no pueden ser nulos.</p>
Parámetros de entrada	<p>pldpadron: VARCHAR(20)</p> <p>pNombre: VARCHAR(100)</p> <p>pApellido1: VARCHAR (100)</p> <p>pApellido2: VARCHAR (100)</p> <p>pNumdoc: VARCHAR (20)</p> <p>pFecnac: DATE</p> <p>pProvnac: VARCHAR (50)</p> <p>pMunnac: VARCHAR (85)</p> <p>pPais: VARCHAR (60)</p> <p>pSexo: VARCHAR (6)</p> <p>pProvdomic: VARCHAR (25)</p> <p>pMundomic: VARCHAR (50)</p> <p>pDireccion: VARCHAR (100)</p> <p>pldtitulacion: NUMBER(2)</p>
Parámetros de salida	<p>“OK”, si el proceso finaliza correctamente.</p> <p>“ERROR: El código del padrón no puede ser nulo.”</p> <p>“ERROR: El nombre de la persona no puede ser nulo.”</p> <p>“ERROR: El primer apellido de la persona no puede ser nulo.”</p> <p>“ERROR: El número de documento de la persona no puede ser nulo.”</p> <p>“ERROR: La fecha de nacimiento de la persona no puede ser nulo.”</p> <p>“ERROR: La provincia de nacimiento de la persona no puede ser nulo.”</p> <p>“ERROR: El municipio de nacimiento de la persona no puede ser nulo.”</p> <p>“ERROR: El país de la persona no puede ser nulo.”</p> <p>“ERROR: El sexo de la persona no puede ser nulo.”</p> <p>“ERROR: El sexo de la persona es incorrecto. Valores correctos: HOMBRE, MUJER.”</p> <p>“ERROR: La provincia del domicilio de la persona no puede ser nulo.”</p> <p>“ERROR: El municipio del domicilio de la persona no puede ser nulo.”</p> <p>“ERROR: La dirección del domicilio de la persona no puede ser nulo.”</p> <p>“ERROR: El código de la titulación de la persona no puede ser nulo.”</p> <p>“ERROR: La persona ya ha sido dado de alta.”</p> <p>“ERROR: La provincia del domicilio no existe.”</p> <p>“ERROR: El municipio del domicilio no existe.”</p> <p>“ERROR: La titulación no existe.”</p>

	"ERROR: SQLERRM"
Procedimiento Almacenado	BAJA_PADRON_MUNICIPAL
Descripción	Eliminar una persona del padrón municipal
Condiciones	La persona debe de existir en la base de datos. La persona no puede tener dependencias de foreign key. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIpadron: VARCHAR(20)
Parámetros de salida	"OK", si el proceso finaliza correctamente. "ERROR: El código del padrón no puede ser nulo." "ERROR: La persona no existe." "ERROR: SQLERRM"
Procedimiento Almacenado	MODIFICACION_PADRON_MUNICIPAL
Descripción	Modifica los datos de una persona del padrón municipal
Condiciones	La persona debe de existir en la base de datos. La provincia del domicilio debe existir en base de datos. El municipio del domicilio debe existir en base de datos. La titulación debe existir en base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIpadron: VARCHAR(20) pProvdomic: VARCHAR (25) pMundomic: VARCHAR (50) pDireccion: VARCHAR (100) pDtitulacion: NUMBER(2)
Parámetros de salida	"OK", si el proceso finaliza correctamente. "ERROR: El código del padrón no puede ser nulo." "ERROR: La provincia del domicilio de la persona no puede ser nulo." "ERROR: El municipio del domicilio de la persona no puede ser nulo." "ERROR: La dirección del domicilio de la persona no puede ser nulo." "ERROR: El código de la titulación de la persona no puede ser nulo." "ERROR: La persona no existe." "ERROR: La provincia del domicilio no existe." "ERROR: El municipio del domicilio no existe." "ERROR: La titulación no existe." "ERROR: SQLERRM"
Procedimiento Almacenado	ALTA_DEMARCACION
Descripción	Dar de alta una demarcación
Condiciones	La demarcación no puede existir en base de datos. El número de escaños no puede ser menor que cero. El número de votantes no puede ser menor que cero. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIddemarcacion: NUMBER(3) pEscanos: NUMBER(3) pVotantes: NUMBER(9) pIldconvocatoria: VARCHAR(20)
Parámetros de salida	"OK", si el proceso finaliza correctamente. "ERROR: El código de la demarcación no puede ser nulo." "ERROR: El número de escaños de la demarcación no puede ser nulo o menor que CERO." "ERROR: El número de votantes no puede ser nulo o menor que CERO." "ERROR: El código de la convocatoria no puede ser nulo." "ERROR: La demarcación ya ha sido dado de alta." "ERROR: SQLERRM"
Procedimiento Almacenado	BAJA_DEMARCACION
Descripción	Eliminar una demarcación
Condiciones	La demarcación debe de existir en la base de datos. Los parámetros de entrada no pueden ser nulos.

Parámetros de entrada	plddemarcacion: NUMBER(3) pldconvocatoria: VARCHAR(20)
Parámetros de salida	"OK", si el proceso finaliza correctamente. "ERROR: El código de la demarcación no puede ser nulo." "ERROR: El código de la convocatoria no puede ser nulo." "ERROR: La demarcación no existe." "ERROR: SQLERRM"

Procedimiento Almacenado	MODIFICACION_DEMARCACION
Descripción	Modifica los datos de una demarcación
Condiciones	La demarcación debe de existir en la base de datos. El número de escaños no puede ser menor que cero. El número de votantes no puede ser menor que cero. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	plddemarcacion: NUMBER(3) pEscanos: NUMBER(3) pVotantes: NUMBER(9) pldconvocatoria: VARCHAR(20)
Parámetros de salida	"OK", si el proceso finaliza correctamente. "ERROR: El código de la demarcación no puede ser nulo." "ERROR: El número de escaños de la demarcación no puede ser nulo o menor que CERO." "ERROR: El número de votantes no puede ser nulo o menor que CERO." "ERROR: El código de la convocatoria no puede ser nulo." "ERROR: La demarcación no existe." "ERROR: SQLERRM"

Procedimiento Almacenado	ALTA_CENTRO_VOTACION
Descripción	Dar de alta un centro de votación
Condiciones	El centro de votación no puede existir en base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	plddemarcacion: NUMBER(3) pldmunicipio: NUMBER(3) pDistrito: NUMBER(3) pSeccion: NUMBER(3) pDireccion: VARCHAR(100) pCodpostal: NUMBER(5)
Parámetros de salida	"OK", si el proceso finaliza correctamente. "ERROR: El código de la demarcación no puede ser nulo." "ERROR: El código del municipio no puede ser nulo." "ERROR: El distrito no puede ser nulo." "ERROR: La sección no puede ser nulo." "ERROR: La dirección del centro no puede ser nulo." "ERROR: El código postal no puede ser nulo." "ERROR: El centro de votación ya ha sido dado de alta." "ERROR: SQLERRM"

Procedimiento Almacenado	BAJA_CENTRO_VOTACION
Descripción	Eliminar un centro de votación
Condiciones	El centro de votación debe de existir en la base de datos. El centro de votación no puede tener dependencias de foreign key. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	plddemarcacion: NUMBER(3) pldmunicipio: NUMBER(3)
Parámetros de salida	"OK", si el proceso finaliza correctamente. "ERROR: El código de la demarcación no puede ser nulo." "ERROR: El código del municipio no puede ser nulo." "ERROR: El centro de votación no existe." "ERROR: SQLERRM"

Procedimiento	MODIFICACION_CENTRO_VOTACION
---------------	-------------------------------------

Almacenado	
Descripción	Modifica los datos de un centro de votación
Condiciones	El centro de votación debe de existir en la base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	plddemarcacion: NUMBER(3) pldmunicipio: NUMBER(3) pDistrito: NUMBER(3) pSeccion: NUMBER(3) pDireccion: VARCHAR(100) pCodpostal: NUMBER(5)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la demarcación no puede ser nulo.” “ERROR: El código del municipio no puede ser nulo.” “ERROR: El distrito no puede ser nulo.” “ERROR: La sección no puede ser nulo.” “ERROR: La dirección del centro no puede ser nulo.” “ERROR: El código postal no puede ser nulo.” “ERROR: El centro de votación no existe.” “ERROR: SQLERRM”

Procedimiento Almacenado	ALTA_MESA
Descripción	Dar de alta una mesa electoral
Condiciones	La mesa electoral no puede existir en base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pldmesa: NUMBER(9) pDescripcion: VARCHAR(100) plddemarcacion: NUMBER(3) pldmunicipio: NUMBER(3)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la mesa no puede ser nulo.” “ERROR: La descripción de la mesa no puede ser nulo.” “ERROR: El código demarcación no puede ser nulo.” “ERROR: El código de municipio no puede ser nulo.” “ERROR: La mesa ya ha sido dado de alta.” “ERROR: SQLERRM”

Procedimiento Almacenado	BAJA_MESA
Descripción	Cambia el estado de la mesa electoral a ANULADA.
Condiciones	La mesa electoral debe de existir en la base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pldmesa: NUMBER (9)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la mesa no puede ser nulo.” “ERROR: La mesa no existe.” “ERROR: SQLERRM”

Procedimiento Almacenado	MODIFICACION_MESA
Descripción	Modifica los datos de una mesa electoral
Condiciones	La mesa electoral debe de existir en la base de datos. El estado de la mesa solo puede contener los valores abierta, cerrada, finalizada o anulada. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pldmesa: NUMBER(9) pDescripcion: VARCHAR(100) pEstado: VARCHAR(10)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la mesa no puede ser nulo.” “ERROR: La descripción de la mesa no puede ser nulo.” “ERROR: El estado de la mesa no puede ser nulo.” “ERROR: El valor del estado de la mesa es incorrecto. Valores correctos: ABIERTA, CERRADA, FINALIZADA, ANULADA.”

	<p>"ERROR: La mesa no existe." "ERROR: SQLERRM"</p>
--	--

Procedimiento Almacenado	ALTA_CENSO_ELECTORAL
Descripción	Dar de alta una persona en el censo electoral
Condiciones	La persona no puede existir en base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIdconvocatoria: VARCHAR(20) pIdpadron: VARCHAR(20) pFeInsc: DATE pIdmesa: NUMBER(9)
Parámetros de salida	"OK", si el proceso finaliza correctamente. "ERROR: El código de la convocatoria no puede ser nulo." "ERROR: El código del padrón no puede ser nulo." "ERROR: La fecha de inscripción no puede ser nulo." "ERROR: El código de la mesa no puede ser nulo." "ERROR: La persona ya ha sido dado de alta en el censo." "ERROR: SQLERRM"

Procedimiento Almacenado	BAJA_CENSO_ELECTORAL
Descripción	Eliminar una persona del censo electoral
Condiciones	La persona debe de existir en la base de datos. La persona no puede tener dependencias de foreign key. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIdconvocatoria: VARCHAR(20) pIdpadron: VARCHAR(20)
Parámetros de salida	"OK", si el proceso finaliza correctamente. "ERROR: El código de la convocatoria no puede ser nulo." "ERROR: El código del padrón no puede ser nulo." "ERROR: La persona no existe." "ERROR: SQLERRM"

Procedimiento Almacenado	MODIFICACION_CENSO_ELECTORAL
Descripción	Modifica los datos de una persona del censo electoral
Condiciones	La persona debe de existir en la base de datos. El indicador de si ha sido asignado a una mesa solo puede contener el valor S o N. Si el valor del campo esmesa es S, los campos tipo, notificado y presentado no pueden ser nulos. El campo tipo solo puede contener el valor TITULAR o SUPLENTE. El campo notificado solo puede contener el valor S o N. El campo presentado solo puede contener el valor S o N. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIdconvocatoria: VARCHAR(20) pIdpadron: VARCHAR(20) pEsmesa: CHAR(1) pTipo: VARCHAR(8) pNotificado: CHAR(1) pPresentado: CHAR(1)
Parámetros de salida	"OK", si el proceso finaliza correctamente. "ERROR: El código de la convocatoria no puede ser nulo." "ERROR: El código del padrón no puede ser nulo." "ERROR: El indicador de si pertenece a una mesa electoral no puede ser nulo." "ERROR: El valor del campo tipo no puede ser nulo si la persona ha sido designada para una mesa." "ERROR: El valor del campo tipo es incorrecto. Valores correctos: TITULAR,SUPLENTE." "ERROR: El valor del campo notificado no puede ser nulo si la persona ha sido designada para una mesa." "ERROR: El valor del campo notificado es incorrecto. Valores correctos: S(SI) ,N(NO)"

	<p>“ERROR: El valor del campo presentado no puede ser nulo si la persona ha sido designada para una mesa.”</p> <p>“ERROR: El valor del campo presentado es incorrecto. Valores correctos: S(SI) ,N(NO).”</p> <p>“ERROR: La persona no existe.”</p> <p>“ERROR: SQLERRM”</p>
--	--

Procedimiento Almacenado	ALTA_CANDIDATURA
Descripción	Dar de alta una candidatura
Condiciones	La candidatura no puede existir en base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	plddemarcacion: NUMBER(3) pldpartido: NUMBER(3)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la demarcación no puede ser nulo.” “ERROR: El código del partido político no puede ser nulo.” “ERROR: La candidatura ya ha sido dado de alta.” “ERROR: SQLERRM”

Procedimiento Almacenado	BAJA_CANDIDATURA
Descripción	Cambia el estado de la candidatura a ANULADA.
Condiciones	La candidatura debe de existir en la base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	plddemarcacion: NUMBER(3) pldpartido: NUMBER(3)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la demarcación no puede ser nulo.” “ERROR: El código del partido político no puede ser nulo.” “ERROR: La candidatura no existe.” “ERROR: SQLERRM”

Procedimiento Almacenado	MODIFICACION_CANDIDATURA
Descripción	Modifica los datos de una candidatura
Condiciones	La candidatura debe de existir en la base de datos. El estado de la candidatura solo puede ser presentada, aprobada, rechazada o anulada. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	plddemarcacion: NUMBER(3) pldpartido: NUMBER(3) pEstado: VARCHAR(10)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la demarcación no puede ser nulo.” “ERROR: El código del partido político no puede ser nulo.” “ERROR: El estado de la candidatura no puede ser nula.” “ERROR: ERROR: El valor del estado de la candidatura es incorrecto. Valores correctos: PRESENTADA, APROBADA, RECHAZADA, ANULADA.” “ERROR: La candidatura no existe.” “ERROR: SQLERRM”

Procedimiento Almacenado	ALTA_CANDIDATO
Descripción	Dar de alta un candidato
Condiciones	El candidato no puede existir en base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pldconvocatoria: VARCHAR(20) pldpadron: VARCHAR(20) plddemarcacion: NUMBER(3) pldpartido: NUMBER(3) pOrden: NUMBER(3)
Parámetros de salida	“OK”, si el proceso finaliza correctamente.

	<p>“ERROR: El código de la convocatoria no puede ser nulo.”</p> <p>“ERROR: El código del padrón no puede ser nulo.”</p> <p>“ERROR: El código de la demarcación no puede ser nulo.”</p> <p>“ERROR: El código del partido político no puede ser nulo.”</p> <p>“ERROR: El orden en la candidatura no puede ser nulo.”</p> <p>“ERROR: El candidato ya ha sido dado de alta.”</p> <p>“ERROR: SQLERRM”</p>
--	--

Procedimiento Almacenado	BAJA_CANDIDATO
Descripción	Eliminar un candidato
Condiciones	El candidato debe de existir en la base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIdconvocatoria: VARCHAR(20) pIdpadron: VARCHAR(20) pIddemarcacion: NUMBER(3) pIdpartido: NUMBER(3)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la convocatoria no puede ser nulo.” “ERROR: El código del padrón no puede ser nulo.” “ERROR: El código de la demarcación no puede ser nulo.” “ERROR: El código del partido político no puede ser nulo.” “ERROR: El candidato no existe.” “ERROR: SQLERRM”

Procedimiento Almacenado	MODIFICACION_CANDIDATO
Descripción	Modifica los datos de un candidato
Condiciones	El candidato debe de existir en la base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIdconvocatoria: VARCHAR(20) pIdpadron: VARCHAR(20) pIddemarcacion: NUMBER(3) pIdpartido: NUMBER(3) pOrden: NUMBER(3)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la convocatoria no puede ser nulo.” “ERROR: El código del padrón no puede ser nulo.” “ERROR: El código de la demarcación no puede ser nulo.” “ERROR: El código del partido político no puede ser nulo.” “ERROR: El orden en la candidatura no puede ser nulo.” “ERROR: El candidato no existe.” “ERROR: SQLERRM”

Tabla 20: Definición de los procedimientos del paquete PK_ABM

2.6.2 Procedimientos paquete PK_PRC

Procedimiento Almacenado	ALTA_RECUESTO
Descripción	Dar de alta un recuento para una mesa y un partido en concreto.
Condiciones	El recuento no puede existir en base de datos. El número de votos no puede ser menor que cero. La mesa debe de encontrarse en estado Cerrada. El partido político debe de tener candidatura presentada en la mesa. El número de votos del recuento no puede superar al número de votantes. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIdpartido: NUMBER(3) pVotos: NUMBER(9) pIdmesa: NUMBER(9)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código del partido político no puede ser nulo.” “ERROR: El número de votos no puede ser nulo o menor que CERO.” “ERROR: El código de la mesa no puede ser nulo.”

	<p>“ERROR: La mesa no existe.”</p> <p>“ERROR: El partido político no tiene representación en esta mesa.”</p> <p>“ERROR: La mesa no se encuentra cerrada para poder realizar el recuento.”</p> <p>“ERROR: El recuento ya ha sido dado de alta.”</p> <p>“ERROR: El número de votos indicados excede en el número de votantes de la mesa.”</p> <p>“ERROR: SQLERRM”</p>
--	---

Procedimiento Almacenado	BAJA_RECUESTO
Descripción	Eliminar un recuento de una mesa electoral
Condiciones	El recuento debe de existir en la base de datos. La mesa no debe de encontrarse en estado Finalizada. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIdpartido: NUMBER(3) pIdmesa: NUMBER(9)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código del partido político no puede ser nulo.” “ERROR: El código de la mesa no puede ser nulo.” “ERROR: El recuento no existe.” “ERROR: El recuento no se puede eliminar si la mesa se encuentra finalizada.” “ERROR: SQLERRM”

Procedimiento Almacenado	MODIFICACION_RECUESTO
Descripción	Modifica el número de votos de un recuento realizado
Condiciones	El recuento debe de existir en la base de datos. El número de votos no puede ser menor que cero. La mesa debe de encontrarse en estado Cerrada. El número de votos del recuento no puede superar al número de votantes. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pIdpartido: NUMBER(3) pVotos: NUMBER(9) pIdmesa: NUMBER(9)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código del partido político no puede ser nulo.” “ERROR: El número de votos no puede ser nulo o menor que CERO.” “ERROR: El código de la mesa no puede ser nulo.” “ERROR: El recuento no existe.” “ERROR: El recuento no se puede actualizar si la mesa no está cerrada.” “ERROR: El número de votos indicados excede en el número de votantes de la mesa.” “ERROR: SQLERRM”

Procedimiento Almacenado	ALTA_VOTACION
Descripción	Dar de alta una votación de una persona en una mesa
Condiciones	La votación no puede existir en base de datos. El valor del campo tipo solo puede ser presencial o telemático. La mesa debe de encontrarse en estado Abierta. Si el tipo de voto es Telemático, es obligatorio informar el certificado utilizado. El número de votos registrados no puede superar al número de votantes. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pTipo: VARCHAR(10) pCertificado: VARCHAR(100) pIdmesa: NUMBER(9)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El tipo de voto no puede ser nulo.” “ERROR: El valor del tipo de voto es incorrecto. Valores correctos: PRESENCIAL, TELEMATICO” “ERROR: El certificado no puede ser nulo si el voto es Telemático.” “ERROR: El código de la mesa no puede ser nulo.”

	<p>“ERROR: La mesa no existe.”</p> <p>“ERROR: La mesa no se encuentra Abierta para poder votar.”</p> <p>“ERROR: En esta mesa no se pueden registrar más votos.”</p> <p>“ERROR: SQLERRM”</p>
--	---

Procedimiento Almacenado	GENERAR_DEMARCACION
Descripción	Dar de alta las demarcaciones de una convocatoria, así como sus centros de votación y mesas electorales correspondientes.
Condiciones	La demarcación no puede existir en base de datos. La convocatoria debe de estar creada y en estado Abierta. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pldconvocatoria: VARCHAR(20)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la convocatoria no puede ser nulo.” “ERROR: La convocatoria no existe.” “ERROR: La convocatoria no se encuentra ABIERTA, por lo que no se pueden generar las demarcaciones.” “ERROR: La convocatoria ya tiene las demarcaciones creadas.” “ERROR: SQLERRM”

Procedimiento Almacenado	GENERAR_CENSO_ELECTORAL
Descripción	Dar de alta el censo electoral que puede participar en una convocatoria.
Condiciones	La convocatoria debe de existir en base de datos. La convocatoria debe de encontrarse en el estado Abierta. El censo electoral no debe de estar ya generado. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pldconvocatoria: VARCHAR(20)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la convocatoria no puede ser nulo.” “ERROR: La convocatoria no existe.” “ERROR: La convocatoria no se encuentra ABIERTA, por lo que no se pueden generar el censo electoral.” “ERROR: La convocatoria ya tiene generado el censo electoral.” “ERROR: SQLERRM”

Procedimiento Almacenado	SORTEO_MIEMBROS_MESA
Descripción	Genera el sorteo de las personas que formarán las mesas electorales, tanto titulares como suplentes.
Condiciones	La convocatoria debe de existir en base de datos. La demarcación debe de existir en base de datos. La convocatoria debe tener generado el censo electoral. Las candidaturas de la demarcación deben de encontrarse aprobadas. El sorteo no debe de estar realizado. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pldconvocatoria: VARCHAR(20) plddemarcacion: NUMBER(15)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la convocatoria no puede ser nulo.” “ERROR: El código de la demarcación no puede ser nulo.” “ERROR: La convocatoria no existe.” “ERROR: La demarcación no existe.” “ERROR: La convocatoria no tiene generado el censo electoral.” “ERROR: El sorteo de formación de mesas ya se encuentra realizado.” “ERROR: Existen candidaturas pendientes de validar.” “ERROR: SQLERRM”

Procedimiento Almacenado	ENVIAR_NOTIFICACION_MESA
Descripción	Se realiza el envío de las notificaciones de formación de una mesa electoral a las personas elegidas en el sorteo.

Condiciones	La convocatoria debe de existir en base de datos. El municipio debe de existir en base de datos. El municipio debe de tener realizado el sorteo de los miembros de la mesa. La notificación del sorteo no debe de estar realizada. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pldconvocatoria: VARCHAR(20) pldmunicipio: NUMBER(3)
Parámetros de salida	"OK", si el proceso finaliza correctamente. "ERROR: El código de la convocatoria no puede ser nulo." "ERROR: El código del municipio no puede ser nulo." "ERROR: La convocatoria no existe." "ERROR: El sorteo de formación de mesas no se encuentra realizado." "ERROR: La notificación del sorteo de formación de mesas ya se encuentra realizado." "ERROR: SQLERRM"

Procedimiento Almacenado	VALIDAR_CANDIDATURA
Descripción	Se valida que la candidatura presentada por un partido político cumple con los requisitos para poder participar en la demarcación.
Condiciones	La convocatoria debe de existir en base de datos. La demarcación debe de existir en base de datos. El partido político debe de existir en base de datos. La candidatura debe de existir en base de datos. La candidatura debe de contener el mínimo de candidatos requerido. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pldconvocatoria: VARCHAR(20) plddemarcacion: NUMBER(15) pldpartido: NUMBER(3)
Parámetros de salida	"OK", si el proceso finaliza correctamente. "ERROR: El código de la convocatoria no puede ser nulo." "ERROR: El código de la demarcación no puede ser nulo." "ERROR: El código del partido político no puede ser nulo." "ERROR: La convocatoria no existe." "ERROR: La demarcación no existe." "ERROR: El partido político no existe." "ERROR: La candidatura no existe." "ERROR: La candidatura no cumple con el mínimo de candidatos necesarios para poder participar en la convocatoria." "ERROR: SQLERRM"

Procedimiento Almacenado	CIERRE_CANDIDATURAS
Descripción	Se validan todas las candidaturas presentadas en la demarcación, aprobándose o rechazándose en función de si cumplen con los requisitos.
Condiciones	La convocatoria debe de existir en base de datos. La demarcación debe de existir en base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pldconvocatoria: VARCHAR(20) plddemarcacion: NUMBER(15)
Parámetros de salida	"OK", si el proceso finaliza correctamente. "ERROR: El código de la convocatoria no puede ser nulo." "ERROR: El código de la demarcación no puede ser nulo." "ERROR: La convocatoria no existe." "ERROR: La demarcación no existe." "ERROR: SQLERRM"

Procedimiento Almacenado	CONFIRMAR_NOTIFICACION_MESA
Descripción	Se confirma que la persona notificada para su participación en la mesa electoral ha recepcionado dicha notificación.
Condiciones	La convocatoria debe de existir en base de datos. La persona debe de existir en el censo de la base de datos. La persona debe de haber sido elegida en el sorteo de la mesa.

	Se debe de haber generado una notificación a la persona sobre su participación en la formación de la mesa electoral. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pldconvocatoria: VARCHAR(20) pldpadron: VARCHAR(20)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la convocatoria no puede ser nulo.” “ERROR: El código del padrón no puede ser nulo.” “ERROR: La convocatoria no existe.” “ERROR: La persona no existe en el censo electoral.” “ERROR: La persona no ha sido designada como miembro de la mesa.” “ERROR: La persona no ha sido notificada como miembro de la mesa.” “ERROR: SQLERRM”

Procedimiento Almacenado	ALEGA_ASISTENCIA_MESA
Descripción	La persona designada para formar parte de una mesa electoral presenta una alegación para eximirse de la obligación de su participación en dicha formación.
Condiciones	La convocatoria debe de existir en base de datos. La persona debe de existir en el censo de la base de datos. La persona debe de haber sido elegida en el sorteo de la mesa. Se debe de haber generado una notificación a la persona sobre su participación en la formación de la mesa electoral. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pldconvocatoria: VARCHAR(20) pldpadron: VARCHAR(20)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la convocatoria no puede ser nulo.” “ERROR: El código del padrón no puede ser nulo.” “ERROR: La convocatoria no existe.” “ERROR: La persona no existe en el censo electoral.” “ERROR: La persona no ha sido designada como miembro de la mesa.” “ERROR: La persona no ha sido notificada como miembro de la mesa.” “ERROR: SQLERRM”

Procedimiento Almacenado	CIERRE_MESA
Descripción	Se realiza el cierre de la mesa electoral para que no se pueda realizar ninguna votación más.
Condiciones	La demarcación debe de existir en base de datos. El municipio debe de existir en base de datos. La mesa debe de existir en base de datos. La mesa debe de encontrarse abierta. La convocatoria debe de encontrarse abierta. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	plddemarcacion: NUMBER(15) pldmunicipio: NUMBER(3) pldmesa: NUMBER(9)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la demarcación no puede ser nulo.” “ERROR: El código del municipio no puede ser nulo.” “ERROR: El código de la mesa no puede ser nulo.” “ERROR: La demarcación no existe.” “ERROR: El municipio no existe.” “ERROR: La mesa no existe.” “ERROR: La mesa no se encuentra ABIERTA.” “ERROR: La convocatoria no se encuentra ABIERTA.” “ERROR: SQLERRM”

Procedimiento Almacenado	FINALIZAR_MESA
Descripción	Se finaliza la mesa una vez que se ha realizado el recuento correspondiente y se han asignado a sus correspondientes partidos

	políticos.
Condiciones	La demarcación debe de existir en base de datos. El municipio debe de existir en base de datos. La mesa debe de existir en base de datos. La mesa debe de encontrarse cerrada. La convocatoria debe de encontrarse cerrada. El número de votos del recuento deben de coincidir con los emitidos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	plddemarcacion: NUMBER(15) pldmunicipio: NUMBER(3) pldemesa: NUMBER(9)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la demarcación no puede ser nulo.” “ERROR: El código del municipio no puede ser nulo.” “ERROR: El código de la mesa no puede ser nulo.” “ERROR: La demarcación no existe.” “ERROR: El municipio no existe.” “ERROR: La mesa no existe.” “ERROR: La mesa no se encuentra CERRADA.” “ERROR: La convocatoria no se encuentra CERRADA.” “ERROR: El número de votos contados es inferior al número de votos emitidos.” “ERROR: El número de votos contados es mayor al número de votos emitidos.” “ERROR: SQLERRM”

Procedimiento Almacenado	ABRIR_MESA
Descripción	Se realiza la apertura de la mesa electoral para que pueda llevarse a cabo la votación.
Condiciones	La demarcación debe de existir en base de datos. El municipio debe de existir en base de datos. La mesa debe de existir en base de datos. La mesa debe de encontrarse cerrada. La convocatoria debe de encontrarse abierta. Deben de encontrarse presentes los miembros que formarán la mesa. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	plddemarcacion: NUMBER(15) pldmunicipio: NUMBER(3) pldemesa: NUMBER(9)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la demarcación no puede ser nulo.” “ERROR: El código del municipio no puede ser nulo.” “ERROR: El código de la mesa no puede ser nulo.” “ERROR: La demarcación no existe.” “ERROR: El municipio no existe.” “ERROR: La mesa no existe.” “ERROR: La mesa no se encuentra CERRADA.” “ERROR: La convocatoria no se encuentra ABIERTA.” “ERROR: La mesa no tiene los miembros necesarios para poder abrirla.” “ERROR: SQLERRM”

Procedimiento Almacenado	PRESENTACION_MIEMBRO_MESA
Descripción	Se registran los miembros de la mesa que se presentan el día de la votación en el centro electoral.
Condiciones	La convocatoria debe de existir en base de datos. La persona debe de existir en el censo de la base de datos. La mesa debe de existir en base de datos. La persona debe de haber sido elegida en el sorteo de la mesa. La persona debe de haber sido notificada de su elección para la mesa. La persona ya se encuentra registrada en la mesa como presentada. La mesa ya tiene registrados todos los miembros necesarios. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pldconvocatoria: VARCHAR(20)

	pldpadron: VARCHAR(20) pldmesa: NUMBER(9)
Parámetros de salida	<p>“OK”, si el proceso finaliza correctamente.</p> <p>“ERROR: El código de la convocatoria no puede ser nulo.”</p> <p>“ERROR: El código del padrón no puede ser nulo.”</p> <p>“ERROR: El código de la mesa no puede ser nulo.”</p> <p>“ERROR: La convocatoria no existe.”</p> <p>“ERROR: La persona no existe en el censo electoral.”</p> <p>“ERROR: La mesa no existe.”</p> <p>“ERROR: La persona no ha sido designada como miembro de la mesa.”</p> <p>“ERROR: La persona no ha sido notificada como miembro de la mesa.”</p> <p>“ERROR: La persona ya se encuentra registrada como presentado.”</p> <p>“ERROR: La mesa ya tiene registrados los miembros necesarios.”</p> <p>“ERROR: SQLERRM”</p>

Procedimiento Almacenado	CIERRE_CONVOCATORIA
Descripción	Se realiza el cierre de la convocatoria para no permitir el registro de más votos en la convocatoria.
Condiciones	<p>La convocatoria debe de existir en base de datos.</p> <p>La convocatoria debe de estar en estado abierta.</p> <p>Las mesas de la convocatoria deben de estar en estado cerrada.</p> <p>Los parámetros de entrada no pueden ser nulos.</p>
Parámetros de entrada	pldconvocatoria: VARCHAR(20)
Parámetros de salida	<p>“OK”, si el proceso finaliza correctamente.</p> <p>“ERROR: El código de la convocatoria no puede ser nulo.”</p> <p>“ERROR: La convocatoria no existe.”</p> <p>“ERROR: La convocatoria no se encuentra abierta.”</p> <p>“ERROR: La convocatoria no se puede cerrar porque existen mesas abiertas.”</p> <p>“ERROR: SQLERRM”</p>

Procedimiento Almacenado	FINALIZAR_CONVOCATORIA
Descripción	Se finaliza la convocatoria una vez que se han completado todos los procesos. Los datos ya no pueden ser modificados.
Condiciones	<p>La convocatoria debe de existir en base de datos.</p> <p>La convocatoria debe de estar en estado cerrada.</p> <p>Las mesas de la convocatoria deben de estar en estado finalizada.</p> <p>Deben de estar realizados todos los escrutinios de todas las demarcaciones que engloba la convocatoria.</p> <p>Los parámetros de entrada no pueden ser nulos.</p>
Parámetros de entrada	pldconvocatoria: VARCHAR(20)
Parámetros de salida	<p>“OK”, si el proceso finaliza correctamente.</p> <p>“ERROR: El código de la convocatoria no puede ser nulo.”</p> <p>“ERROR: La convocatoria no existe.”</p> <p>“ERROR: La convocatoria no se encuentra cerrada.”</p> <p>“ERROR: La convocatoria no se puede finalizar porque existen mesas sin finalizar.”</p> <p>“ERROR: Quedan escrutinios pendientes de realizar en la convocatoria.”</p> <p>“ERROR: SQLERRM”</p>

Procedimiento Almacenado	REALIZAR_ESCRUTINIO
Descripción	Se realiza el escrutinio de los recuentos obtenidos de cada una de las mesas y se asignan los escaños a cada uno de los partidos en función de los votos obtenidos. El procedimiento está preparado para emplear los métodos de D'hondt y Sainte-Lague ⁸ .

⁸ Método Sainte-Laguë - Wikipedia, la enciclopedia libre. (s. f.). Recuperado 23 de mayo de 2021, de https://es.wikipedia.org/wiki/Método_Sainte-Laguë

Condiciones	La convocatoria debe de existir en base de datos. La demarcación debe de existir en base de datos. Las mesas electorales deben de estar en estado finalizada. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pldconvocatoria: VARCHAR(20) plddemarcacion: NUMBER(15)
Parámetros de salida	"OK", si el proceso finaliza correctamente. "ERROR: El código de la convocatoria no puede ser nulo." "ERROR: El código de la demarcación no puede ser nulo." "ERROR: La convocatoria no existe." "ERROR: La demarcación no existe." "ERROR: Existen mesas sin finalizar en la demarcación." "ERROR: SQLERRM"

Tabla 21: Definición de los procedimientos del paquete PK_PRC

2.6.3 Procedimientos paquete PK_DWH

Procedimiento Almacenado	DWH_PARTICIPACION_CONVOCATORIA
Descripción	Registra el proceso electoral que más participación ha tenido de todos los registrados en el sistema.
Condiciones	La convocatoria debe de existir en base de datos. La participación debe de ser superior a la registrada en base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pldconvocatoria: VARCHAR(20)
Parámetros de salida	"OK", si el proceso finaliza correctamente. "ERROR: El código de la convocatoria no puede ser nulo." "ERROR: La convocatoria no existe." "ERROR: La participación de la convocatoria es inferior a la ya registrada." "ERROR: SQLERRM"

Procedimiento Almacenado	DWH_PARTICIPACION_MUNICIPIO
Descripción	Registra el municipio que más participación ha tenido de todas las convocatorias ya celebradas.
Condiciones	La convocatoria debe de existir en base de datos. La participación debe de ser superior a la registrada en base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pldconvocatoria: VARCHAR(20)
Parámetros de salida	"OK", si el proceso finaliza correctamente. "ERROR: El código de la convocatoria no puede ser nulo." "ERROR: La convocatoria no existe." "ERROR: La participación municipal es inferior a la ya registrada." "ERROR: SQLERRM"

Procedimiento Almacenado	DWH_PARTICIPACION_TELEMATICA
Descripción	Registra el porcentaje de votos telemáticos emitidos de todas las convocatorias ya celebradas agrupadas por año.
Condiciones	La convocatoria debe de existir en base de datos. La participación debe de ser superior a la registrada en base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pldconvocatoria: VARCHAR(20)
Parámetros de salida	"OK", si el proceso finaliza correctamente. "ERROR: El código de la convocatoria no puede ser nulo." "ERROR: La convocatoria no existe." "ERROR: La participación telemática es inferior a la ya registrada." "ERROR: SQLERRM"

Procedimiento Almacenado	DWH_VARIACION_VOTOS
Descripción	Registra la variación del voto (incremento o decremento) con respecto al anterior proceso electoral celebrado.
Condiciones	La convocatoria debe de existir en base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pldconvocatoria: VARCHAR(20)
Parámetros de salida	"OK", si el proceso finaliza correctamente. "ERROR: El código de la convocatoria no puede ser nulo." "ERROR: La convocatoria no existe." "ERROR: SQLERRM"

Procedimiento Almacenado	DWH_FIJOS_MESA
Descripción	Registra el total de personas que han formado parte de una mesa electoral en más de 2 ocasiones.
Condiciones	
Parámetros de entrada	NINGUNO
Parámetros de salida	"OK", si el proceso finaliza correctamente. "ERROR: SQLERRM"

Procedimiento Almacenado	DWH_ALEGACION_MESA
Descripción	Registra el porcentaje de personas que han presentado alguna alegación a formar parte de una mesa en las próximas elecciones disponibles.
Condiciones	La convocatoria debe de existir en base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pldconvocatoria: VARCHAR(20)
Parámetros de salida	"OK", si el proceso finaliza correctamente. "ERROR: El código de la convocatoria no puede ser nulo." "ERROR: La convocatoria no existe." "ERROR: SQLERRM"

Procedimiento Almacenado	DWH_PARTIDOS_PRESENTACION
Descripción	Registra los partidos políticos que se han presentado en todas las demarcaciones de todas las convocatorias celebradas en los últimos 10 años.
Condiciones	
Parámetros de entrada	NINGUNO
Parámetros de salida	"OK", si el proceso finaliza correctamente. "ERROR: SQLERRM"

Procedimiento Almacenado	DWH_SORTEO_MESAS
Descripción	Registra el porcentaje de municipios que todavía no han realizado el sorteo de asignación de mesas electorales.
Condiciones	
Parámetros de entrada	NINGUNO
Parámetros de salida	"OK", si el proceso finaliza correctamente. "ERROR: SQLERRM"

Procedimiento Almacenado	DWH_MESAS_MENOS_VOTOS
Descripción	Registra las 20 mesas que menos votos han recibido en cualquier convocatoria celebrada.
Condiciones	
Parámetros de entrada	NINGUNO
Parámetros de salida	"OK", si el proceso finaliza correctamente. "ERROR: SQLERRM"

Procedimiento Almacenado	DWH_NOTIFICACION_MESA
Descripción	Registra el porcentaje de personas que han sido designadas para una mesa electoral y no han sido posible entregarle la notificación.
Condiciones	La convocatoria debe de existir en base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pldconvocatoria: VARCHAR(20)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la convocatoria no puede ser nulo.” “ERROR: La convocatoria no existe.” “ERROR: SQLERRM”

Procedimiento Almacenado	DWH_MAX_CANDIDATOS
Descripción	Registra la convocatoria y el número de candidatos de la candidatura presentada con más integrantes.
Condiciones	La convocatoria debe de existir en base de datos. El número de candidatos debe de ser superior al registrado en base de datos. Los parámetros de entrada no pueden ser nulos.
Parámetros de entrada	pldconvocatoria: VARCHAR(20)
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: El código de la convocatoria no puede ser nulo.” “ERROR: La convocatoria no existe.” “ERROR: El número de candidatos es inferior a los que ya se encuentran registrados.” “ERROR: SQLERRM”

Procedimiento Almacenado	DWH_REPARTO_ESCANOS
Descripción	Registra los escaños obtenidos por cada partido político en la última convocatoria celebrada.
Condiciones	La convocatoria no puede encontrarse previamente registrada.
Parámetros de entrada	NINGUNO
Parámetros de salida	“OK”, si el proceso finaliza correctamente. “ERROR: Los datos de la última convocatoria ya se encuentran registrados.” “ERROR: SQLERRM”

Tabla 22: Definición de los procedimientos del paquete PK_DWH

2.7 Ajustes de diseño

A lo largo de la elaboración de los procedimientos de la BBDD se han echado en falta algunas consideraciones que no se tuvieron en cuenta en el diseño conceptual y lógico. Los ajustes realizados en estos son los siguientes:

- Se modifican las agrupaciones ‘Estado’ y ‘EstadoCandidatura’ de entidades para añadir el valor ‘Anulada’.
- Se crea una nueva agrupación llamada ‘EstadoUsuario’ que contiene los valores ‘Habilitado’ y ‘Deshabilitado’.
- En Oracle no existe el tipo de datos Boolean, por lo que se sustituye por el tipo de dato CHAR estableciendo una restricción de integridad que solo permita los valores ‘S’ y ‘N’.
- Se crea una clave única en la tabla Recuento, para los campos Partido e Id_Mesa.

- Se añade el campo fechaInscripcion a la tabla partido_politico para poder tener un campo con el que parametrizar su participación o no en una convocatoria.
- Se añaden los campos sorteado y notificado a la tabla centro_votacion, para poder dar respuesta a las consultas 8 y 10 del datawarehouse.
- Se añade el campo alega a la tabla censo_electoral, para poder dar respuesta a la consulta 6 del datawarehouse.

Se modifican los diagramas del diseño conceptual y lógico para que recojan estos cambios.

2.8 Elaboración de pruebas

Para la elaboración de las pruebas del proyecto desarrollado, se han diseñado una serie de pruebas unitarias de los diferentes procedimientos almacenados para comprobar que ante todas las casuísticas posibles detectadas, estos responden con el resultado esperado. Para ello, se ha elaborado una serie de juego de datos previos que servirán como apoyo para la realización de estas.

2.8.1 Pruebas unitarias Pk_ABM

En primer lugar nos vamos a ocupar de la realización de las pruebas correspondientes a los procedimientos de alta, baja y modificación. Estos procedimientos corresponden a las entidades que han sido identificadas como más relevantes de la base de datos. Hemos implementado el script **“20_Precarga_Pruebas_Unitarias_Pk_ABM.sql”** que contendrá el juego de datos mínimo necesario para las pruebas que vamos a llevar a cabo. A continuación, mediante el script **“21_Pruebas_Unitarias_Pk_ABM”**, se ejecutarán dichas pruebas y donde por medio de la tabla adjunta se describe la prueba lanzada con el resultado obtenido, en el mismo orden en el que se llevan a cabo con el script.

Procedimientos_pk_ABM	Tarea	Resultado
ALTA_COMUNIDAD_AUTONOMA	Alta de una comunidad autónoma que no existe, cumpliendo las validaciones.	OK
	Error porque falta el código de la comunidad autónoma.	OK
	Error porque falta la descripción de la comunidad autónoma.	OK
	Error porque la comunidad autónoma ya existe.	OK
MODIF_COMUNIDAD_AUTONOMA	Error porque falta el código de la comunidad autónoma.	OK
	Error porque falta la descripción de la comunidad autónoma.	OK
	Error porque la comunidad no existe.	OK
	Error por la restricción UNIQUE de la descripción de la comunidad autónoma.	OK
	Modificación de una comunidad autónoma existente, cumpliendo las validaciones.	OK
BAJA_COMUNIDAD_AUTONOMA	Error porque falta el código de la comunidad autónoma.	OK

	Error porque la comunidad no existe.	OK
	Error porque la comunidad tiene dependencias de FK.	OK
	Baja de una comunidad autónoma existente, cumpliendo las validaciones.	OK
ALTA_PROVINCIA	Error porque falta el código de la comunidad autónoma.	OK
	Error porque falta el código de la provincia.	OK
	Error porque falta la descripción de la provincia.	OK
	Error porque la comunidad no existe.	OK
	Error porque la provincia ya existe.	OK
	Alta de una provincia que no existe, cumpliendo las validaciones.	OK
MODIFICACION_PROVINCIA	Error porque falta el código de la provincia.	OK
	Error porque falta la descripción de la provincia.	OK
	Error porque la provincia no existe.	OK
	Error por la restricción UNIQUE de la descripción de la provincia.	OK
	Modificación de una provincia existente, cumpliendo las validaciones.	OK
BAJA_PROVINCIA	Error porque falta el código de la provincia.	OK
	Error porque la provincia no existe.	OK
	Error porque la provincia tiene dependencias de FK.	OK
	Baja de una provincia existente, cumpliendo las validaciones.	OK
ALTA_MUNICIPIO	Error porque falta el código de la provincia.	OK
	Error porque falta el código del municipio.	OK
	Error porque falta la descripción del municipio.	OK
	Error porque la provincia no existe.	OK
	Error porque el municipio ya existe.	OK
	Alta de un municipio que no existe, cumpliendo las validaciones.	OK
MODIFICACION_MUNICIPIO	Error porque falta el código del municipio.	OK
	Error porque falta la descripción del municipio.	OK
	Error porque el municipio no existe.	OK
	Error por la restricción UNIQUE de la descripción del municipio.	OK
	Modificación de un municipio existente, cumpliendo las validaciones.	OK
BAJA_MUNICIPIO	Error porque falta el código del municipio.	OK
	Error porque el municipio no existe.	OK
	Error porque el municipio tiene dependencias de FK.	OK
	Baja de un municipio existente, cumpliendo las validaciones.	OK
ALTA_TITULACION	Error porque falta el código de la titulación.	OK
	Error porque falta la descripción de la titulación.	OK
	Error porque la titulación ya existe.	OK
	Alta de una titulación que no existe, cumpliendo las validaciones.	OK
MODIFICACION_TITULACION	Error porque falta el código de la titulación.	OK
	Error porque falta la descripción de la titulación.	OK
	Error porque la titulación no existe.	OK
	Error por la restricción UNIQUE de la descripción de la titulación.	OK
	Modificación de una titulación existente, cumpliendo las validaciones.	OK
BAJA_TITULACION	Error porque falta el código de la titulación.	OK
	Error porque la titulación no existe.	OK
	Error porque la titulación tiene dependencias de FK.	OK
	Baja de una titulación existente, cumpliendo las validaciones.	OK
ALTA_USUARIOS	Error porque falta el código del usuario.	OK
	Error porque falta la clave del usuario.	OK
	Error porque falta el nivel del usuario.	OK
	Error porque el valor del campo TIPO no es válido.	OK
	Error porque el usuario ya existe.	OK

	Alta de un usuario que no existe, cumpliendo las validaciones.	OK
MODIFICACION_USUARIOS	Error porque falta el código del usuario.	OK
	Error porque falta la clave del usuario.	OK
	Error porque falta el nivel del usuario.	OK
	Error porque falta el estado del usuario.	OK
	Error porque el valor del campo TIPO no es válido.	OK
	Error porque el valor del campo ESTADO no es válido.	OK
	Error porque el usuario no existe.	OK
	Modificación de un usuario existente, cumpliendo las validaciones.	OK
BAJA_USUARIOS	Error porque falta el código del usuario.	OK
	Error porque el usuario no existe.	OK
	Baja de un usuario existente, cumpliendo las validaciones.	OK
ALTA_PADRON_MUNICIPAL	Error porque falta el código del padrón.	OK
	Error porque falta el nombre de la persona.	OK
	Error porque falta el primer apellido de la persona.	OK
	Error porque falta el número de documento de la persona.	OK
	Error porque falta la fecha nacimiento de la persona.	OK
	Error porque falta la provincia nacimiento de la persona.	OK
	Error porque falta el municipio nacimiento de la persona.	OK
	Error porque falta el país de la persona.	OK
	Error porque falta el sexo de la persona.	OK
	Error porque falta la provincia domicilio de la persona.	OK
	Error porque falta el municipio domicilio de la persona.	OK
	Error porque falta la dirección domicilio de la persona.	OK
	Error porque falta el código de la titulación.	OK
	Error porque el valor del campo SEXO no es válido.	OK
	Error porque la provincia domicilio no existe.	OK
	Error porque el municipio domicilio no existe.	OK
	Error porque la titulación no existe.	OK
	Error porque la persona ya existe.	OK
	Alta de una persona en el padrón que no existe, cumpliendo las validaciones.	OK
	MODIFICACION_PADRON_MUNICIPAL	Error porque falta el código del padrón y el número de documento.
Error porque falta la provincia domicilio de la persona.		OK
Error porque falta el municipio domicilio de la persona.		OK
Error porque falta la dirección domicilio de la persona.		OK
Error porque falta el código de la titulación.		OK
Error porque la provincia domicilio no existe.		OK
Error porque el municipio domicilio no existe.		OK
Error porque la titulación no existe.		OK
Error porque la persona no existe.		OK
Modificación de una persona en el padrón existente, cumpliendo las validaciones.		OK
BAJA_PADRON_MUNICIPAL	Error porque falta el código del padrón y el número de documento.	OK
	Error porque la persona no existe.	OK
	Error porque la persona del padrón tiene dependencias de FK.	OK
	Baja de una persona en el padrón existente, cumpliendo las validaciones.	OK
ALTA_PARTIDO_POLITICO	Error porque falta el código del partido político.	OK
	Error porque falta las siglas del partido político.	OK
	Error porque falta el nombre del partido político.	OK
	Error porque falta la dirección del partido político.	OK
	Error porque falta la fecha de inscripción del partido	OK

	político.	
	Error porque el partido político ya existe.	OK
	Alta de un partido político que no existe, cumpliendo las validaciones.	OK
MODIFICACION_PARTIDO_POLITICO	Error porque falta el código del partido político.	OK
	Error porque falta las siglas del partido político.	OK
	Error porque falta el nombre del partido político.	OK
	Error porque falta la dirección del partido político.	OK
	Error porque falta la fecha de inscripción del partido político.	OK
	Error porque el partido político no existe.	OK
	Error por la restricción UNIQUE de las siglas o nombre del partido político.	OK
	Modificación de un partido político existente, cumpliendo las validaciones.	OK
BAJA_PARTIDO_POLITICO	Error porque falta el código del partido político.	OK
	Error porque el partido político no existe.	OK
	Error porque la persona del padrón tiene dependencias de FK.	OK
	Baja de un partido político existente, cumpliendo las validaciones.	OK
ALTA_PARAMETROS	Error porque falta el código del parámetro.	OK
	Error porque falta el tipo del parámetro.	OK
	Error porque falta el campo del parámetro.	OK
	Error porque falta el valor del parámetro.	OK
	Error porque el valor del campo TIPO no es válido.	OK
	Error porque el parámetro ya existe.	OK
	Alta de un parámetro que no existe, cumpliendo las validaciones.	OK
MODIFICACION_PARAMETROS	Error porque falta el código del parámetro.	OK
	Error porque falta el valor del parámetro.	OK
	Error porque el parámetro no existe.	OK
	Modificación de un parámetro existente, cumpliendo las validaciones.	OK
BAJA_PARAMETROS	Error porque falta el código del parámetro.	OK
	Error porque el parámetro no existe.	OK
	Error porque el parámetro tiene dependencias de FK.	OK
	Baja de un parámetro existente, cumpliendo las validaciones.	OK
ALTA_CONVOCATORIA	Error porque falta el código de la convocatoria.	OK
	Error porque falta la descripción de la convocatoria.	OK
	Error porque falta la fecha publicación de la convocatoria.	OK
	Error porque falta la fecha inicio de la convocatoria.	OK
	Error porque falta la fecha finalización de la convocatoria.	OK
	Error porque falta el tipo de la convocatoria.	OK
	Error porque falta el número de escaños de la convocatoria.	OK
	Error porque el valor del campo TIPO no es válido.	OK
	Error porque el valor del campo TIPO es AUTONOMICA y el campo AUTONOMIA es nulo.	OK
	Error porque el valor del campo AUTONOMIA no es válido.	OK
	Error porque la convocatoria ya existe.	OK
	Alta de una convocatoria que no existe, cumpliendo las validaciones.	OK
MODIFICACION_CONVOCATORIA	Error porque falta el código de la convocatoria.	OK
	Error porque falta la descripción de la convocatoria.	OK
	Error porque falta la fecha publicación de la convocatoria.	OK
	Error porque falta la fecha inicio de la convocatoria.	OK
	Error porque falta la fecha finalización de la convocatoria.	OK
	Error porque falta el tipo de la convocatoria.	OK

	Error porque falta el número de escaños de la convocatoria.	OK
	Error porque el valor del campo TIPO no es válido.	OK
	Error porque el valor del campo ESTADO no es válido.	OK
	Error porque la convocatoria no existe.	OK
	Error porque el valor del campo TIPO es AUTONOMICA y el campo AUTONOMIA es nulo.	OK
	Error porque el valor del campo AUTONOMIA no es válido.	OK
	Modificación de una convocatoria existente, cumpliendo las validaciones.	OK
BAJA_CONVOCATORIA	Error porque falta el código de la convocatoria.	OK
	Error porque la convocatoria no existe.	OK
	Baja de una convocatoria existente, cumpliendo las validaciones.	OK
ALTA_CONVOCATORIA_PARAMETROS	Error porque falta el código de la convocatoria.	OK
	Error porque falta el código del parámetro.	OK
	Error porque la convocatoria no existe.	OK
	Error porque el parámetro no existe.	OK
	Error porque el parámetro-convocatoria ya existe.	OK
	Alta de un parámetro-convocatoria que no existe, cumpliendo las validaciones.	OK
BAJA_CONVOCATORIA_PARAMETROS	Error porque falta el código de la convocatoria.	OK
	Error porque falta el código del parámetro.	OK
	Error porque el parámetro-convocatoria no existe.	OK
	Baja de un parámetro-convocatoria existente, cumpliendo las validaciones.	OK
ALTA_DEMARCACION	Error porque falta el número de escaños.	OK
	Error porque falta el número de votantes.	OK
	Error porque falta el código de la convocatoria.	OK
	Error porque el valor del campo ESCANOS y/o VOTANTES no es válido.	OK
	Alta de una demarcación, cumpliendo las validaciones.	OK
MODIFICACION_DEMARCACION	Error porque falta el código de la demarcación.	OK
	Error porque falta el número de escaños.	OK
	Error porque falta el número de votantes.	OK
	Error porque falta el código de la convocatoria.	OK
	Error porque el valor del campo ESCANOS y/o VOTANTES no es válido.	OK
	Error porque la demarcación no existe.	OK
	Modificación de una demarcación existente, cumpliendo las validaciones.	OK
BAJA_DEMARCACION	Error porque falta el código de la demarcación.	OK
	Error porque falta el código de la convocatoria.	OK
	Error porque la demarcación no existe.	OK
	Error porque la convocatoria tiene dependencias de FK.	OK
	Baja de una demarcación existente, cumpliendo las validaciones.	OK
ALTA_CENTRO_VOTACION	Error porque falta el código de la demarcación.	OK
	Error porque falta el código del municipio.	OK
	Error porque falta el distrito.	OK
	Error porque falta la sección.	OK
	Error porque falta la dirección del centro.	OK
	Error porque falta el código postal.	OK
	Error porque el centro de votación ya existe.	OK
	Alta de un centro de votación, cumpliendo las validaciones.	OK
MODIFICACION_CENTRO_VOTACION	Error porque falta el código de la demarcación.	OK
	Error porque falta el código del municipio.	OK
	Error porque falta el distrito.	OK
	Error porque falta la sección.	OK
	Error porque falta la dirección del centro.	OK

	Error porque falta el código postal.	OK
	Error porque falta el identificador de sorteado.	OK
	Error porque falta el identificador de notificado.	OK
	Error porque el valor del campo SORTEADO y/o NOTIFICADO no es válido.	OK
	Error porque el centro de votación no existe.	OK
	Modificación de un centro de votación existente, cumpliendo las validaciones.	OK
BAJA_CENTRO_VOTACION	Error porque falta el código de la demarcación.	OK
	Error porque falta el código del municipio.	OK
	Error porque el centro de votación no existe.	OK
	Error porque el centro de votación tiene dependencias de FK.	OK
	Baja de un centro de votación existente, cumpliendo las validaciones.	OK
ALTA_MESA	Error porque falta la descripción de la mesa.	OK
	Error porque falta el código de la demarcación.	OK
	Error porque falta el código del municipio.	OK
	Alta de una mesa, cumpliendo las validaciones.	OK
MODIFICACION_MESA	Error porque falta el código de la mesa.	OK
	Error porque falta la descripción de la mesa.	OK
	Error porque falta el estado de la mesa.	OK
	Error porque el valor del campo ESTADO no es válido.	OK
	Error porque la mesa no existe.	OK
	Modificación de una mesa existente, cumpliendo las validaciones.	OK
BAJA_MESA	Error porque falta el código de la mesa.	OK
	Error porque la mesa no existe.	OK
	Baja de una mesa existente, cumpliendo las validaciones.	OK
ALTA_CENSO_ELECTORAL	Error porque falta el código de la convocatoria.	OK
	Error porque falta el código del padrón.	OK
	Error porque falta la fecha inscripción.	OK
	Error porque falta el código de la mesa.	OK
	Error porque la persona ya existe en el censo.	OK
	Alta de una persona en el censo, cumpliendo las validaciones.	OK
MODIFICACION_CENSO_ELECTORAL	Error porque falta el código de la convocatoria.	OK
	Error porque falta el código del padrón.	OK
	Error porque falta el indicador de si pertenece a una mesa.	OK
	Error porque el valor del campo ES_MESA es 'S' y el campo TIPO es nulo.	OK
	Error porque el valor del campo ES_MESA es 'S' y el campo NOTIFICADO es nulo.	OK
	Error porque el valor del campo ES_MESA es 'S' y el campo ALEGA es nulo.	OK
	Error porque el valor del campo ES_MESA es 'S' y el campo PRESENTADO es nulo.	OK
	Error porque el valor del campo ES_MESA, TIPO, NOTIFICADO, ALEGA y/o PRESENTADO no es válido.	OK
	Error porque la persona no existe en el censo.	OK
	Modificación de una persona existente en el censo, cumpliendo las validaciones.	OK
BAJA_CENSO_ELECTORAL	Error porque falta el código de la convocatoria.	OK
	Error porque falta el código del padrón.	OK
	Error porque la persona no existe en el censo.	OK
	Error porque la persona tiene dependencias de FK.	OK
	Baja de una persona existente en el censo, cumpliendo las validaciones.	OK
ALTA_CANDIDATURA	Error porque falta el código de la demarcación.	OK
	Error porque falta el código del partido político.	OK
	Error porque la candidatura ya existe.	OK

	Alta de una candidatura que no existe, cumpliendo las validaciones.	OK
MODIFICACION_CANDIDATURA	Error porque falta el código de la demarcación.	OK
	Error porque falta el código del partido político.	OK
	Error porque falta el estado de la candidatura.	OK
	Error porque el valor del campo ESTADO no es válido.	OK
	Error porque la candidatura no existe.	OK
	Modificación de una candidatura existente, cumpliendo las validaciones.	OK
BAJA_CANDIDATURA	Error porque falta el código de la demarcación.	OK
	Error porque falta el código del partido político.	OK
	Error porque la candidatura no existe.	OK
	Baja de una candidatura existente, cumpliendo las validaciones.	OK
ALTA_CANDIDATO	Error porque falta el código de la convocatoria.	OK
	Error porque falta el código del padrón.	OK
	Error porque falta el código de la demarcación.	OK
	Error porque falta el código del partido político.	OK
	Error porque falta el orden en la candidatura.	OK
	Error porque el candidato ya existe.	OK
	Alta de un candidato que no existe, cumpliendo las validaciones.	OK
MODIFICACION_CANDIDATO	Error porque falta el código de la convocatoria.	OK
	Error porque falta el código del padrón.	OK
	Error porque falta el código de la demarcación.	OK
	Error porque falta el código del partido político.	OK
	Error porque falta el orden en la candidatura.	OK
	Error porque el candidato no existe en el censo.	OK
	Modificación de un candidato existente, cumpliendo las validaciones.	OK
BAJA_CANDIDATO	Error porque falta el código de la convocatoria.	OK
	Error porque falta el código del padrón.	OK
	Error porque falta el código de la demarcación.	OK
	Error porque falta el código del partido político.	OK
	Error porque el candidato no existe.	OK
	Baja de un candidato existente, cumpliendo las validaciones.	OK

Tabla 23: Definición de pruebas unitarias pk_ABM

2.8.2 Pruebas unitarias Pk_PRC

A continuación, se procede a la ejecución de los procedimientos asociados a la lógica y funcionalidad del programa, donde se debe de dar cumplimiento a los requisitos funcionales y a las restricciones definidas. Para ello, se implementa el script “**31_Precarga_Pruebas_Unitarias_Pk_PRC.sql**” que contendrá el juego de datos mínimo necesario para las pruebas que van a ser ejecutadas. En la tabla adjunta se definen todas las funcionalidades a probar sobre dichos procedimientos, así como el resultado obtenido en cada uno de ellos. Para la ejecución de dichas pruebas se implementa el script “**21_Pruebas_Unitarias_Pk_ABM**”, que contienen todas están pruebas en el orden en el que se detallan en la tabla.

Procedimientos_pk_PRC	Tarea	Resultado
GENERAR_DEMARCACION	Alta de una demarcación, cumpliendo las validaciones.	OK
	Error porque falta el código de la convocatoria.	OK
	Error porque la convocatoria no existe.	OK
	Error porque la convocatoria no se encuentra Abierta.	OK

	Error porque la demarcación ya existe.	OK
GENERAR_CENSO_ELECTORAL	Alta de un censo electoral, cumpliendo las validaciones.	OK
	Error porque falta el código de la convocatoria.	OK
	Error porque la convocatoria no existe.	OK
	Error porque la convocatoria no se encuentra Abierta.	OK
	Error porque el censo electoral ya existe.	OK
VALIDAR_CANDIDATURA	Error porque falta el código de la convocatoria.	OK
	Error porque falta el código de la demarcación.	OK
	Error porque falta el código del partido político.	OK
	Error porque la convocatoria no existe.	OK
	Error porque la demarcación no existe.	OK
	Error porque el partido político no existe.	OK
	Error porque la candidatura no existe.	OK
	Error porque la candidatura no cumple con el mínimo de candidatos necesarios.	OK
	Se valida la candidatura, cumpliendo las validaciones.	OK
CIERRE_CANDIDATURAS	Error porque falta el código de la convocatoria.	OK
	Error porque falta el código de la demarcación.	OK
	Error porque la convocatoria no existe.	OK
	Error porque la demarcación no existe.	OK
	Se realiza el cierre de candidaturas y se actualizan sus estados, cumpliendo las validaciones.	OK
SORTEO_MIEMBROS_MESA	Error porque falta el código de la convocatoria.	OK
	Error porque falta el código de la demarcación.	OK
	Error porque la convocatoria no existe.	OK
	Error porque la demarcación no existe.	OK
	Error porque existen candidaturas pendientes de validar.	OK
	Se realiza sorteo sobre el censo para formar las mesas, cumpliendo las validaciones.	OK
ENVIAR_NOTIFICACION_MESA	Error porque el sorteo ya se ha realizado.	OK
	Error porque falta el código de la convocatoria.	OK
	Error porque falta el código del municipio.	OK
	Error porque la convocatoria no existe.	OK
	Error porque el sorteo no se encuentra realizado.	OK
	Se realiza la notificación del sorteo de mesas al censo electoral, cumpliendo las validaciones.	OK
CONFIRMAR_NOTIFICACION_MESA	Error porque la notificación ya se ha realizado.	OK
	Error porque falta el código de la convocatoria.	OK
	Error porque falta el código del padrón municipal.	OK
	Error porque la convocatoria no existe.	OK
	Error porque la persona no existe.	OK
	Error porque la persona no ha sido designada como miembro de la mesa.	OK
	Error porque la persona no ha sido notificada como miembro de la mesa.	OK
Se realiza la confirmación de la notificación como miembro de la mesa, cumpliendo las validaciones.	OK	
ALEGA_ASISTENCIA_MESA	Error porque falta el código de la convocatoria.	OK
	Error porque falta el código del padrón municipal.	OK
	Error porque la convocatoria no existe.	OK
	Error porque la persona no existe.	OK
	Error porque la persona no ha sido designada como miembro de la mesa.	OK
	Error porque la persona no ha sido notificada como miembro de la mesa.	OK
	Se realiza la alegación de la notificación como miembro de la mesa, cumpliendo las validaciones.	OK
PRESENTACION_MIEMBRO_MESA	Error porque falta el código de la convocatoria.	OK
	Error porque falta el código del padrón municipal.	OK
	Error porque falta el código de la mesa.	OK
	Error porque la convocatoria no existe.	OK
	Error porque la persona no existe.	OK
	Error porque la mesa no existe.	OK

	Error porque la persona no ha sido designada como miembro de la mesa.	OK
	Error porque la persona no ha sido notificada como miembro de la mesa.	OK
	Se realiza la confirmación de asistencia a la mesa, cumpliendo las validaciones.	OK
	Error porque la presentación ya existe.	OK
	Error porque la mesa ya contiene todos los miembros.	OK
ABRIR_MESA	Error porque falta el código de la demarcación.	OK
	Error porque falta el código del municipio.	OK
	Error porque falta el código de la mesa.	OK
	Error porque la demarcación no existe.	OK
	Error porque el municipio no existe.	OK
	Error porque la mesa no existe.	OK
	Se realiza la apertura de la mesa, cumpliendo las validaciones.	OK
	Error porque la mesa no se encuentra Cerrada.	OK
	Error porque la mesa no cumple con los miembros necesarios.	OK
	Error porque la convocatoria no se encuentra Abierta.	OK
ALTA_VOTACION	Error porque falta el tipo de voto.	OK
	Error porque el valor del campo TIPO no es válido.	OK
	Error porque el valor del campo TIPO es TELEMATICO y el campo CERTIFICADO es nulo.	OK
	Error porque falta el código de la mesa.	OK
	Error porque la mesa no existe.	OK
	Error porque la mesa no se encuentra Abierta.	OK
	Alta de un voto en una mesa, cumpliendo las validaciones.	OK
	Error porque en la mesa ya se han registrado todos los votos posibles.	OK
CIERRE_MESA	Error porque falta el código de la demarcación.	OK
	Error porque falta el código del municipio.	OK
	Error porque falta el código de la mesa.	OK
	Error porque la demarcación no existe.	OK
	Error porque el municipio no existe.	OK
	Error porque la mesa no existe.	OK
	Error porque la mesa no se encuentra Abierta.	OK
	Se realiza el cierre de la mesa, cumpliendo las validaciones.	OK
	Error porque la convocatoria no se encuentra Abierta.	OK
ALTA_RECUESTO	Error porque falta el código del partido político.	OK
	Error porque falta el número de votos.	OK
	Error porque falta el código de la mesa.	OK
	Error porque la mesa no existe.	OK
	Error porque el partido político no existe.	OK
	Error porque el partido político no tiene representación.	OK
	Error porque el número de votos es mayor que el número de votantes.	OK
	Error porque la mesa no se encuentra Cerrada.	OK
	Alta de un recuento en una mesa, cumpliendo las validaciones.	OK
	Error porque el recuento ya existe.	OK
MODIFICACION_RECUESTO	Error porque falta el código del partido político.	OK
	Error porque falta el número de votos.	OK
	Error porque falta el código de la mesa.	OK
	Error porque el recuento no existe.	OK
	Error porque la mesa no se encuentra Cerrada.	OK
	Error porque el número de votos es mayor que el número de votantes.	OK
	Modificación de un recuento existente, cumpliendo las validaciones.	OK
BAJA_RECUESTO	Error porque falta el código del partido político.	OK
	Error porque falta el código de la mesa.	OK

	Error porque el recuento no existe.	OK
	Error porque la mesa se encuentra Finalizada.	OK
	Baja de un recuento existente, cumpliendo las validaciones.	OK
CIERRE_CONVOCATORIA	Error porque falta el código de la convocatoria.	OK
	Error porque la convocatoria no existe.	OK
	Error porque la convocatoria no se encuentra Abierta.	OK
	Error porque existen mesas abiertas.	OK
	Se realiza el cierre de una convocatoria, cumpliendo las validaciones.	OK
FINALIZAR_MESA	Error porque falta el código de la demarcación.	OK
	Error porque falta el código del municipio.	OK
	Error porque falta el código de la mesa.	OK
	Error porque la demarcación no existe.	OK
	Error porque el municipio no existe.	OK
	Error porque la mesa no existe.	OK
	Error porque la convocatoria no se encuentra Cerrada.	OK
	Error porque la mesa no se encuentra Cerrada.	OK
	Error porque el número de votos es inferior a los emitidos.	OK
	Error porque el número de votos es mayor a los emitidos.	OK
	Se realiza la finalización de la mesa, cumpliendo las validaciones.	OK
REALIZAR_ESCRUTINIO	Error porque falta el código de la convocatoria.	OK
	Error porque falta el código de la demarcación.	OK
	Error porque la convocatoria no existe.	OK
	Error porque la demarcación no existe.	OK
	Error porque existen mesas sin finalizar.	OK
	Se realiza el escrutinio, cumpliendo las validaciones.	OK
FINALIZAR_CONVOCATORIA	Error porque falta el código de la convocatoria.	OK
	Error porque la convocatoria no existe.	OK
	Error porque la convocatoria no se encuentra Cerrada.	OK
	Error porque existen mesas sin finalizar.	OK
	Error porque quedan escrutinios pendientes de realizar.	OK
	Se realiza la finalización de una convocatoria, cumpliendo las validaciones.	OK

Tabla 24: Definición de pruebas unitarias pk_PRC

2.8.3 Pruebas unitarias Pk_DHW

Por último, se lleva a cabo la definición de las pruebas del módulo estadístico de la base de datos. El objetivo de estos procedimientos es dar respuesta a las consultas planteadas, por lo que las pruebas no son tan extensas como las anteriores. Implementamos el script **"41_Pruebas_Unitarias_Pk_DWH.sql"**, que contiene el juego de pruebas definidas para el módulo estadístico en el mismo orden que el definido en la tabla adjunta.

Procedimientos_pk_DWH	Tarea	Resultado
DWH_PARTICIPACION_CONVOCATORIA	Error porque falta el código de la convocatoria.	OK
	Error porque la convocatoria no existe.	OK
	La participación es inferior a la ya registrada.	OK
	Alta de los datos del proceso con más participación, cumpliendo las validaciones.	OK
DWH_PARTICIPACION_MUNICIPIO	Error porque falta el código de la convocatoria.	OK
	Error porque la convocatoria no existe.	OK

	Error porque la participación es inferior a la ya registrada.	OK
	Alta de los datos del municipio con más participación, cumpliendo las validaciones.	OK
DWH_PARTICIPACION_TELEMATICA	Error porque falta el código de la convocatoria.	OK
	Error porque la convocatoria no existe.	OK
	Error porque la participación es inferior a la ya registrada.	OK
	Alta de los datos del porcentaje de votos telemáticos, cumpliendo las validaciones.	OK
DWH_VARIACION_VOTOS	Error porque falta el código de la convocatoria.	OK
	Error porque la convocatoria no existe.	OK
	Alta de los datos de incremento o decremento en el total de votos, cumpliendo las validaciones.	OK
DWH_FIJOS_MESA	Alta de los datos de las personas que han formado parte de una mesa más de 2 veces, cumpliendo las validaciones.	OK
DWH_ALEGACION_MESA	Error porque falta el código de la convocatoria.	OK
	Error porque la convocatoria no existe.	OK
	Alta del porcentaje de personas que han presentado alegaciones a una mesa, cumpliendo las validaciones.	OK
DWH_PARTIDOS_PRESENTACION	Alta de los partidos que se han presentado a todas las demarcaciones en los últimos 10 años, cumpliendo las validaciones.	OK
DWH_SORTEO_MESAS	Alta del porcentaje de municipios que no han realizado el sorteo de los miembros de las mesas, cumpliendo las validaciones.	OK
DWH_MESAS_MENOS_VOTOS	Alta del Top 20 de mesas que han recibido menos votos, cumpliendo las validaciones.	OK
DWH_NOTIFICACION_MESA	Error porque falta el código de la convocatoria.	OK
	Error porque la convocatoria no existe.	OK
	Alta del porcentaje de personas que no se le ha localizado para entregar notificación, cumpliendo las validaciones.	OK
DWH_MAX_CANDIDATOS	Error porque falta el código de la convocatoria.	OK
	Error porque la convocatoria no existe.	OK
	Error porque el número de candidatos es inferior al ya registrado.	OK
	Alta del proceso que más candidatos presentados ha tenido, cumpliendo las validaciones.	OK
DWH_REPARTO_ESCANOS	Alta de los datos de los escaños obtenidos por los partidos en la última convocatoria, cumpliendo las validaciones.	OK

Tabla 25: Definición de pruebas unitarias pk_DWH

3. Conclusiones

3.1 Seguimiento de la planificación

3.1.1 PEC 1 – Plan de trabajo

El desarrollo de esta Pec se ha sucedido sin problemas, dando cumplimiento a los objetivos marcados en los requerimientos del enunciado del desarrollo del TFG. También se tiene en cuenta los objetivos del informe de evaluación a cumplir, así como los criterios de evaluación de este.

3.1.2 PEC 2

En el desarrollo de esta Pec el módulo del diseño conceptual es el que más problemas se han producidos. Se le han dado muchas vueltas y se ha tenido que invertir más horas en la ejecución de este módulo porque los resultados que se obtenían, o bien no satisfacían todos los requerimientos del enunciado o bien no cumplía con los requisitos de un diseño relacional. Finalmente se resuelve con un diseño óptimo para las necesidades descritas y se avanza al resto de módulos cumpliendo con los objetivos de entrega y planificación.

3.1.3 PEC 3

El comienzo de esta Pec, con el diseño físico del modelo obtenido previamente en la pec anterior, se resuelve sin contratiempos y cumpliendo el planning definido inicialmente para este apartado. De la misma manera ocurre con el diseño del data-warehouse, no requiere de mayor complejidad y se cumplen los plazos marcados inicialmente para cubrir este módulo del proyecto.

El apartado de la implementación de procedimientos es el que se ha visto seriamente afectado por la planificación marcada. A medida que se ha ido avanzando en el desarrollo de los procedimientos, se han ido detectando algunos ajustes del diseño conceptual y lógico que han sido tenidos que corregir o añadir a dicho modelo según el caso detectado, lo que ha provocado que se haya tenido que corregir parte de lo desarrollado anteriormente tal y como se indica en el apartado 2.9 del documento.

Además de esto, el desarrollo de procedimientos en PL/SQL no se ha desarrollado con la fluidez esperada, lo que ha provocado la necesidad de invertir más horas que las inicialmente definidas. Por este motivo, se pone en marcha la acción correctora A1R01 para poder cumplir con la planificación para esta entrega. A pesar de esto, no es suficiente y quedan por entregar una serie de procedimientos y triggers, sobre todo de la parte del data-warehouse, que se incluirán en la entrega final del TFG.

3.1.4 PEC 4 – Entrega Final

En esta pec comenzamos como finalizamos la anterior, esto es, desarrollando los procedimientos faltantes del proyecto que no se pudieron finalizar. Se hace un esfuerzo aumentando el número de horas de dedicación para afectar lo menos posible a la realización del resto de tareas que estaban programadas para esta entrega.

A la finalización se comienza con el desarrollo del plan de pruebas y la realización de estas. Gracias a que durante el desarrollo de los procedimientos se tenían bastantes pruebas ya definidas, se consigue reducir las horas de implicación definidas inicialmente, lo que favorece la continuación del planning definido a pesar del retraso en el módulo de procedimientos.

El resto de las tareas finalizan correctamente en tiempo y forma y se finaliza el proyecto sin ningún incidente reseñable.

A continuación, se adjunta el diagrama de Gantt recogiendo todas las afecciones sucedidas en el proyecto.

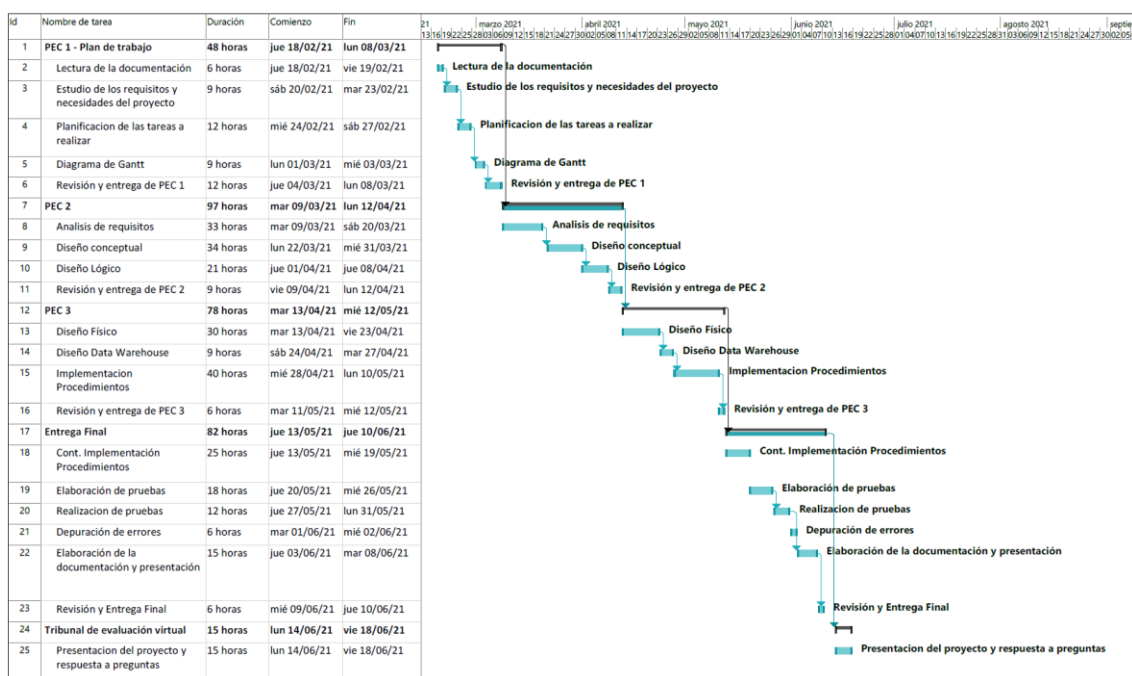


Ilustración 5: Diagrama de Gantt ajustado

3.2 Conclusiones

La dedicación a este proyecto me ha resultado de un mayor esfuerzo y dedicación del que inicialmente imaginaba que me iba a ocupar. A pesar de poseer experiencia en la gestión de proyectos, este a sido un muy interesante reto personal, que me ha enriquecido y me ha hecho ver todos los puntos de vista del ciclo del proyecto. A pesar del esfuerzo requerido, ha sido posible ir cumpliendo todas las etapas gracias a la planificación definida inicialmente y a los plazos de entrega, que servían como motivación adicional al esfuerzo a realizar.

La metodología empleada ha permitido realizar una planificación y desarrollo del proyecto mucho más ágil, permitiendo la detección y corrección precoz a los problemas que iban surgiendo sin que ello implicara un desajuste en la programación que conllevara la aplicación de las contingencias correspondientes.

El talón de Aquiles del proyecto ha estado en el desarrollo de los correspondientes procedimientos almacenados, que ha requerido de un sobreesfuerzo no contemplado y que ya requerido del aumento de la dedicación a estas tareas debido a la falta de experiencia en el desarrollo de estos.

A pesar de los contratiempos surgidos, se ha podido conseguir cumplir con todos los objetivos marcados en el enunciado del proyecto y dar respuesta a los requerimientos planteados, por ello me siento totalmente satisfecho del trabajo desarrollado y que me ha permitido plasmar los conocimientos adquiridos durante el desarrollo de mi formación académica.

3.3 Acciones futuras

A pesar de que se ha conseguido dar respuesta a los requerimientos indicados en el enunciado del proyecto, hay una serie de puntos que requieren de una mejor definición de concepción y desarrollo.

Por un lado tenemos la gestión de los centros de votación, los cuales solo se genera uno por cada población cuando en la realizada existen varios centros en cada uno de los municipios donde se celebra una convocatoria.

En el desarrollo de los procedimientos almacenados se utiliza repetitivamente en varios de ellos una serie de validaciones de datos, lo cual sería más óptimo desarrollarlo como funciones independientes a las que llamar para realizar dichas validaciones.

4. Glosario

Cardinalidad: Número de entidades con la cual otra entidad se puede asociar mediante una relación.

Clave foránea: Limitación referencial entre dos tablas que identifica una columna o grupo de columnas en una tabla con una columna o grupo de columnas en otra tabla.

Clave primaria: Campo o a una combinación de campos que identifica de forma única a cada fila de una tabla.

Data Warehouse: Colección de datos orientada a un determinado ámbito, integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones en la entidad en la que se utiliza.

Diagrama de Gantt: Herramienta gráfica cuyo objetivo es exponer el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado

Diseño conceptual: Representación de un sistema, hecho de la composición de conceptos que se utilizan para ayudar a las personas a conocer, comprender o simular un tema que representa el modelo, incluye las entidades importantes y las relaciones entre ellos.

Diseño lógico: Etapa de diseño que desarrolla un esquema lógico para una base de datos relacional a partir de un esquema conceptual.

Disparador: Objeto que se asocia con tablas y que se ejecuta cuando sucede algún evento sobre las tablas a las que se encuentra asociado.

Entidad: Representación de un concepto en el mundo real.

Modelo entidad-relación: Representación de las entidades de un sistema de información con sus correspondientes interrelaciones y propiedades.

Normalización: Proceso que consiste en designar y aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional.

Procedimiento almacenado: Programa almacenado en base de datos que se ejecuta directamente en el motor de bases de datos.

SQL: Lenguaje de consulta de base de datos.

UML: Lenguaje unificado de modelado.

5. Bibliografía

Casas Roma, J., & Conesa i Caralt, J. (2014). *Diseño conceptual de bases de datos en UML*. 156.

Illa, X. B., & Illa, X. B. (s. f.). *Diseño lógico de bases de datos*.

Roma, J. C., Costa, D. C., & Juanola, P. J. (s. f.). *Diseño físico de bases de datos*.

C.J.Date, & C.J.Date. (1994). «*An Introduction to Database Systems*». Addison-Wesley.

E.F.Codd, & E.F.Codd. (1970). «*A Relational Model of Data for Large Shared Databanks*». Communications of the ACM.

BOE.es - BOE-A-1985-11672 Ley Orgánica 5/1985, de 19 de junio, del Régimen Electoral General. (s. f.). Recuperado 10 de marzo de 2021, de <https://www.boe.es/buscar/act.php?id=BOE-A-1985-11672>

Censo electoral - Wikipedia, la enciclopedia libre. (s. f.). Recuperado 11 de marzo de 2021, de https://es.wikipedia.org/wiki/Censo_electoral

Normalización de bases de datos - Wikipedia, la enciclopedia libre. (s. f.). Recuperado 27 de marzo de 2021, de https://es.wikipedia.org/wiki/Normalización_de_bases_de_datos

INEbase/ Demografía y población / Padrón. Población por municipios. (s. f.). Recuperado 20 de abril de 2021, de https://www.ine.es/dyngs/INEbase/es/categoria.htm?c=Estadistica_P&cid=1254734710990

INE - CensoElectoral - Procesos Electorales. (s. f.). Recuperado 20 de abril de 2021, de https://www.ine.es/ss/Satellite?L=es_ES&c=Page&cid=1254735832447&p=1254735832447&pagename=CensoElectoral%2FINELayout

*SQL*Plus® User's Guide and Reference - Contents*. (s. f.). Recuperado 26 de abril de 2021, de https://docs.oracle.com/cd/E11882_01/server.112/e16604/toc.htm

Formula - Sistema D'Hondt - Justicia Electoral · República del Paraguay. (s. f.). Recuperado 30 de abril de 2021, de <https://www.tsje.gov.py/sistema-d'hondt.html>

Cómo funciona la fórmula d'Hondt. (s. f.). Recuperado 02 de mayo de 2021, de <https://hipertextual.com/2015/05/formula-dhondt-como-funciona>

PL/SQL. Procedimientos y Funciones. (s. f.). Recuperado 03 de mayo de 2021, de <https://elbauldeldelprogramador.com/plsql-procedimientos-y-funciones/>

ORACLE-BASE - Global Temporary Tables. (s. f.). Recuperado 15 de mayo de 2021, de <https://oracle-base.com/articles/misc/temporary-tables>