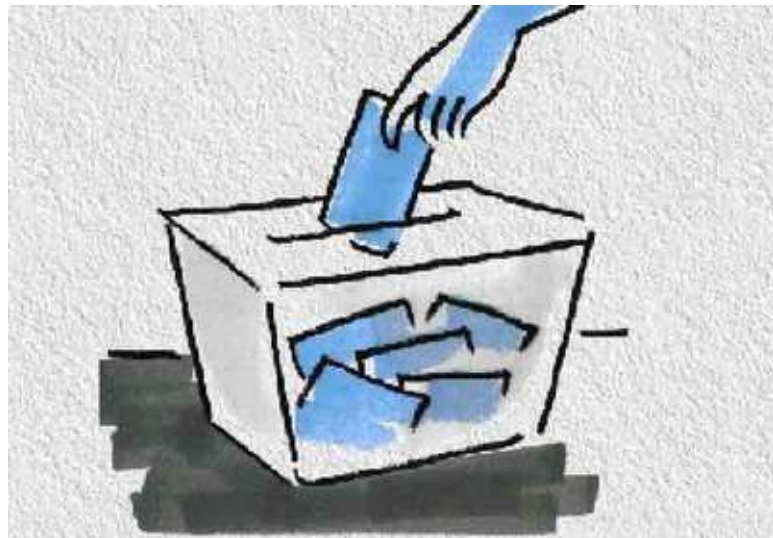


# DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE VOTACIÓN CIUDADANA A NIVEL EUROPEO



## MEMORIA del TFC

**Laura Sánchez Milián**  
ETIG

**Ismael Perez Laguna**

15/01/2011

<b>MEMORIA DEL TRABAJO DEL TFC .....</b>	<b>4</b>
<b>1. Introducción .....</b>	<b>4</b>
1.1. Resumen del TFC .....	4
1.2. Objetivos Generales y específico.....	4
1.3. Enfoque y método seguido .....	5
1.4. Planificación del Proyecto .....	6
1.4.1. Planificación PACs .....	9
1.5. Recursos Empleados .....	10
1.6. Análisis y control de riesgos .....	10
1.7. Valoración y Costos del Proyecto.....	11
1.7.1. Valoración Económica Optimista .....	13
1.7.2. Valoración Económica Pesimista .....	13
<b>2. Análisis y Diseño .....</b>	<b>14</b>
2.1. Análisis de requisitos.....	14
2.2. Diseño del Modelo E/R .....	17
2.2.1. Comentarios sobre el Modelo E/R realizado: .....	17
2.2.2. Supuestos Semánticos no reflejados en el Diagrama .....	19
2.2.3. Atributos de las Entidades .....	20
2.3. Diseño del Modelo Lógico .....	22
2.3.1. Modelo Relacional .....	22
2.3.2. Restricciones de Integridad Referencial o clave ajena.....	24
2.3.3. Diagrama del Modelo Relacional .....	25
<b>3. Implementación de la Base de Datos .....</b>	<b>27</b>
3.1. Tablas Generales .....	27
3.2. Tablas del Modulo Estadístico.....	28
3.3. Otras Tablas.....	29
3.4. Secuencias .....	29
3.5. Triggers .....	29
3.6. Vistas .....	31
3.7. Procedimientos Almacenados.....	31
3.7.1. INSERTAR LOG.....	32
3.7.2. ALTA ESTADO .....	32
3.7.3. ALTA AMBITO .....	33
3.7.4. ALTA CENSO GLOBAL .....	33
3.7.5. ALTA CENSO NACIONAL.....	34
3.7.6. ALTA CENSO REGIONAL .....	34
3.7.7. MODIFICACIÓN CENSO REGIONAL.....	35
3.7.8. BAJA CENSO REGIONAL.....	36
3.7.9. ALTA CENSO LOCAL .....	37
3.7.10. MODIFICACIÓN CENSO LOCAL.....	37
3.7.11. BAJA CENSO LOCAL .....	38
3.7.12. ALTA CENTRO ELECTORAL .....	39
3.7.13. MODIFICACIÓN CENTRO ELECTORAL .....	40
3.7.14. BAJA CENTRO ELECTORAL .....	40
3.7.15. ALTA MESA ELECTORAL.....	41
3.7.16. BAJA MESA ELECTORAL .....	41
3.7.17. ALTA VOTO NULO.....	42
3.7.18. ALTA VOTO EN BLANCO .....	43
3.7.19. ALTA PARTIDO POLITICO.....	43
3.7.20. MODIFICACIÓN PARTIDO POLITICO.....	44
3.7.21. BAJA PARTIDO POLÍTICO .....	45
3.7.22. ALTA ELECTOR .....	45
3.7.23. MODIFICACIÓN ELECTOR .....	46
3.7.24. BAJA ELECTOR.....	48
3.7.25. ALTA CODIGO ALEATORIO ELECTOR.....	48
3.7.26. ALTA CANDIDATO .....	49

3.7.27.	MODIFICACIÓN CANDIDATO .....	50
3.7.28.	BAJA CANDIDATO .....	50
3.7.29.	ALTA VOTACION .....	51
3.7.30.	MODIFICACIÓN VOTACIÓN .....	52
3.7.31.	BAJA VOTACIÓN.....	53
3.7.32.	ALTA DE UNA OPCION DE VOTACION .....	53
3.7.33.	BAJA DE UNA OPCIÓN DE VOTACIÓN.....	54
3.7.34.	ALTA DE UN CENSO EN UNA VOTACIÓN .....	55
3.7.35.	BAJA DE UN CENSO EN UNA VOTACIÓN .....	56
3.7.36.	ALTA DE UN CANDIDATO A UN PARTIDO POLITICO EN UNA VOTACIÓN.....	56
3.7.37.	MODIFICACIÓN DE UN CANDIDATO A UN PARTIDO POLITICO EN UNA VOTACIÓN ...	57
3.7.38.	BAJA DE UN CANDIDATO A UN PARTIDO POLITICO EN UNA VOTACIÓN .....	58
3.7.39.	ALTA VOTAR .....	59
3.7.40.	RELLENAR ESTADISTICAS.....	60
3.7.41.	INSERTAR ELECTORES EN UNA VOTACIÓN .....	61
3.7.42.	INSERTAR UN ELECTOR EN UN CENSO .....	62
3.7.43.	ACTUALIZAR ESTADO .....	63
3.7.44.	EDITAR ESTADO DE UNA VOTACIÓN.....	64
3.8.	LISTADOS.....	65
3.8.1.	LISTADO DE VOTACIONES DE UN PAÍS .....	65
3.8.2.	LISTADO DE LAS 10 VOTACIONES CON MAYOR DIFERENCIA PORCENTUAL .....	66
3.8.3.	LISTADO DE LAS 10 VOTACIONES CON MENOR DIFERENCIA PORCENTUAL.....	67
3.8.4.	LISTADO DE LOS CENSOS EN LOS QUE ESTÁ INSCRITO UN CIUDADANO. ....	68
3.8.5.	LISTADO DE LAS VOTACIONES EN LAS QUE HA PARTICIPADO UN CIUDADANO. ....	69
<b>4.</b>	<b>Bibliografía .....</b>	<b>71</b>

# MEMORIA DEL TRABAJO DEL TFC

## 1. Introducción

### 1.1. Resumen del TFC

La Comunidad Europea para fomentar la participación ciudadana dentro del ámbito político Europeo, quiere implementar una aplicación de votaciones ciudadanas a través de Internet.

El proyecto consiste en desarrollar la primera fase de esta aplicación, para la cual la Comunidad Europea ha abierto un concurso público, diseñando una Base de datos que permita almacenar toda la información necesaria para realizar las votaciones y guardar un histórico de las mismas, dejando el desarrollo de la interfaz gráfica y la gestión a través de Internet para una segunda fase.

A nivel general la Base de datos deberá guardar la información de las distintas votaciones de los ciudadanos manteniendo en todo momento el secreto de las mismas, así como permitir gestionar los distintos censos electorales asignados a los ciudadanos y votaciones. Con esta información el sistema deberá permitir introducir los votos de los ciudadanos, guardando información relativa a los mismos tales como votos emitidos, personas con capacidad de voto, porcentaje de participación, etc... permitiendo realizar con dicha información distintas consultas habituales como votaciones emitidas por un país, los censos electorales a los que pertenece un ciudadano, etc.

Además la Base de Datos deberá permitir precalcular y almacenar diversa información estadística a partir de procedimientos ya implementados en el proyecto que aumenten la eficacia y rapidez del sistema.

### 1.2. Objetivos Generales y específico

El principal objetivo del TFC consiste en desarrollar un proyecto que permita poner en práctica y ampliar los conocimientos adquiridos durante el estudio de la carrera de Ingeniería Técnica de Gestión, principalmente en el área de las asignaturas de Bases de Datos I, Bases de Datos II y Estructura de la Información.

Otro objetivo general a tener en cuenta, que va ligado intrínsecamente al objetivo principal, es la capacidad de realizar un proyecto en su totalidad pasando por las distintas etapas que lo conforman desde el análisis previo de los objetivos hasta la entrega de un producto final.

Los objetivos específicos son los siguientes:

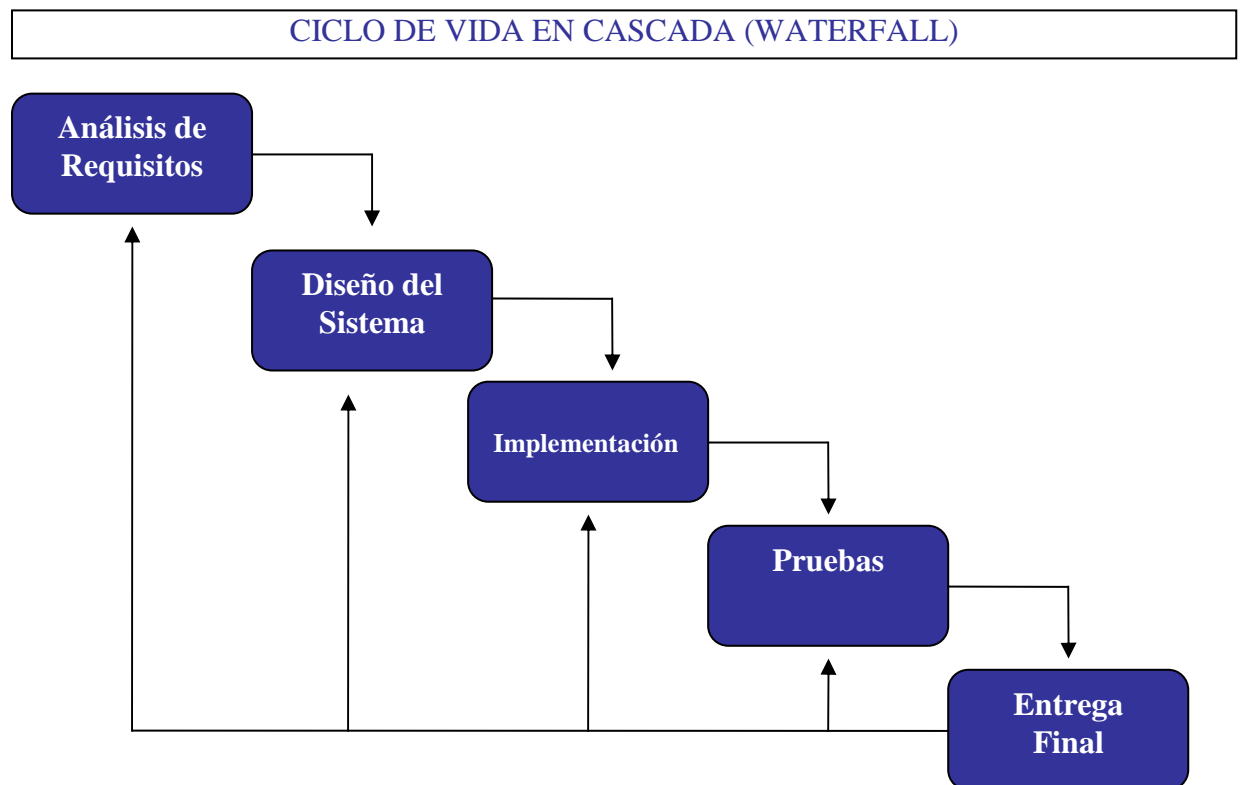
- Adquirir conocimientos y soltura con el sistema de gestión de Bases de Datos Objeto-Relacional denominado Oracle.
- Aprender a desenvolverse con el lenguaje PL/SQL, extensión del lenguaje SQL con características de los lenguajes de programación.
- Profundizar en el conocimiento del lenguaje SQL
- Entrega de tres pruebas de evaluación continuada PAC1, PAC2, PAC3.

- Entrega final del TFC formado por Memoria, Presentación y Producto.

### 1.3. Enfoque y método seguido

Para realizar la distribución de las tareas a largo de la duración del proyecto y poder garantizar la obtención de un producto finalizado y con garantías en una fecha clave 15/01/2012, se ha optado por un modelo de ciclo de vida del software en Cascada, dada las características del proyecto de mediana complejidad y de los medios humanos y técnicos disponibles.

Hay que considerar la característica peculiar del proyecto, en el que una sola persona tiene que adoptar los distintos roles, desde jefe de Proyecto, Analista, Programador hasta Testeador.



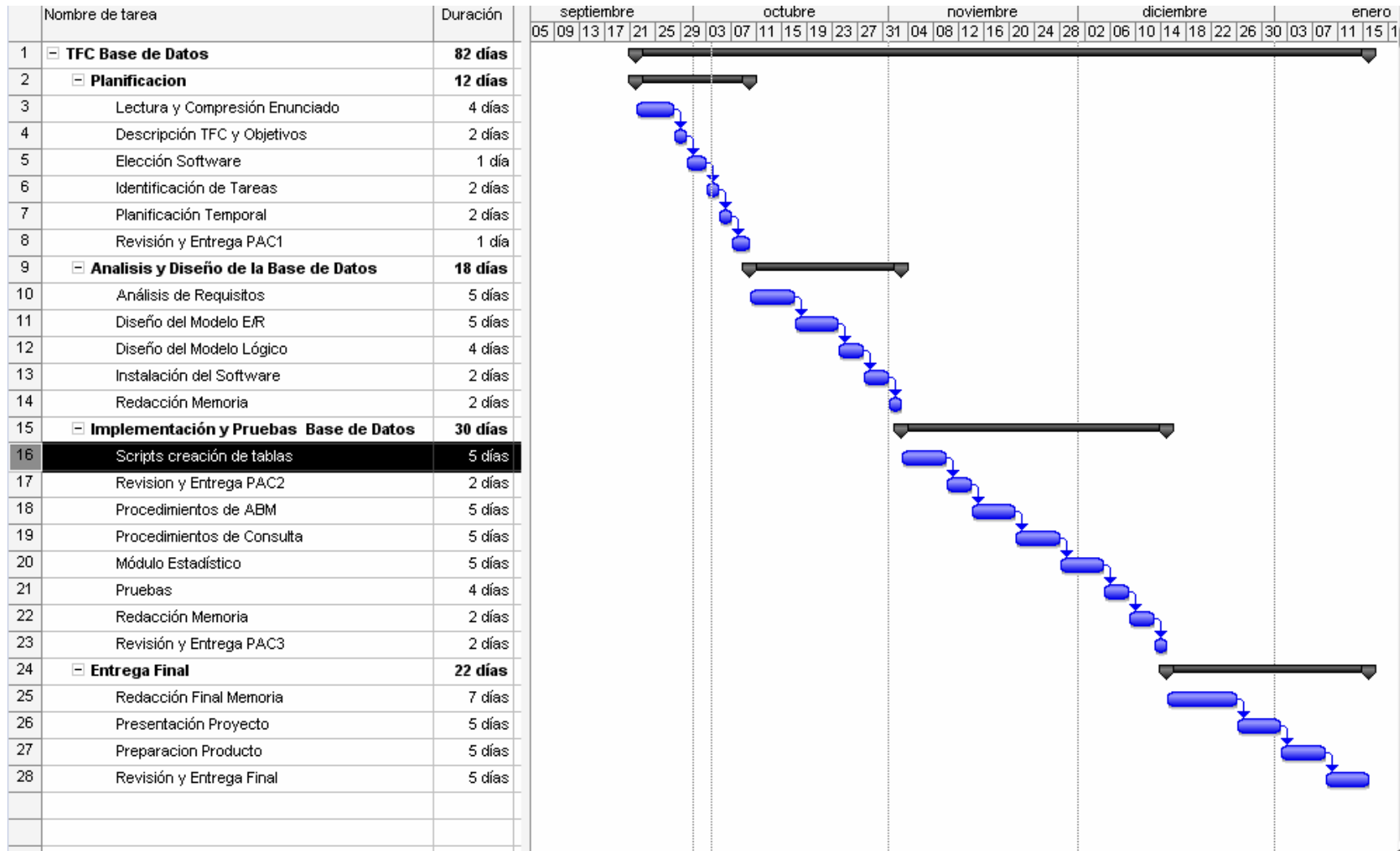
## 1.4. Planificación del Proyecto

Las Tareas principales en las que se descompone el proyecto son las siguientes:

- Elaboración del Plan de Trabajo
  - Lectura y comprensión del enunciado del TFC
  - Redacción de los puntos Descripción del TFC y Objetivos generales y específicos, que servirán posteriormente para la Memoria.
  - Elección del Software para el proyecto.
  - Definir las tareas en la que se puede descomponer el proyecto.
  - Planificar la distribución de las tareas a lo largo del Semestre.
  - Elaboración del Plan de Trabajo.
- Análisis y Diseño de la Base de Datos
  - Análisis de Requisitos que solicita el cliente
  - Instalación del Software de desarrollo necesario.
  - Diseño del Modelo E/R
  - Diseño del Modelo lógico
- Implementación de la Base de Datos
  - Implementar las tablas necesarias sobre votaciones , censos electorales y ciudadanos para crear la Base de Datos
  - Implementar los procedimientos almacenados para poder dar de Alta, Baja y Modificación los elementos anteriores.
  - Implementar los procedimientos de consulta sobre votaciones.
  - Implementación del módulo Estadístico
- Pruebas
  - Elaboración de un juego de pruebas que permita comprobar el buen funcionamiento de la Base de Datos.
- Entrega Final
  - Redacción final de la Memoria: Durante el semestre se habrán elaborado aquellas partes de la Memoria relacionadas con las entregas de las respectivas PAC.
  - Elaboración de la Presentación Virtual del Proyecto.
  - Revisión y Entrega final del Proyecto: Memoria, Presentación y Base de Datos.

	Nombre de tarea	Duración	Comienzo	Fin
1	<b>TFC Base de Datos</b>	<b>82 días</b>	<b>jue 22/09/11</b>	<b>dom 15/01/12</b>
2	<b>Planificación</b>	<b>12 días</b>	<b>jue 22/09/11</b>	<b>dom 09/10/11</b>
3	Lectura y Compresión Enunciado	4 días	jue 22/09/11	mar 27/09/11
4	Descripción TFC y Objetivos	2 días	mié 28/09/11	jue 29/09/11
5	Elección Software	1 día	vie 30/09/11	dom 02/10/11
6	Identificación de Tareas	2 días	lun 03/10/11	mar 04/10/11
7	Planificación Temporal	2 días	mié 05/10/11	jue 06/10/11
8	Revisión y Entrega PAC1	1 día	vie 07/10/11	dom 09/10/11
9	<b>Análisis y Diseño de la Base de Datos</b>	<b>18 días</b>	<b>lun 10/10/11</b>	<b>mié 02/11/11</b>
10	Análisis de Requisitos	5 días	lun 10/10/11	dom 16/10/11
11	Diseño del Modelo E/R	5 días	lun 17/10/11	dom 23/10/11
12	Diseño del Modelo Lógico	4 días	lun 24/10/11	jue 27/10/11
13	Instalación del Software	2 días	vie 28/10/11	lun 31/10/11
14	Redacción Memoria	2 días	mar 01/11/11	mié 02/11/11
15	<b>Implementación y Pruebas Base de Datos</b>	<b>30 días</b>	<b>jue 03/11/11</b>	<b>mié 14/12/11</b>
16	Scripts creación de tablas	5 días	jue 03/11/11	mié 09/11/11
17	Revisión y Entrega PAC2	2 días	jue 10/11/11	dom 13/11/11
18	Procedimientos de ABM	5 días	lun 14/11/11	dom 20/11/11
19	Procedimientos de Consulta	5 días	lun 21/11/11	dom 27/11/11
20	Módulo Estadístico	5 días	lun 28/11/11	dom 04/12/11
21	Pruebas	4 días	lun 05/12/11	jue 08/12/11
22	Redacción Memoria	2 días	vie 09/12/11	lun 12/12/11
23	Revisión y Entrega PAC3	2 días	mar 13/12/11	mié 14/12/11
24	<b>Entrega Final</b>	<b>22 días</b>	<b>jue 15/12/11</b>	<b>dom 15/01/12</b>
25	Redacción Final Memoria	7 días	jue 15/12/11	dom 25/12/11
26	Presentación Proyecto	5 días	lun 26/12/11	dom 01/01/12
27	Preparación Producto	5 días	lun 02/01/12	dom 08/01/12
28	Revisión y Entrega Final	5 días	lun 09/01/12	dom 15/01/12

Diagrama de Gantt con planificación temporal del proyecto





### 1.4.1. Planificación PACs

De acuerdo con la planificación realizada en el Diagrama de Gantt, se tiene previsto entregar las siguientes tareas:

Fecha	Evento	Entrega
09/10/2011	PAC1	Plan de Trabajo
13/11/2011	PAC2	Análisis y Diseño BD Scripts de Creación BD Memoria
14/12/2011	PAC3	Procedimientos Almacenados Disparadores Pruebas Memoria
15/01/2011	Entrega Final	Memoria Completa Presentación Virtual Producto

**Plan de Trabajo:** Elaborar la plantilla o documento que nos sirva de directriz para llegar a cabo la inicialización y seguimiento del proyecto, mediante la identificación de los objetivos y tareas principales así como su ejecución en el tiempo.

**Análisis y Diseño de la Base de Datos:** Identificar las necesidades y requerimientos del cliente mediante la lectura detallada del enunciado y comunicación fluida con el cliente, en este caso consultor del Proyecto, para resolver cualquier duda que pudiera surgir al respecto. A partir de esta Información se diseñará el Modelo E/R y se procederá a su correspondiente Transformación al Modelo Lógico

**Scripts de Creación de BD:** Scripts del lenguaje PL/SQL necesarios para crear la Base de datos, sus entidades y relaciones de acuerdo con el diseño anterior.

**Procedimientos Almacenados y Disparadores:** Scripts de código PL/SQL necesarios para crear los procedimientos almacenados de consulta y estadísticos que requiere el enunciado.

**Pruebas:** Conjunto de pruebas que confirmen el correcto funcionamiento de la Base de Datos.

**Memoria:** Documento explicativo del Proyecto.

## 1.5. Recursos Empleados

Para el desarrollo del siguiente proyecto se ha estimado emplear los siguientes medios humanos, técnicos y materiales:

- **Hardware:** Ordenador de sobremesa HP Pavilion con Procesador AMD Sempron 140 2.71 GHz, 1 GB de Memoria RAM, 200GB de Disco Duro y sistema operativo Microsoft Windows XP Profesional 2002 Service Pack 2. Este ordenador desempeñará los roles tanto de servidor como de cliente de la Base de Datos.
- **Software:** Oracle Express v10.2.0.1 como SGBD, SQL Developer como entorno de desarrollo, prueba, depuración de errores y optimización de PL/SQL de Oracle. Microsoft Word 2003 como procesador de textos. PDF Creator como conversor de documentos a formato pdf. Microsoft Project 2007 para la elaboración del Plan de Trabajo. Microsoft PowerPoint para la elaboración de la presentación Virtual.
- **Humanos:** La elaboración del proyecto será realizada en la totalidad por una misma persona que dependiendo de la tarea del proyecto desempeñará las funciones de Jefe de Proyecto, Analista, Programador y Testeador.

## 1.6. Análisis y control de riesgos

Los distintos riesgos a enfrentar a la hora de desarrollar el proyecto pueden surgir de la falta de conocimiento de los programas a utilizar, principalmente el SGBD Oracle así como de imprevistos tanto humanos (enfermedades, trabajo, accidentes,...) como técnicos (fallo del disco duro, pérdida de datos ocasionados por un virus,...).

Para solucionar los retrasos ocasionados por los contratiempos anteriores en el correcto desarrollo del proyecto se tomarán las siguientes medidas preventivas:

- La jornada de trabajo se entenderá de Lunes a Viernes dejando el fin de semana como medida de seguridad en el caso de tener que recuperar tiempo, debido a una mala planificación del plan de trabajo o a un retraso por los riesgos descritos anteriormente.
- Para prevenir la pérdida de datos debido a un problema técnico o simplemente a un descuido humano, se realizará una copia de seguridad diaria una vez finalizado el trabajo.

## 1.7. Valoración y Costos del Proyecto

La valoración del Costo del proyecto se ha realizado dedicando aproximadamente una jornada de 2 horas diarias de Lunes a Viernes.

Según el estudio realizado sobre los salarios en el campo de las tecnologías, por Euribor Empleo: <http://www.euribor.com.es/empleo/salarios-2011/>

Podemos establecer como referencia los siguientes salarios:

### RESPONSABLE DE INFORMÁTICA

#### REMUNERACIÓN EN EUROS

	+35.000

### ANALISTA PROGRAMADOR

#### REMUNERACIÓN EN EUROS

Experiencia previa		2-3 años	+ 3 años
FPI / FPII	Mín.	33.000	NS
	Máx.	36.000	
ING TCO / SUP	Mín.	33.000	NS
	Máx.	36.000	

\* NS = No significativo

### PROGRAMADOR

#### REMUNERACIÓN EN EUROS

Experiencia previa		0-1 años	2-3 años	3-5 años
FPI / FPII	Mín.	18.000	21.000	25.000
	Máx.	21.000	25.000	30.000
DIP / LIC	Mín.	18.000	21.000	27.000
	Máx.	21.000	27.000	35.000

## TÉCNICO DE CALIDAD DE SOFTWARE (TESTER)

### REMUNERACIÓN EN EUROS

Experiencia previa	0-1 años	2-4 años	+4 años
Min.	22.000	26.000	NS
Máx.	26.000	33.000	NS

\*NS = No significativo

Para realizar los cálculos aproximados seleccionaremos el sueldo de un profesional de experiencia media entre 2 – 3 años y estableceremos el salario mínimo y máximo como una valoración optimista y pesimista del coste.

Teniendo en cuenta que la jornada anual es aproximadamente de 1800 horas, calculamos el precio por hora de cada trabajador.

Profesión	Salario mín	Salario máx	Precio/hora mín	Precio/ hora máx
Jefe Proyecto	35.000	38.000	19.44	21.11
Analista Programador	33.000	36.000	18.33	20
Programador	21.000	25.000	11.66	13.88
Testeador	26.000	33.000	14.44	18.33

Tomando de referencia nuestra planificación y estimando que podremos dedicar al proyecto unas 2 h/diarias de Lunes a Viernes, que podrían aumentar en el caso de imprevistos, vamos a realizar una estimación del número de horas necesarias para cada profesional.

La Planificación correspondería al Jefe del Proyecto, dedicando a esta fase 12 días y en cada día se emplean 2 horas, tendríamos en total 12días x 2h/día = 24 horas.

Así mismo el Jefe del Proyecto se encargará de revisar el Proyecto durante todo su ciclo de vida así como de redactar la memoria del mismo. Por lo que a estas horas le sumaremos las dedicadas a estas tareas que son 8 (Revisión y Memoria) + 22 (Entrega final) días = 30 días x 2 = 60 horas.

Horas Jefe del Proyecto 24 h + 60 h = 84 horas

Si tuviéramos que emplear el fin de semana las horas se incrementarían en 6días x 2h/día = 12 h.

Nº horas del Jefe de Proyecto = Entre 84 y 96 horas.

Procederíamos a realizar los mismos cálculos para los restantes profesionales teniendo en cuenta que el Analista-Programador se encargará del Análisis y Diseño de la Base de Datos, el Programador de la Implementación y el Testeador de las Pruebas.

El resultado de todos los cálculos se puede resumir en la siguiente Tabla:

<b>Profesión</b>	<b>Días Trabajo</b>	<b>Horas al día</b>	<b>Total horas</b>
<b>Jefe de Proyecto</b>	42-48	2	84 -96
<b>Analista Programador</b>	16-22	2	32-44
<b>Programador</b>	20-32	2	40-62
<b>Testeador</b>	4-6	2	8-12

Una vez que tenemos calculado el precio por hora y el número de horas que tiene que dedicar cada profesional podemos hacer una estimación del coste total.

### 1.7.1. Valoración Económica Optimista

<b>Profesión</b>	<b>Horas</b>	<b>Coste</b>	<b>Total Euros</b>
<b>Jefe de Proyecto</b>	84	19,44	1.633
<b>Analista Programador</b>	32	18,33	587
<b>Programador</b>	40	11,66	466
<b>Testeador</b>	8	14,44	116
			<b>2.802</b>

### 1.7.2. Valoración Económica Pesimista

<b>Profesión</b>	<b>Horas</b>	<b>Coste</b>	<b>Total Euros</b>
<b>Jefe de Proyecto</b>	96	21,11	2.027
<b>Analista Programador</b>	44	20	880
<b>Programador</b>	62	13,88	861
<b>Testeador</b>	12	18,33	220
			<b>3.988</b>

En cuanto al software entendemos que el cliente tiene la licencia de SGBD Oracle por lo que no lo incluimos en la valoración, en caso contrario se añadiría a la valoración económica total.

## 2. Análisis y Diseño

Para recoger la información necesaria para el análisis de los requisitos del usuario se ha tomado como referencia tanto el enunciado del trabajo final de Carrera como la información relativa a elecciones electorales obtenidas por Internet. Se ha estimado mantener la estructura de colegios electorales y mesas con el fin de mantener la igualdad de oportunidades de los ciudadanos a la hora de acceder a los medios de votación, ya que se estima que hoy en día no todos los ciudadanos disponen de un ordenador conectado a Internet, para poder gestionar las votaciones únicamente por este medio. Además este sistema nos permitirá aumentar la funcionalidad futura de analizar cuantos votantes han realizado las votaciones desde cabinas telemáticas preparadas en los centros electorales o desde su propia casa. Pudiendo realizar un estudio estadístico de la implantación y uso de las nuevas tecnologías en los hogares de la población Europea.

### 2.1. Análisis de requisitos

Los Requisitos funcionales que vienen recogidos en el enunciado del proyecto y aquellos otros que se pueden deducir, son los siguientes:

- R1: Se ha de poder almacenar todos los datos concernientes a una votación: Título, Descripción, opciones a votar, fecha inicio y fecha fin de la votación, fecha de publicación, censos habilitados, nombre del presidente y dirección Web.
- R2: La votación tiene un ámbito definido. con el cual se habilitan los censos a los que tiene acceso: GLOBAL, ESTATAL, REGIONAL o LOCAL. Resumiendo cada votación tiene habilitados unos censos dependiendo del ámbito de la votación.
- R3: Cada votación tiene varias opciones a votar, como mínimo dos, pudiendo ser los distintos partidos políticos, o la opción de voto blanco o voto nulo.
- R4: Se almacenará la información relativa a los distintos partidos políticos que se pueden presentar en cada votación.
- R5: Se almacenará el número de personas con capacidad de voto en dichas elecciones al cierre de las mismas, los votos totales emitidos y el porcentaje de participación de cada votación También permitirá almacenar el número de votos obtenidos en cada opción y su porcentaje con respecto al total.
- R6: Se permitirá la gestión de los distintos censos electorales: GLOBAL, ESTATAL, REGIONAL Y LOCAL.
- R7: Solamente se podrán inscribir en los censos electorales los ciudadanos de la Comunidad Europea que sean mayores de 18 años.
- R8:
  - Solamente se podrá crear una instancia del Censo GLOBAL. Estará formado por todos los ciudadanos de los países de la Unión Europea.

- En el Censo Estatal se guardará el Código del País como un metadata según la codificación ISO 3166-1 alfa 2. Estará formado por todos los ciudadanos de un país de la Comunidad Europea.
- En el Censo Regional se almacenará el país y la región a la que pertenece. Estará formado por todos los ciudadanos de una región concreta de Europa.
- En el Censo Local se almacenará el código del país, de la región y el metadata del nombre del pueblo. Estará formado por todos los ciudadanos de un pueblo o ciudad de Europa.
- R9: Un ciudadano puede pertenecer a más de un Censo.
- R10: Un ciudadano sólo puede emitir un voto en cada votación y una vez emitido éste no se podrá anular ni rectificar.
- R11: En los ciudadanos pertenecientes a los censos electorales únicamente se podrá almacenar un código aleatorio, la fecha nacimiento, el código del país, región y localidad.
- R12: Si un ciudadano no es Europeo no puede votar, ni puede inscribirse en ninguno de los censos.
- R13: Todo ciudadano Europeo puede participar en las elecciones a nivel Europeo.
- R14: Todo ciudadano Europeo puede votar en las elecciones municipales donde resida si está inscrito en el Padrón electoral Municipal y por lo tanto en el censo municipal, pero sólo podrá votar a las elecciones Nacionales y Regionales pertenecientes a su país y por tanto de la región donde esté inscrito en su país. De acuerdo con ello puede estar inscrito en un censo Municipal y en un censo Nacional y Regional que no corresponde con el país y región a la que pertenece el municipio donde reside.
- R15: La Aplicación debe tener como mínimo las siguientes funcionalidades:
  - Alta, Baja y Modificación de Votaciones.
  - Alta, Baja y Modificación de Partidos Políticos.
  - Alta, Baja y Modificación de los distintos Censos.
  - Alta, Baja y Modificación de Ciudadanos.
  - Alta, Baja y Modificación de Centros Electorales.
  - Alta, Baja y Modificación de Mesas Electorales.
- R16: Cada vez que se ejecute un procedimiento almacenado, se almacenará en una tabla denominada log: El nombre del procedimiento, fecha, parámetros de entrada y parámetros de salida.
- R17: Se han de permitir los siguientes procedimientos de consulta:

- Todas las votaciones que han sido celebradas en un país en concreto ordenadas cronológicamente de forma ascendente por la fecha de inicio de la votación.

Deberá mostrar los siguientes campos: Título, fecha inicio y fin de la votación, número de ciudadanos de los censos habilitados en la votación, estado de la votación y porcentaje de participación total, opción ganadora y votos de ésta. En caso de que no este finalizada la votación poner en los campos pertinentes “-“.

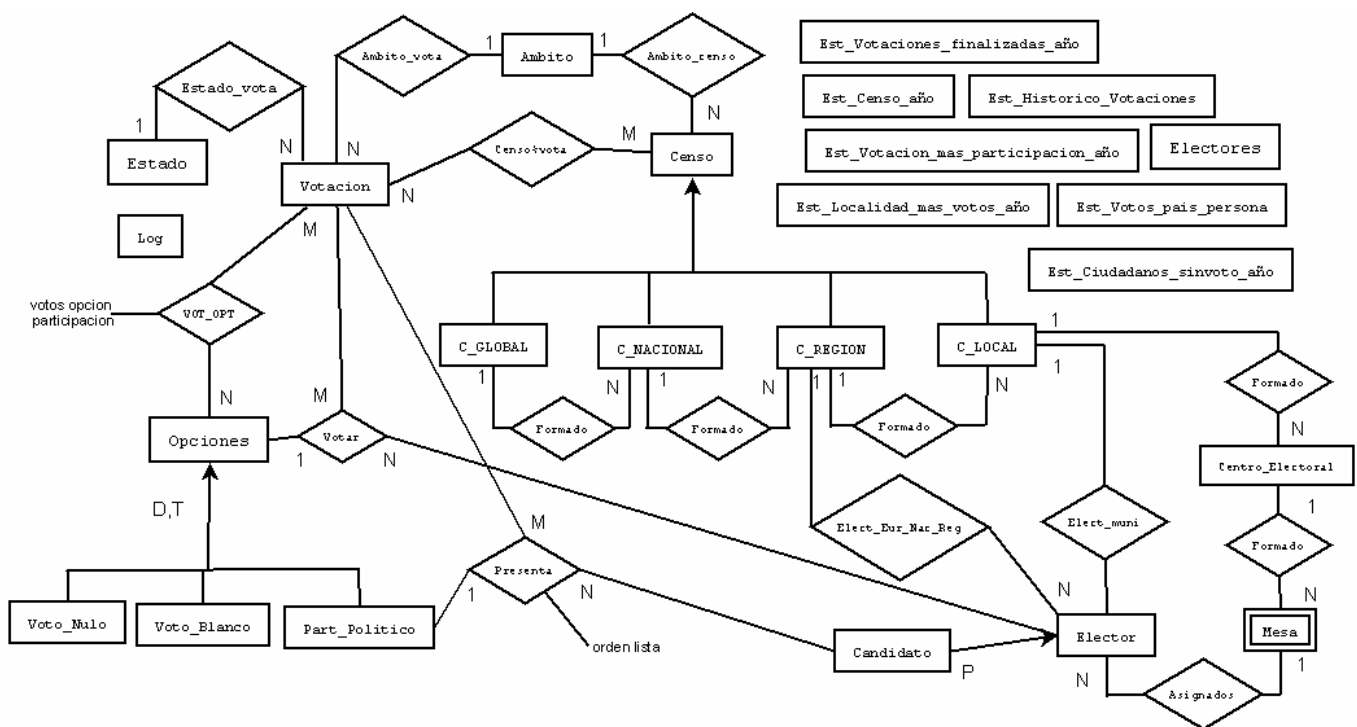
- Las 10 votaciones con más diferencia porcentual de votos, entre la opción más votada y la menos votada.
  - Las 10 votaciones con menos diferencia porcentual de votos, entre la opción más votada y la menos votada.
  - Ambos listados deberán mostrar: Título, fecha inicio y fin de la votación, opción más votada, menos votada y sus porcentajes respecto al total.
  - Todos los censos a los que pertenece un ciudadano dado su código aleatorio.
  - Todas las votaciones en las que ha participado y en las que ha dejado de participar un ciudadano dado su código aleatorio.
- R18: Se debe crear un módulo estadístico que permita la consulta de una serie de informes en tiempo constante 1. Las consultas necesarias son las siguientes:
- Numero de votaciones finalizadas de un país en un año concreto.
  - El valor medio del porcentaje de participación de las votaciones celebradas durante un año concreto en un censo determinado.
  - El número de votaciones asociadas a un censo en un año concreto.
  - La votación que más participación ha tenido en un año en concreto.
  - La localidad Europea en la que más votos han emitido sus ciudadanos en un año en concreto.
  - Porcentaje de ciudadanos que nunca han votado en un año en concreto.
  - Votación que históricamente ha tenido más participación.
  - Votación que históricamente ha tenido más diferencia porcentual entre la opción más votada y la menos.
  - Votación que históricamente ha tenido menos diferencia porcentual entre la opción más votada y la menos.
  - Número máximo de votaciones en las que ha participado una misma persona y el país de esta persona, teniendo en cuenta toda la historia de votaciones de Europa.



## 2.2. Diseño del Modelo E/R

En esta fase del proyecto se define la estructura de la información relativa a la Base de Datos de las elecciones, independiente de la tecnología que se emplee. Para expresar el resultado se utiliza un modelo de datos de alto nivel “El modelo Entidad Relación”.

Una vez analizado el universo del proyecto de las elecciones a partir tanto del enunciado del problema como del intercambio de información con el cliente se obtiene el siguiente diagrama E/R:



### 2.2.1. Comentarios sobre el Modelo E/R realizado:

Las votaciones tienen un ámbito definido, que puede ser de tipo Global (Europa), Nacional, Regional o Local al igual que los Censos. Se interpreta que una votación sólo puede tener un ámbito definido. Lo mismo ocurre con los Censos, un censo pertenece a un solo ámbito, pero una votación dentro de un ámbito establecido puede tener varios censos de ese ámbito habilitado.

En este supuesto puede parecer que la relación Censo\_votacion se puede eliminar al producirse un ciclo entre esta y las relaciones ambito\_voto y ambito\_censo, ya que sabiendo el ámbito de una votación sabemos los censos de ese ámbito. Pero esto no es correcto ya que una votación no tiene porque tener habilitados todos los censos de un ámbito, por ejemplo una votación de ámbito nacional puede tener sólo habilitados los censos de España y Alemania y al eliminar la relación censos habilitados, podríamos acceder a aquellos censos que realmente puede habilitar la votación pero no los que se han habilitado para esa votación en concreto. Luego debemos mantener el ciclo.

Como las votaciones una veces se van a relacionar con un tipo de censo y otras veces con otros tipos de censo, para no crear una entidad de relación por cada tipo de censo, se ha optado por una generalización con la superclase censo que tiene el atributo común ámbito, que nos sirve para determinar con que tipo de censo se va a relacionar una votación concreta a través del atributo tipo\_censo.

Todo elemento de la entidad Censo ha de pertenecer a alguna de sus subclases, pero no a varias al mismo tiempo. La generalización es Total y Disjunta.

Una localidad pertenece a una sola región, y esta sólo puede pertenecer a un país y todos los países pertenecen al Censo Global que es Europa. Estos censos están relacionados entre sí a través de los atributos cod\_pais, cod\_region y cod\_localidad. Si sabemos a que localidad pertenece un ciudadano, se puede obtener la región y el país al que pertenece.

Los ciudadanos cuando se inscriben comprobamos que son mayores de 18 años, para que puedan ser electores. Un ciudadano puede participar en las elecciones Globales, Nacionales y Regionales de la Nacionalidad a la que pertenezcan y a las elecciones Municipales de la localidad donde residan. De tal forma que un ciudadano que está inscrito en el padrón de un municipio, independientemente de su nacionalidad puede votar en esa localidad. En cambio un ciudadano que no pertenezca a la nacionalidad del país donde reside, no podrá participar a las Nacionales y Regionales de ese país pero si a las de su país de origen donde este inscrito, lo que llamaríamos el voto por correo.

Para solucionar estas cuestiones y que un ciudadano que este viviendo en una localidad que no pertenezca a su país de origen no pueda votar en las elecciones de ese país hemos realizado la siguiente decisión. Para las votaciones Globales, Nacionales y Regionales los ciudadanos estarán inscritos en un censo regional perteneciente al de su Nacionalidad, mientras que para las elecciones municipales los ciudadanos estarán inscritos en el censo local del país en el que residen. Por ello en el diagrama un elector está inscrito en el Censo Local y en el Censo Regional.

Si una votación es de ámbito global, sabemos que participan todos los países de Europa, por ello cuando en una votación se defina el ámbito Global, automáticamente se tienen que habilitar para esa votación todos los Censos Nacionales.

La Clase Elector es una superclase de especialización de la Clase Candidato, ya que algunos electores además de ser electores pueden ser candidatos. Por lo que se trata de una especialización Parcial, algunos electores además pueden ser candidatos.

Una votación puede tener varias opciones a votar, entre ellas los votos en blanco, los votos nulo y los distintos partidos políticos, pero solamente la opción de los partidos políticos van a estar formadas por candidatos y van a tener unos atributos específicos de esta opción. Por ello hemos optado por otra generalización Total y Disjunta. Un partido político está formado por varios candidatos en cada votación. Se produce una relación ternaria entre las entidades votaciones, partidos políticos y candidatos N:M:1. En una votación, un partido político puede estar formado por varios candidatos. Un candidato de un partido político se puede presentar en varias votaciones y un candidato en una votación sólo se puede presentar por un partido político.

Otra decisión de diseño ha sido mantener la estructura de organización en colegios o centros electorales y mesas. En un Municipio se establecen varios centros electorales y cada centro pertenece a un solo municipio. Cada centro a su vez está compuesto por varias mesas y cada mesa pertenece a un solo centro. Esta decisión se ha tomado a raíz de considerar que aunque

no es necesario, ya que al ser un sistema telemático el voto se puede realizar desde cualquier ordenador, por la ley de igualdad de oportunidades es preciso proporcionar a todos los ciudadanos las mismas facilidades de acceso al voto, y actualmente existen hogares en toda Europa que no tienen acceso a un ordenador conectado a Internet. Por lo que se facilitarán unas cabinas a los ciudadanos que se encuentren en estas circunstancias de tal manera que puedan ejercer su derecho al voto.

Por último nos queda el voto propiamente dicho, en un voto una persona en una votación concreta elige una opción. Luego están relacionadas las entidades Opciones, votación y votantes de la siguiente forma: En una votación, una opción puede ser elegida por varios votantes, un votante una misma opción la puede elegir en distintas votaciones y por último un votante en una votación sólo puede elegir una opción N:M:1.

En el modelo E/R también tenemos en cuenta las entidades estadísticas para poder realizar las consultas en tiempo constante y la entidad Log para poder almacenar las llamadas a los procedimientos.

### 2.2.2. Supuestos Semánticos no reflejados en el Diagrama

- La fecha de publicación de una votación será siempre un mínimo de una semana antes de la fecha de inicio de la votación, entendiéndose este periodo como de reflexión.
- La fecha de inicio de una votación siempre tiene que ser anterior a la fecha fin de la votación.
- Los ciudadanos que pueden inscribirse en los Censos han de ser mayores de 18 años.
- Sólo puede haber una instancia del Censo Global
- El código del ciudadano debe ser un código aleatorio que actúe de forma de alias, para mantener el secreto de las votaciones.
- Para que un voto sea válido tiene que ser emitido entre la fecha de inicio y fin de una votación. (No se admiten votos fuera de ese periodo)
- El Estado de una votación puede ser:
  - **Edición:** Cuando se está dando de alta la votación y la fecha es anterior al periodo de reflexión, fecha de publicación.
  - **Pendiente:** de apertura: cuando la fecha está comprendida entre la fecha de publicación y la fecha de inicio.
  - **Abierta:** cuando la fecha está comprendida entre la fecha\_inicio y la fecha\_fin.
  - **Finalizada:** cuando la fecha es superior a la fecha de finalización de la votación.

- Una vez publicada la votación, no se pueden asignar nuevas opciones de votación, ni presentar candidatos nuevos en los partidos políticos de esa votación.
- Los candidatos y partidos políticos no se pueden modificar una vez que hayan participado en alguna votación que ya haya pasado el estado de edición.

### 2.2.3. Atributos de las Entidades

A continuación se detallan los atributos definidos para las entidades del diseño E/R. Para identificar la clave Primaria de la Entidad subrayaremos el atributo correspondiente.

#### **VOTACIÓN**

cod\_votacion, titulo, descripción, fecha\_inicio, fecha\_fin, fecha\_publicación, dirección \_Web, electores, votos\_totales, percent\_participacion, optmasvotada, percent\_ganadora, optmenosvotada, percent\_perdedora, presidente.

#### **AMBITO**

cod\_ambito, descripción

#### **ESTADO**

cod\_estado, nombre, descripción

#### **CENSO**

cod\_censo, tipo\_censo

#### **CENSO GLOBAL**

cod\_cglobal, nombre

#### **CENSO NACIONAL**

cod\_cnacional, cod\_pais, nom\_pais

#### **CENSO REGIONAL**

cod\_cregional, cod\_region, nom\_region

#### **CENSO LOCAL**

cod\_clocal, cod\_pais, cod\_region, cod\_localidad, nom\_localidad

#### **OPCIONES**

cod\_opcion, tipo\_opcion

#### **VOTO \_ BLANCO**

cod\_vblanco, descripcion

#### **VOTO \_ NULO**

cod\_vnulo, descripcion

**PART\_POLITICO**

cod\_partido, siglas, nombre, sede

**ELECTOR**

cod\_elector, fecha\_nacimiento

**ELECTORES**

cod\_votacion

**CANDIDATO**

cod\_candidato, nombre, apellido1, apellido2

**CENTRO\_ELECTORAL**

cod\_centro, nombre, dirección

**MESA**

cod\_centro, cod\_mesa

**EST\_VOT\_FINAL\_AÑO**

cod\_pais, anyo, vot\_finalizadas.

**EST\_CENSO\_ANYO\_PORCT**

Cod\_censo, anyo, porct\_medio\_participacion

**EST\_CENSO\_ANYO\_VOTA**

Cod\_censo, anyo, num\_votacion

**EST\_VOT\_MAS\_PARTI\_ANYO**

cod\_votacion, anyo, participacion

**EST\_LOC\_MAS\_VOTOS\_ANYO**

cod\_localidad, nom\_localidad, anyo, votos

**EST\_CIUDADANOS\_SINVOTOS\_ANYO**

anyo, porct\_sinvotar

**EST\_VOTOS\_PAIS\_PERSONA**

cod\_elector, cod\_pais, votaciones

**EST\_HISTORICO\_VOTACIONES**

cod\_votacion, porct\_participacion.

**EST\_HISTORICO\_VOTACIONES\_MAS**

cod\_votacion, porct\_masvotada.

**EST\_HISTORICO\_VOTACIONES\_MENOS**

cod\_votacion, porct\_menosvotada.

**LOGS**

cod\_log, nombre, fecha, param\_entrada, param\_salida.

## 2.3. Diseño del Modelo Lógico

El Diseño Lógico toma como referencia el modelo E/R para transformarlo en un conjunto de relaciones y claves a partir de unas reglas que se aplican sobre las entidades y sus relaciones.

Para realizar el diseño lógico tomaremos la siguiente notación:

- Las claves primarias aparecen subrayadas.
- Las claves alternativas aparecen subrayadas con una línea discontinua
- Las claves ajenas se muestran entre llaves indicando a la tabla que apuntan.
- Los atributos que pueden tomar valores nulos aparecen con un asterisco.

### 2.3.1. Modelo Relacional

**VOTACIÓN** (cod\_votacion, titulo, descripción\*, fecha\_inicio, fecha\_fin, fecha\_publicación, estado, direccionweb\*, electores, votos\_totales, porcent\_participacion, optmasvotada, porcent\_ganadora, optmenosvotada, porcent\_perdedora, ambito, presidente)  
 {estado} referencia ESTADO  
 {ambito} referencia AMBITO

**ESTADO** (cod\_estado, nombre, descripción)

**AMBITO** (cod\_ambito, descripción)

**CENSO** (cod\_censo, tipo\_censo)  
 {tipo\_censo} referencia AMBITO

**CENSOS HABILITADOS** (cod\_votacion, cod\_censo)  
 {cod\_votacion} referencia VOTACION  
 {cod\_censo} referencia CENSO

**CENSO GLOBAL** (cod\_cglobal, nombre)  
 {cod\_cglobal} referencia CENSO

**CENSO NACIONAL** (cod\_cnacional, cod\_cglobal, cod\_pais, nom\_pais)  
 {cod\_cnacional} referencia CENSO  
 {cod\_cglobal} referencia CENSO GLOBAL

**CENSO REGIONAL** (cod\_cregional, cod\_pais, cod\_region, nom\_region)  
 {cod\_cregional} referencia CENSO  
 {cod\_pais} referencia CENSO NACIONAL

**CENSO LOCAL** (cod\_clocal, cod\_pais, cod\_region, cod\_localidad, nombre)  
{cod\_clocal} referencia CENSO  
{cod\_pais, cod\_region} referencia CENSO REGIONAL

**OPCIONES** (cod\_opcion, tipo\_opcion)

**OPCIONES\_VOTACION** (cod\_votacion, cod\_opcion, votos\_opcion, percent\_participacion)  
{cod\_votacion} referencia VOTACION  
{cod\_opcion} referencia OPCIONES

**VOTO\_BLANCO** (cod\_vblanco, descripcion)  
{cod\_vblanco} referencia OPCIONES

**VOTO\_NULO** (cod\_vnulo, descripcion)  
{cod\_vnulo} referencia OPCIONES

**PART\_POLITICO** (cod\_partido, siglas, nombre, sede)  
{cod\_partido} referencia OPCIONES

**ELECTOR** (cod\_elector, cod\_cregional, cod\_clocal, fecha\_nacimiento, cod\_centro, cod\_mesa)  
{cod\_cregional} referencia CENSO REGIONAL  
{cod\_clocal} referencia CENSO LOCAL  
{cod\_centro, cod\_mesa} referencia MESA

**CANDIDATO** (cod\_candidato, nombre, apellido1, apellido2)  
{cod\_candidato} referencia ELECTOR

**CANDIDATOS\_VOTACION\_PART\_POLITICO**(cod\_candidato, cod\_votacion, cod\_partido, orden\_lista)  
{cod\_candidato} referencia CANDIDATO  
{cod\_votacion} referencia VOTACION  
{cod\_partido} referencia PARTIDO

**VOTAR** (cod\_votacion, cod\_votante, cod\_opcion)  
{cod\_votacion} referencia VOTACION  
{cod\_votante} referencia ELECTOR  
{cod\_opcion} referencia OPCIONES

**CENTRO ELECTORAL** (cod\_centro, cod\_clocal, nombre, dirección\*)  
{cod\_clocal} referencia CENSO LOCAL

**MESA** (cod\_mesa, cod\_centro)  
{cod\_centro} referencia CENTRO ELECTORAL

**EST\_VOT\_FINAL\_AÑO** (cod\_pais, anyo, vot\_finalizadas)  
{cod\_pais} referencia CENSO NACIONAL

**EST\_CENSO\_ANYO\_PORCT**(cod\_censo, anyo, porct\_medio\_participacion)  
{cod\_censo} referencia CENSO

**EST\_CENSO\_ANYO\_VOTA**(cod\_censo, anyo, num\_votacion)  
{cod\_censo} referencia CENSO

**EST\_VOT\_MAS\_PARTI\_ANYO** (cod\_votacion, anyo, participacion)  
{cod\_votacion} referencia VOTACION

**EST\_LOC\_MAS\_VOTOS\_ANYO** (cod\_localidad, nom\_localidad, anyo, votos)  
{cod\_localidad} referencia CENSO LOCAL

**EST\_CIUDADANOS\_SINVOTOS\_ANYO** (anyo, porct\_sinvotar)

**EST\_VOTAC\_PAIS\_PERSONA** (cod\_elector, cod\_pais, votaciones)  
{cod\_elector} referencia ELECTOR  
{cod\_pais} referencia CENSO NACIONAL

**EST\_HISTORICO\_VOTACIONES** (cod\_votacion, porct\_participacion)  
{cod\_votacion} referencia VOTACIONES

**EST\_HISTORICO\_MAS** (cod\_votacion, porct\_masvotada)  
{cod\_votacion} referencia VOTACIONES

**EST\_HISTORICO\_MENOS** (cod\_votacion, porct\_menosvotada)  
{cod\_votacion} referencia VOTACIONES

**LOGS** (cod\_log, nom, fecha, param\_entrada, param\_salida)

**ELECTORES** (cod\_votacion)

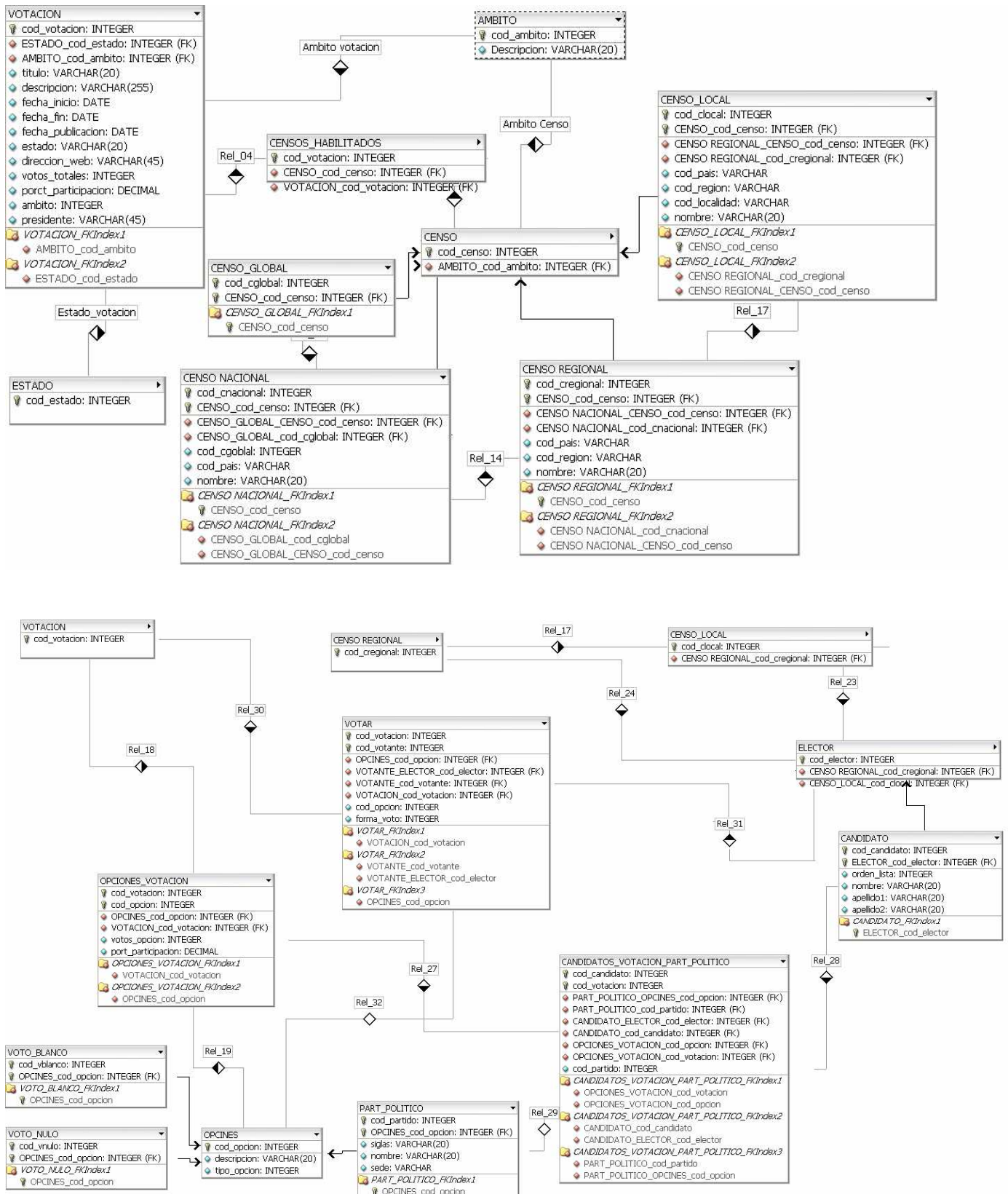
### **2.3.2. Restricciones de Integridad Referencial o clave ajena**

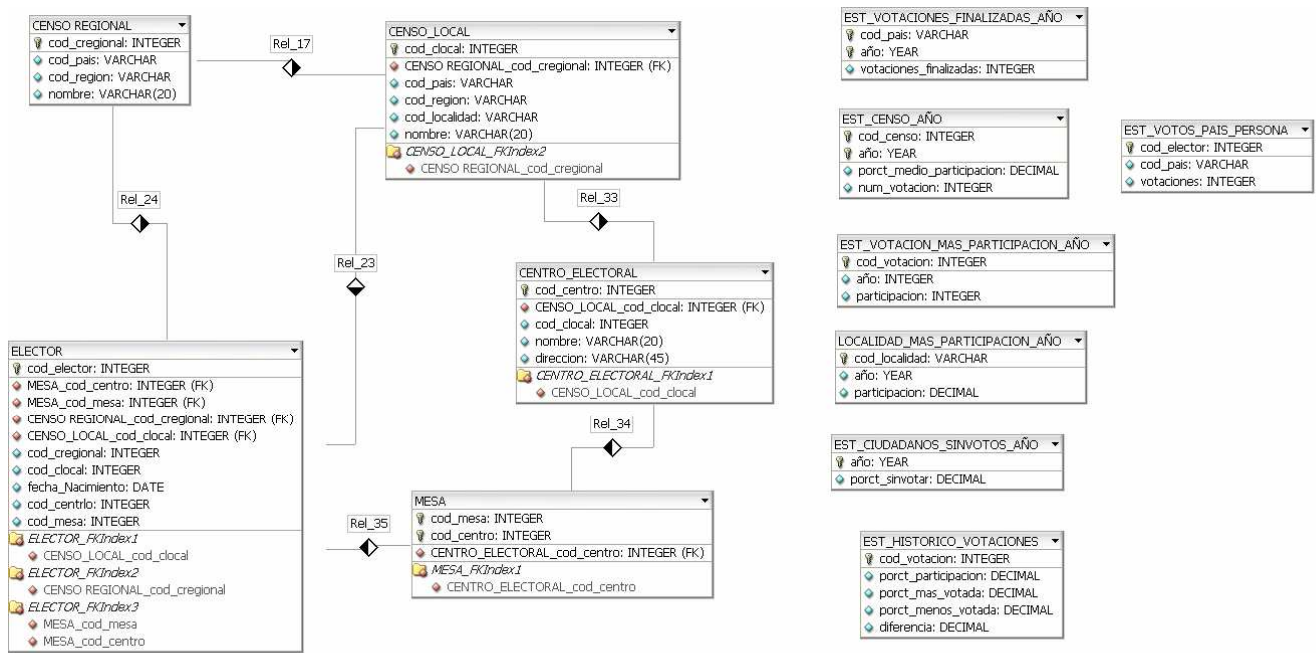
- No se puede borrar un ámbito si existen votaciones o censos de ese ámbito.
- Si se borra un censo se borran en cascada todas las tuplas relacionadas en la relación hija.
- No se puede borrar o modificar una votación si existen votos de la misma
- Si borramos una votación sin votos, borramos en cascada las opciones que se pueden votar en esa votación.
- No podemos borrar una opción de votación si esta asignada a una votación.
- Si se borra un partido político, se borran en cascada todos los candidatos de ese partido político (siempre y cuando en la votación no se hayan realizado votos).



- Si se borra un centro electoral, se borrarán en cascada las mesas asociadas a dicho centro.
- No podemos borrar una mesa si tenemos electores asignados a esa mesa.
- No se podrá borrar un censo regional o local si tiene electores asociados.

### 2.3.3. Diagrama del Modelo Relacional





### 3. Implementación de la Base de Datos

El proceso de Implementación de una Base de Datos consiste en crear las tablas y reglas necesarias para asegurar la integridad referencial así como todas aquellas consultas e informes que nos permitan visualizar la información solicitada.

En el diseño físico describimos las relaciones basándonos en sus atributos, claves primarias, claves foráneas y reglas de integridad. También se crean disparadores y procedimientos para gestionar la base de datos.

El sistema Gestor de Base de Datos que vamos a emplear es el Oracle 10g Express Edition, por lo que tendremos que adaptar el modelo lógico obtenido anteriormente al Lenguaje PL/SQL

Para crear las tablas y trigger así como las secuencias de la base de datos ejecutaremos el script *tablas*. Una vez implementados los elementos anteriores tendríamos que implementar los procedimientos de la base de datos de las elecciones ejecutando el script *procedimientos*. Por último quedaría ejecutar el script *Test\_General* para poder proceder a la utilización del programa, incluyendo datos de ejemplo del mismo al tiempo que se realiza una prueba de test de todos los procedimientos, consultas y estadísticas de la aplicación..

A continuación se explica para que sirve cada uno de los elementos implementados en la base de datos:

#### 3.1.Tablas Generales

- ESTADO: Almacena las distintas fases en las que puede estar una votación: edición, pendiente, abierta, finalizada.
- AMBITO: Almacena los distintos ámbitos que puede adoptar tanto los censos electorales como las votaciones. Global: Toda Europa, Nacional: Uno o varios países. Regional: Una o varias regiones y Local: Una o varias localidades, ciudades o pueblos.
- VOTACION: Almacena las distintas votaciones con sus datos correspondientes.
- CENSO: Almacena los distintos censos que se pueden habilitar en una votación.
- CENSOS\_HABILITADOS: Permite relacionar las votaciones con los censos que se pretenden habilitar para cada votación.
- CENSO\_GLOBAL: Almacena el censo global que se habilita al dar de alta una votación de ámbito Global.
- CENSO\_NACIONAL: Almacena los distintos censos de ámbito Nacional, Países.
- CENSO\_REGIONAL: Almacena los distintos censos de ámbito regional, de cada uno de los países del censo nacional.
- CENSO\_LOCAL: Almacena los distintos censos de ámbito local, de cada una de las regiones del Censo Regional.

- OPCIONES: Almacena las distintas opciones que se pueden votar en todas las elecciones.
- OPCIONES\_VOTACION: Relaciona cada votacion con las opciones que se pueden votar para cada votacion en concreto.
- VOTO\_BLANCO: Almacena los datos relativos a la opción Voto en Blanco.
- VOTO\_NULO: Almacena los datos relativos a la opción Voto Nulo.
- PART\_POLITICO: Almacena los distintos partidos políticos que se pueden presentar a las distintas votaciones y sus datos relativos.
- CENTRO\_ELECTORAL: Almacena los distintos Centros Electorales asociados a cada localidad, en los cuales se habilitaran las mesas electorales.
- MESA: Almacena las mesas electorales asociadas a cada Centro Electoral, donde los ciudadanos pueden acercarse a votar.
- ELECTOR: Almacena los datos de los ciudadanos de la Unión Europa que están legitimados para votar.
- CANDIDATO: Almacena los datos de los electores que se presentan a las distintas elecciones.
- CANDIDATOS\_VOTACION\_PARTIDO: Permite relacionar en una votacion, los partidos políticos que se presentan a las mismas así como los candidatos del partido y su posición en las listas electorales.
- VOTAR: Almacena los votos que realizan los ciudadanos en cada votación, a los distintos partidos políticos, voto en blanco o voto nulo. Relacionando las tablas votacion, opciones y elector.

### **3.2.Tablas del Modulo Estadístico**

- EST\_VOT\_FINAL\_ANYO: Almacena el número de votaciones finalizadas que han tenido lugar por un país por año.
- EST\_CENSO\_ANYO\_PORCT: Almacena el valor medio del porcentaje de participación, teniendo en cuenta todas las votaciones asociadas a cada censo electoral finalizadas por cada año.
- EST\_CENSO\_ANYO\_VOTA: Almacena el número de votaciones teniendo en cuenta todas las votaciones asociadas a cada censo electoral finalizadas por cada año.
- EST\_VOT\_MAS\_PARTI\_ANYO: Almacena las votaciones que más participación han tenido por cada año.

- EST\_LOC\_MAS\_VOTOS\_ANYO: Almacena por cada año, la localidad Europea que más votos han emitido sus ciudadanos.
- EST\_CIUDADANOS\_SINVOTOS\_ANYO: Almacena por cada año el porcentaje de ciudadanos que nunca han votado.
- EST\_VOTAC\_PAIS\_PERSONA: Almacena el número máximo de votaciones en las que se ha participado por persona, indicando el país al que pertenece.
- EST\_HISTORICO\_VOTACIONES: Almacena la votación que ha tenido mayor diferencia porcentual entre la opción más votada y la menos históricamente.
- EST\_HISTORICO\_MAS: Almacena la votación que ha tenido menor diferencia porcentual entre la opción más votada y la menos históricamente.
- EST\_HISTORICO\_MENOS: Almacena la votación que ha tenido mayor participación históricamente.

### 3.3.Otras Tablas

- LOGS: Nos permite almacenar los logs de toda la aplicación.
- ELECTORES: Tabla auxiliar donde se almacenan las votaciones que se deben actualizar al dar de alta, baja o modificación un elector en los censos electorales.

### 3.4.Secuencias

- SEQ\_ESTADO: Secuencia para generar los códigos de identificación de la tabla Estado.
- SEQ\_AMBITO: Secuencia para generar los códigos de identificación de la tabla Ambito.
- SEQ\_VOTACION: Secuencia para generar los códigos de identificación de la tabla Votación.
- SEQ\_CENSO: Secuencia para generar los códigos de identificación de la tabla Censo.
- SEQ OPCIONES: Secuencia para generar los códigos de identificación de la tabla Opciones.
- SEQ\_CENTRO\_ELECTORAL: Secuencia para generar los códigos de identificación de la tabla Centro\_Electoral.

### 3.5.Triggers

- **INSERT\_ESTADO:** Asigna un nuevo código, el siguiente número de la secuencia Seq\_Estado, en la tabla Estado al insertar un nuevo registro en la tabla.
- **INSERT\_AMBITO:** Asigna un nuevo código, el siguiente número de la secuencia Seq\_Ambito, en la tabla Ambito al insertar un nuevo registro en la tabla.
- **INSERT\_VOTACION:** Asigna un nuevo código, el siguiente número de la secuencia Seq\_Votacion, en la tabla Votación al insertar un nuevo registro en la tabla.
- **CENSO\_GLOBAL:** Habilita los censos nacionales en una votación cuando al dar de alta la misma, el ámbito es global o en el caso de que al modificar el ámbito de la votación se cambie al ámbito global.
- **INSERT\_CENSO\_GLOBAL:** Asigna un nuevo código, el siguiente número de la secuencia Seq\_Censo, en las tablas Censo y Censo\_Global al insertar un nuevo registro en la tabla Censo\_Global.
- **INSERT\_CENSO\_NACIONAL:** Asigna un nuevo código, el siguiente número de la secuencia Seq\_Censo, en las tablas Censo y Censo\_Nacional al insertar un nuevo registro en la tabla Censo\_Nacional.
- **INSERT\_CENSO\_REGIONAL:** Asigna un nuevo código, el siguiente número de la secuencia Seq\_Censo, en las tablas Censo y Censo\_Regional al insertar un nuevo registro en la tabla Censo\_Regional.
- **INSERT\_CENSO\_LOCAL:** Asigna un nuevo código, el siguiente número de la secuencia Seq\_Censo, en las tablas Censo y Censo\_Local al insertar un nuevo registro en la tabla Censo\_Nacional.
- **INSERT\_VOTO\_BLANCO:** Asigna un nuevo código, el siguiente número de la secuencia Seq\_Opciones, en las tablas Opciones y Voto\_Blanco al insertar un nuevo registro en la tabla Voto\_Blanco.
- **INSERT\_VOTO\_NULO:** Asigna un nuevo código, el siguiente número de la secuencia Seq\_Opciones, en las tablas Opciones y Voto\_Nulo al insertar un nuevo registro en la tabla Voto\_Nulo.
- **INSERT\_PART\_POLITICO:** Asigna un nuevo código, el siguiente número de la secuencia Seq\_Opciones, en las tablas Opciones y Part\_Politico al insertar un nuevo registro en la tabla Part\_Politico.
- **INSERT\_CENTROELECTORAL:** Asigna un nuevo código, el siguiente número de la secuencia Seq\_CentroElectoral, en la tabla Centro\_Electoral al insertar un nuevo registro en la tabla.
- **INSERT\_LOGS:** Asigna un nuevo código, el siguiente número de la secuencia Seq\_Logs, en la tabla Logs al insertar un nuevo registro en la tabla.

### 3.6. Vistas

- PAISCENSO: Consulta que nos permite obtener los códigos de país correspondientes a cada código de censo, ya sea local, regional o nacional.
- VOTACENSO: Consulta que nos permite obtener los censos habilitados correspondientes a cada votación así como la fecha de inicio, porcentaje de participación y estado de la votación.
- VOTALOCAL: Consulta que nos permite obtener el número de votos emitidos de cada localidad según el año.
- CENSOSELECTOR: Consulta que nos permite obtener los censos a los que pertenece un elector así como el código de país correspondiente a su nacionalidad.
- MAXVOTACION: Consulta que nos permite obtener el código de elector que tiene el máximo número de votaciones en las que ha participado una misma persona.

### 3.7. Procedimientos Almacenados

En el siguiente apartado se va a proceder a describir la función del procedimiento a Alto nivel, el tipo y los valores posibles tanto de los parámetros de entrada del procedimiento como de los de salida incluyendo los distintos mensajes de error que pueden devolver.

Para ello cada procedimiento se va describir en tres apartados.

Funcionalidad: Explica en alto nivel la función del procedimiento.

Parámetros: Indicamos los parámetros de entrada y salida del procedimiento así como el mensaje que devuelven. Para ello se empleará el siguiente formato de tabla:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	Tipo de dato	Entrada o Salida	Identificación/Mensaje

La tabla utiliza la siguiente sintaxis:

Nombre: Nombre del Parámetro

Tipo: Tipo de dato del Parámetro:

E/S: E si es parámetro de Entrada, S si es parámetro de salida y ES si es parámetro de Entrada y Salida.

Descripción: Explicación detallada del funcionamiento del procedimiento.

### 3.7.1. INSERTAR LOG

#### Funcionalidad:

El procedimiento p\_insertarlog permite insertar un nuevo registro en la tabla logs con información sobre el procedimiento que se está ejecutando.

#### Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	<b>Tipo de dato</b>	<b>Entrada o Salida</b>	Identificación/Mensaje
Nombre_proc	Integer	E	Nombre del procedimiento
Fecha	Date	E	Fecha ejecución procedimiento
Paramentrada	Varchar2	E	Parámetros de entrada
Paramsalida	Varchar2	E	Parámetros de salida

#### Descripción:

Se realiza un insert en la tabla logs con los datos que recibimos. Se confirman los cambios mediante el commit.

En caso de Error se muestra un mensaje por pantalla con la descripción del error.

### 3.7.2. ALTA ESTADO

#### Funcionalidad:

El procedimiento p\_altaestado permite dar de alta los distintos estados de las elecciones. Un trigger se encarga de asignar un nuevo código de estado automáticamente

#### Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	<b>Tipo de dato</b>	<b>Entrada o Salida</b>	Identificación/Mensaje
Nom	Varchar2	E	Estado de la elección
Des	Varchar2	E	Descripción del Estado
RSP	Varchar2	S	Error + Descripción del error

#### Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobar los parámetros obligatorios, confirmando que poseen datos en caso contrario el programa lanzará un error.

Comprobar que los estados que se dan de alta corresponde a los siguientes valores: edicion, pendiente, abierta, finalizada, en caso contrario el programa lanzará un error.



Realizar un Insert en la tabla Estado con los datos pasados como parámetros de entrada.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.7.3. ALTA AMBITO

#### Funcionalidad:

El procedimiento p\_altambito permite dar de alta los distintos ambitos de las elecciones. Un trigger se encarga de asignar un nuevo código de ambito automáticamente

#### Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro des	<b>Tipo de dato</b> Varchar2	<b>Entrada o Salida</b> E	Identificación/Mensaje Nombre del ambito
RSP	Varchar2	S	Error + Descripción del error

#### Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobar los parámetros obligatorios, confirmando que poseen datos en caso contrario el programa lanzará un error.

Comprobar que los ámbitos que se dan de alta corresponde a los siguientes valores: global, nacional, regional, local en caso contrario el programa lanzará un error.

Realizar un Insert en la tabla Ambito con los datos pasados como parámetros de entrada.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.7.4. ALTA CENSO GLOBAL

#### Funcionalidad:

El procedimiento p\_altacensoglobal permite dar de alta un solo censo global o Europeo. Un trigger se encarga de asignar un nuevo código de censo global automáticamente en las tablas Censo\_Global y Censo.

#### Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	<b>Tipo de dato</b>	<b>Entrada o Salida</b>	Identificación/Mensaje

RSP	Varchar2	S	Error + Descripción del error
-----	----------	---	-------------------------------

Descripción:

Se inicializa los parámetros de entrada a Null al no recibir ninguno, para insertarlos en el log.

Realizar un Insert en la tabla Censo\_Global con el valor Europa para el campo “Nombre” de la tabla.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

**3.7.5. ALTA CENSO NACIONAL**Funcionalidad:

El procedimiento p\_altacensonal permite dar de alta todos los censos nacionales de Europa. Un trigger se encarga de asignar un nuevo código de censo nacional automáticamente en las tablas Censo\_Nacional y Censo.

Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro codigo	<b>Tipo de dato</b> Varchar2	<b>Entrada o Salida</b> E	Identificación/Mensaje Código del País
RSP	Varchar2	S	Error + Descripción del error

Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobar los parámetros obligatorios, confirmando que poseen datos en caso contrario el programa lanzará un error.

Realizar un Insert en la tabla Censo\_Nacional con los datos pasados como parámetros de entrada, que tienen que cumplir la norma Iso 3166-1 alfa2, rellenando automáticamente al pasar el código del país, el nombre del mismo.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

**3.7.6. ALTA CENSO REGIONAL**Funcionalidad:

El procedimiento `p_altacensoregional` permite dar de alta los censos regionales de las elecciones. Un trigger se encarga de asignar un nuevo código de censo regional automáticamente en las tablas `Censo_Regional` y `Censo`.

Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	<b>Tipo de dato</b>	<b>Entrada o Salida</b>	Identificación/Mensaje
Pais	Varchar2	E	Código del país
codigo	Varchar2	E	Código de la región
Nombre	Varchar2	E	Nombre de la región
RSP	Varchar2	S	Error + Descripción del error

Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Se comprueba que el país ha sido dato de alta previamente en caso contrario el programa lanzará un error.

Realizar un Insert en la tabla `Censo_Regional` con los datos pasados como parámetros de entrada.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.7.7. MODIFICACIÓN CENSO REGIONAL

Funcionalidad:

El procedimiento `p_editacensoregional` permite modificar un censo regional existente. Para ello hay que pasar como parámetro el código del censo regional y cualquiera de los siguientes campos: país, región o nombre. Los campos se deben pasar en el orden preestablecido. Si no se quiere modificar un campo se puede enviar vacío si es una cadena o con un cero si es un número.

Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	<b>Tipo de dato</b>	<b>Entrada o Salida</b>	Identificación/Mensaje
Cregional	Varchar2	E	Código del Censo regional
Pais	Varchar2	E	Código del país
region	Varchar2	E	Código de la región
nombre	Varchar2	E	Nombre de la región
RSP	Varchar2	S	Error + Descripción del error

Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobar los parámetros obligatorios, confirmando que poseen datos en caso contrario el programa lanzará un error.

Se comprueba que el censo regional ha sido dato de alta previamente en caso contrario el programa lanzará un error.

Comprobar que no hay electores asignados al censo regional, en caso contrario no dejaremos modificar el censo lanzando el programa un error.

Realizar un Update en la tabla Censo\_Regional con los datos pasados como parámetros de entrada.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.7.8. BAJA CENSO REGIONAL

#### Funcionalidad:

El procedimiento p\_bajacensoregional permite dar de baja un censo regional existente siempre y cuando no existan electores inscritos en dicho censo o en los censos locales dependientes de la región y no estén habilitados en una votación en estado de edición o pendiente. Al dar de baja un censo regional se eliminarán sus registros relacionados en las tablas censo y censo\_local.

#### Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro V_region	<b>Tipo de dato</b> Integer	<b>Entrada o Salida</b> E	Identificación/Mensaje Código del Censo regional
RSP	Varchar2	S	Error + Descripción del error

#### Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Se comprueba que el censo regional ha sido dato de alta previamente en caso contrario el programa lanzará un error.

Se comprueba que no existen votaciones, con el censo regional o el censo nacional correspondiente a esa región, habilitado en estado de edición o pendiente, en caso contrario no dejaremos eliminar el censo regional.

Realizar un Delete en la tabla Censo, que borra automáticamente el registro relacionado en la tabla Censo\_Regional y los registros relacionados en la tabla Censo\_Local.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.7.9. ALTA CENSO LOCAL

#### Funcionalidad:

El procedimiento p\_altacensolocal permite dar de alta los censos locales de las elecciones. Un trigger se encarga de asignar un nuevo código de censo local automáticamente en las tablas Censo\_Local y Censo.

#### Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	<b>Tipo de dato</b>	<b>Entrada o Salida</b>	Identificación/Mensaje
Pais	Varchar2	E	Código del país
Región	Varchar2	E	Código de la región
Código	Varchar2	E	Código del Censo local
Nombre	Varchar2	E	Nombre de la localidad
RSP	Varchar2	S	Error + Descripción del error

#### Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Se comprueba que el país ha sido dato de alta previamente en caso contrario el programa lanzará un error.

Se comprueba que la región ha sido dato de alta previamente en caso contrario el programa lanzará un error.

Comprobar que la región pertenece al país en caso contrario el programa lanzará un mensaje de error.

Realizar un Insert en la tabla Censo\_Local con los datos pasados como parámetros de entrada.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.7.10. MODIFICACIÓN CENSO LOCAL

#### Funcionalidad:

El procedimiento p\_editacensolocal permite modificar un censo local existente. Para ello hay que pasar como parámetro el código del censo local y cualquiera de los siguientes campos: país, región, localidad, nombre. Los campos se deben pasar en el orden preestablecido. Si no se quiere modificar un campo se puede enviar vacío si es una cadena o con un cero si es un número.

Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro clocal	<b>Tipo de dato</b> Integer	<b>Entrada o Salida</b> E	Identificación/Mensaje Código del Censo local
pais	Varchar2	E	Código del país
region	Varchar2	E	Código de la región
localidad	Varchar2	E	Código de la localidad
nombre	Varchar2	E	Nombre de la localidad
RSP	Varchar2	S	Error + Descripción del error

Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobar los parámetros obligatorios, confirmando que poseen datos en caso contrario el programa lanzará un error.

Se comprueba que el censo local ha sido dato de alta previamente en caso contrario el programa lanzará un error.

Comprobar que no hay electores asignados al censo local, en caso contrario no dejaremos modificar el censo lanzando el programa un error.

Comprobar que no hay Centros electorales asignados al censo local, en caso contrario no dejaremos modificar el censo lanzando el programa un error.

Realizar un Update en la tabla Censo\_Local con los datos pasados como parámetros de entrada.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

**3.7.11. BAJA CENSO LOCAL**Funcionalidad:

El procedimiento p\_bajacensolocal permite dar de baja un censo local existente siempre y cuando no existan electores inscritos en dicho censo y no estén habilitados en una votación en estado de edición o pendiente. Al dar de baja un censo local se eliminarán sus registros relacionados en las tablas Censo, Censo\_Local y Centro\_Electoral.

Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro V_clocal	<b>Tipo de dato</b> Integer	<b>Entrada o Salida</b> E	Identificación/Mensaje Código del Censo regional

RSP	Varchar2	S	Error + Descripción del error
-----	----------	---	-------------------------------

Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Se comprueba que el censo local ha sido dato de alta previamente en caso contrario el programa lanzará un error.

Se comprueba que no existen votaciones, con el censo local habilitado en estado de edición o pendiente, en caso contrario no dejaremos eliminar el censo local.

Realizar un Delete en la tabla Censo, que borra automáticamente el registro relacionado en la tabla Censo\_Local y los registros relacionados en la tabla Centro\_Electoral.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

**3.7.12. ALTA CENTRO ELECTORAL**Funcionalidad:

El procedimiento p\_altacentroelectoral permite dar de alta los centros electorales de las elecciones. Un trigger se encarga de asignar un nuevo código de centro electoral automáticamente.

Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	<b>Tipo de dato</b>	<b>Entrada o Salida</b>	Identificación/Mensaje
Localidad	Integer	E	Código del Censo Electoral
Nombre	Varchar2	E	Nombre del Centro
Dirección	Varchar2	E	Dirección del Centro
RSP	Varchar2	S	Error + Descripción del error

Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Se comprueba que el censo local ha sido dato de alta previamente en caso contrario el programa lanzará un error.

Realizar un Insert en la tabla Centro\_Electoral con los datos pasados como parámetros de entrada.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.7.13. MODIFICACIÓN CENTRO ELECTORAL

#### Funcionalidad:

El procedimiento p\_editacentroelectoral permite modificar un centro electoral existente. Para ello hay que pasar como parámetro el código del centro electoral y cualquiera de los siguientes campos: localidad, enombre, edirección. Los campos se deben pasar en el orden preestablecido. Si no se quiere modificar un campo se puede enviar vacío si es una cadena o con un cero si es un número.

#### Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	<b>Tipo de dato</b>	<b>Entrada o Salida</b>	Identificación/Mensaje
Centro	Number	E	Código del Centro electoral
Localidad	Integer	E	Código del censo local
Enombre	Varchar2	E	Nombre del centro
edireccion	Varchar2	E	Dirección del centro
RSP	Varchar2	S	Error + Descripción del error

#### Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobar los parámetros obligatorios, confirmando que poseen datos en caso contrario el programa lanzará un error.

Se comprueba que el centro electoral ha sido dato de alta previamente en caso contrario el programa lanzará un error.

Realizar un Update en la tabla Centro\_Electoral con los datos pasados como parámetros de entrada.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.7.14. BAJA CENTRO ELECTORAL

#### Funcionalidad:

El procedimiento p\_bajacentro permite dar de baja un centro electoral existente siempre y cuando no existan electores inscritos en las mesas electorales dependientes de ese centro. Al dar de baja un centro electoral se eliminarán sus registros relacionados en la tabla Mesa.

#### Parámetros:



NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro V_centro	<b>Tipo de dato</b> Number	<b>Entrada o Salida</b> E	Identificación/Mensaje Código del Centro electoral
RSP	Varchar2	S	Error + Descripción del error

Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Se comprueba que el centro electoral ha sido dato de alta previamente en caso contrario el programa lanzará un error.

Realizar un Delete en la tabla Centro\_electoral, que borra automáticamente los registros relacionados en la tabla Mesa.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

**3.7.15. ALTA MESA ELECTORAL**Funcionalidad:

El procedimiento p\_altamesa permite dar de alta las mesas electorales de las elecciones.

Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro Centro mesa	<b>Tipo de dato</b> Number Varchar2	<b>Entrada o Salida</b> E E	Identificación/Mensaje Código del Centro electoral Código de la mesa
RSP	Varchar2	S	Error + Descripción del error

Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Se comprueba que el Centro electoral ha sido dato de alta previamente en caso contrario el programa lanzará un error.

Realizar un Insert en la tabla Mesa con los datos pasados como parámetros de entrada.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

**3.7.16. BAJA MESA ELECTORAL**Funcionalidad:

El procedimiento p\_bajamesa permite dar de baja una mesa electoral existente siempre y cuando no existan electores inscritos en dicho mesa.

Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	<b>Tipo de dato</b>	<b>Entrada o Salida</b>	Identificación/Mensaje
V_Centro	Integer	E	Código del Centro Electoral
V_mesa	Varchar2	E	Código de la Mesa
RSP	Varchar2	S	Error + Descripción del error

Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Se comprueba que el Centro electoral ha sido dato de alta previamente en caso contrario el programa lanzará un error.

Realizar un Delete en la tabla Mesa.

Se comprueba que la consulta ha encontrado el registro solicitado, sino se lanza un mensaje de error.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.7.17. ALTA VOTO NULO

Funcionalidad:

El procedimiento p\_altavotonulo permite dar de alta la opción de voto nulo de las elecciones. Un trigger se encarga de asignar un nuevo código de voto nulo automáticamente en las tablas Voto\_Nulo y Opciones.

Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	<b>Tipo de dato</b>	<b>Entrada o Salida</b>	Identificación/Mensaje
RSP	Varchar2	S	Error + Descripción del error

Descripción:

Se inicializa los parámetros de entrada a Null al no recibir ninguno, para insertarlos en el log.

Realizar un Insert en la tabla Voto\_Nulo con la descripción de dicha opción.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.7.18. ALTA VOTO EN BLANCO

#### Funcionalidad:

El procedimiento p\_altavotoblanco permite dar de alta la opción de voto en blanco de las elecciones. Un trigger se encarga de asignar un nuevo código de voto en blanco automáticamente en las tablas Voto\_Blanco y Opciones.

#### Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	<b>Tipo de dato</b>	<b>Entrada o Salida</b>	Identificación/Mensaje
RSP	Varchar2	S	Error + Descripción del error

#### Descripción:

Se inicializa los parámetros de entrada a Null al no recibir ninguno, para insertarlos en el log.

Realizar un Insert en la tabla Voto\_Blanco con la descripción de dicha opción.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.7.19. ALTA PARTIDO POLITICO

#### Funcionalidad:

El procedimiento p\_altapartido permite dar de alta las distintas opciones de los partidos políticos de las elecciones. Un trigger se encarga de asignar un nuevo código de partido automáticamente en las tablas Part\_Politico y Opciones.

#### Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	<b>Tipo de dato</b>	<b>Entrada o Salida</b>	Identificación/Mensaje
Siglas	Varchar2	E	Siglas del Partido político
Nombre	Varchar2	E	Nombre del Partido político
Sede	Varchar2	E	Sede del Partido político
RSP	Varchar2	S	Error + Descripción del error

#### Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobar los parámetros obligatorios, confirmando que poseen datos en caso contrario el programa lanzará un error.

Realizar un Insert en la tabla Part\_Politico con los datos pasados como parámetros de entrada.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.7.20. MODIFICACIÓN PARTIDO POLITICO

#### Funcionalidad:

El procedimiento p\_editapartido permite modificar un partido político existente siempre y cuando no este asociado a una votación que ya haya sido publicada. Para ello hay que pasar como parámetro el código del partido y cualquiera de los siguientes campos: v\_siglas, v\_nombre, v\_sede. Los campos se deben pasar en el orden preestablecido. Si no se quiere modificar un campo se puede enviar vacío si es una cadena o con un cero si es un número.

#### Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro Partido	<b>Tipo de dato</b> Integer	<b>Entrada o Salida</b> E	Identificación/Mensaje Código del Partido político
V_siglas	Varchar2	E	Siglas de Partido político
V_nombre	Varchar2	E	Nombre del Partido político
V_sede	Varchar2	E	Sede del Partido político
RSP	Varchar2	S	Error + Descripción del error

#### Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobar los parámetros obligatorios, confirmando que poseen datos en caso contrario el programa lanzará un error.

Se comprueba que el partido político ha sido dato de alta previamente en caso contrario el programa lanzará un error.

Comprobar si existe alguna votación en la que haya participado el partido y la votación ya haya sido publicada, en ese caso se lanza un mensaje de error.

Realizar un Update en la tabla Part\_Politico con los datos pasados como parámetros de entrada.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.7.21. BAJA PARTIDO POLÍTICO

#### Funcionalidad:

El procedimiento p\_bajapartido permite dar de baja un partido político existente siempre y cuando no estén asignados a una votación ya publicada. Al dar de baja un partido político se eliminará su registro relacionado en la tabla Opciones.

#### Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro V_partido	<b>Tipo de dato</b> Integer	<b>Entrada o Salida</b> E	Identificación/Mensaje Código del Partido político
RSP	Varchar2	S	Error + Descripción del error

#### Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobar los parámetros obligatorios, confirmando que poseen datos en caso contrario el programa lanzará un error.

Se comprueba que el partido político ha sido dato de alta previamente en caso contrario el programa lanzará un error.

Comprobar que el partido político no está asignado a ninguna votación ya publicada, en ese caso no dejaremos eliminar el partido y se lanzará un mensaje de error.

Realizar un Delete en la tabla Opciones, que borra automáticamente el registro relacionado en la tabla Part\_Politico.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.7.22. ALTA ELECTOR

#### Funcionalidad:

El procedimiento p\_altaelector permite dar de alta los electores que participan en las elecciones, a la vez que se actualiza el número de electores en las votaciones en estado de edición o publicación relacionadas con los censos en los que se inscribe el ciudadano. El código del elector se asigna mediante el empleo del procedimiento p\_codigo, que nos devuelve un código aleatorio que posteriormente el usuario lo utilizará para dar de alta un elector.

#### Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro Elector	<b>Tipo de dato</b> Varchar2	<b>Entrada o Salida</b> E	Identificación/Mensaje Código Aleatorio del Elector
Región	Integer	E	Código Censo Regional para elecciones ámbito Europeo, Nacional y Regional
Localidad	Integer	E	Código Censo Local para elecciones de ámbito local
fechanac	Date	E	Fecha de Nacimiento
Centro	Number	E	Centro Electoral
Mesa	Varchar2	E	Mesa Electoral
RSP	Varchar2	S	Error + Descripción del error

Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobamos que el elector es mayor de 18 años para que tenga acceso a participar en las elecciones.

Comprobamos que la región y la localidad existen. Así mismo comprobamos que si la región y localidad donde se inscribe el ciudadano son del mismo país la localidad debe pertenecer a la región.

En cualquiera de las comprobaciones anteriores sino se cumplen los requisitos el programa envía un mensaje de error.

Realizar un Insert en la tabla Elector con los datos pasados como parámetros de entrada.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

Ejecutamos el procedimiento p\_inelector, cuya funcionalidad es añadir o restar según el caso un elector en todas aquellas votaciones en edición o publicadas tanto Europeas, Nacionales y regionales a las que pertenece el censo regional y aquellas Locales a las que pertenece el censo Local. (En el caso de alta del Elector suma un elector)

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

**3.7.23. MODIFICACIÓN ELECTOR**Funcionalidad:

El procedimiento p\_editaelector permite modificar un elector existente, a la vez que se actualiza el número de electores en las votaciones en estado de edición o publicación relacionadas con los censos en los que se inscribe el ciudadano. Para ello hay que pasar como parámetro el código del elector y cualquiera de los siguientes campos: v\_elector, cregional, clocal, nacimiento, centro, mesa. Los campos se deben pasar en el orden preestablecido. Si no se quiere modificar un campo se puede enviar vacío si es una cadena o con un cero si es un número.

Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	<b>Tipo de dato</b>	<b>Entrada o Salida</b>	Identificación/Mensaje
V_elector	Integer	E	Código Aleatorio del Elector
Cregional	Integer	E	Código Censo Regional para elecciones ámbito Europeo, Nacional y Regional
Clocal	Integer	E	Código Censo Local para elecciones de ámbito local
Nacimiento	Date	E	Fecha de Nacimiento
Centro	Number	E	Código del Centro
Mesa	Varchar2	E	Código de la Mesa
RSP	Varchar2	S	Error + Descripción del error

### Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobar los parámetros obligatorios, confirmando que poseen datos en caso contrario el programa lanzará un error.

Se comprueba que el Código de Elector ha sido dato de alta previamente en caso contrario el programa lanzará un error.

Comprobamos que el elector es mayor de 18 años para que tenga acceso a participar en las elecciones.

Comprobamos que si la región y localidad donde se inscribe el ciudadano son del mismo país la localidad debe pertenecer a la región.

En cualquiera de las comprobaciones anteriores sino se cumplen los requisitos el programa envía un mensaje de error.

Realizar un Update en la tabla Elector con los datos pasados como parámetros de entrada.

Ejecutamos el procedimiento p\_inelector, cuya funcionalidad es añadir o restar según el caso un elector en todas aquellas votaciones en edición o publicadas tanto Europeas, Nacionales y regionales a las que pertenece el censo regional y aquellas Locales a las que pertenece el censo Local. En el caso de Modificación del Elector ejecutamos el procedimiento p\_inelector 2 veces.

1. p\_inelector (Se ejecuta para eliminar un elector en las votaciones relacionadas con el censo regional y local anterior a la modificación).
2. p\_inelector (Se ejecuta para añadir un elector en las votaciones relacionadas con el censo regional y local posterior a la modificación).

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.7.24. BAJA ELECTOR

#### Funcionalidad:

El procedimiento p\_bajaelector permite dar de baja un elector existente, a la vez que se actualiza el número de electores en las votaciones en estado de edición o publicación relacionadas con los censos en los que se inscribe el ciudadano.

#### Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro V_elector	<b>Tipo de dato</b> Varchar2	<b>Entrada o Salida</b> E	Identificación/Mensaje Código aleatorio de Elector
RSP	Varchar2	S	Error + Descripción del error

#### Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobar los parámetros obligatorios, confirmando que poseen datos en caso contrario el programa lanzará un error.

Se comprueba que el elector ha sido dato de alta previamente en caso contrario el programa lanzará un error.

Realizar un Delete en la tabla Elector, que borra automáticamente el registro relacionado en la tabla Part\_Politico.

Ejecutamos el procedimiento p\_inelector, cuya funcionalidad es añadir o restar según el caso un elector en todas aquellas votaciones en edición o publicadas tanto Europeas, Nacionales y regionales a las que pertenece el censo regional y aquellas Locales a las que pertenece el censo Local. (En el caso Eliminación del Elector resta un elector).

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.7.25. ALTA CODIGO ALEATORIO ELECTOR

#### Funcionalidad:

El procedimiento p\_codigo permite obtener un código aleatorio para poder utilizarlo posteriormente en el procedimiento p\_altaelector, como identificador del elector.



Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	<b>Tipo de dato</b>	<b>Entrada o Salida</b>	Identificación/Mensaje
RSP	Varchar2	S	Error + Descripción del error

Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Obtenemos un código aleatorio mediante la función Random.String.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

**3.7.26. ALTA CANDIDATO**Funcionalidad:

El procedimiento p\_altacandidato permite dar de alta los candidatos que se presentan a los distintos partidos políticos en una votación.

Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	<b>Tipo de dato</b>	<b>Entrada o Salida</b>	Identificación/Mensaje
Candidato	Varchar2	E	Código del Candidato
Nombre	Varchar2	E	Nombre del Candidato
Ape1	Varchar2	E	Apellido primero del Candidato
Ape2	Varchar2	E	Apellido segundo del Candidato
RSP	Varchar2	S	Error + Descripción del error

Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobar que el candidato es elector, requisito indispensable para poder presentarse en una candidatura, en caso de no cumplir dicho requisito se envía un mensaje de error.

Realizar un Insert en la tabla Candidato con los datos pasados como parámetros de entrada.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.7.27. MODIFICACIÓN CANDIDATO

#### Funcionalidad:

El procedimiento p\_editacandidato permite modificar un candidato a un partido político en una votación existente siempre y cuando no este asociado a una votación que ya haya sido publicada. Para ello hay que pasar como parámetro el código del candidato y cualquiera de los siguientes campos: v\_nombre, ape1, ape2. Los campos se deben pasar en el orden preestablecido. Si no se quiere modificar un campo se puede enviar vacío si es una cadena o con un cero si es un número.

#### Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	<b>Tipo de dato</b>	<b>Entrada o Salida</b>	Identificación/Mensaje
V_candidato	Varchar2	E	Código del Candidato
V_nombre	Varchar2	E	Nombre del Candidato
Ape1	Varchar2	E	Apellido primero del Candidato
Ape2	Varchar2	E	Apellido segundo del Candidato
RSP	Varchar2	S	Error + Descripción del error

#### Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobar los parámetros obligatorios, confirmando que poseen datos en caso contrario el programa lanzará un error.

Se comprueba que el candidato ha sido dato de alta previamente en caso contrario el programa lanzará un error.

Comprobar si existe alguna votación en la que haya participado el candidato y la votación ya haya sido publicada, en ese caso se lanza un mensaje de error.

Realizar un Update en la tabla Candidato con los datos pasados como parámetros de entrada.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.7.28. BAJA CANDIDATO

#### Funcionalidad:

El procedimiento p\_bajacandidato permite dar de baja un candidato en un partido político en una votación existente siempre y cuando no estén asignados a una votación ya publicada.

Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro V_candidato	<b>Tipo de dato</b> Varchar2	<b>Entrada o Salida</b> E	Identificación/Mensaje Código del Candidato
RSP	Varchar2	S	Error + Descripción del error

Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobar los parámetros obligatorios, confirmando que poseen datos en caso contrario el programa lanzará un error.

Se comprueba que el candidato ha sido dato de alta previamente en caso contrario el programa lanzará un error.

Comprobar que el candidato no está asignado a ninguna votación

Realizar un Delete en la tabla Candidato.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

**3.7.29. ALTA VOTACION**Funcionalidad:

El procedimiento p\_altavotacion permite dar de alta las distintas votaciones que se pueden convocar en Europa.

Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro Titulo	<b>Tipo de dato</b> Varchar2	<b>Entrada o Salida</b> E	Identificación/Mensaje Título de la votación
Descripción	Varchar2	E	Descripción de la votación
Fecha publi	Date	E	Fecha publicación votación
Fecha ini	Date	E	Fecha inicio votación
Fecha fin	Date	E	Fecha fin votación
direccion	Varchar2	E	Dirección web
ambito	Number	E	Ambito votación
presidente	Varchar2	E	Presidente votación
RSP	Varchar2	S	Error + Descripción del error

Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobamos que la fecha actual es menor que la fecha de publicación de la votación.

Si el ámbito con el que damos de alta la votación es global, entonces actualizamos el número de electores de la votación con todos ciudadanos europeos inscritos en un censo electoral.

Realizar un Insert en la tabla Votación con los datos pasados como parámetros de entrada.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.7.30. MODIFICACIÓN VOTACIÓN

#### Funcionalidad:

El procedimiento p\_editavotación permite modificar una votación existente, siempre y cuando la votación no haya sido publicada. Para ello hay que pasar como parámetro el código de la votación y cualquiera de los siguientes campos: v\_titulo, v\_descripcion, f\_publi, f\_ini, f\_fin, web, v\_ambito, v\_presidente. Los campos se deben pasar en el orden preestablecido. Si no se quiere modificar un campo se puede enviar vacío si es una cadena o con un cero si es un número.

#### Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	<b>Tipo de dato</b>	<b>Entrada o Salida</b>	Identificación/Mensaje
V_votación	Integer	E	Código Votación
V_titulo	Varchar2	E	Título de la Votación
V_descripción	Varchar2	E	Descripción de la votación
F_publi	Date	E	Fecha publicación votación
F_ini	Date	E	Fecha inicio votación
F_fin	Date	E	Fecha fin votación
web	Varchar2	E	Dirección web
V_ambito	Number	E	Ambito votación
V_presidente	Varchar2	E	Presidente votación
RSP	Varchar2	S	Error + Descripción del error

#### Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobar los parámetros obligatorios, confirmando que poseen datos en caso contrario el programa lanzará un error.

Se comprueba que la votación ha sido dato de alta previamente en caso contrario el programa lanzará un error.

Se comprueba que la votación está en edición y que la fecha del sistema sea igual o superior a la fecha de publicación

En cualquiera de las comprobaciones anteriores sino se cumplen los requisitos el programa envía un mensaje de error.

Si el ámbito al que se pretende modificar es global, se actualiza el número de electores de la votación como el numero total de electores de la Unión Europea.

Realizar un Update en la tabla Elector con los datos pasados como parámetros de entrada.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.7.31. BAJA VOTACIÓN

#### Funcionalidad:

El procedimiento p\_bajavotación permite dar de baja una votación existente, siempre y cuando no haya sido publicada.

#### Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro V_votación	<b>Tipo de dato</b> Integer	<b>Entrada o Salida</b> E	Identificación/Mensaje Código Votación
RSP	Varchar2	S	Error + Descripción del error

#### Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobar los parámetros obligatorios, confirmando que poseen datos en caso contrario el programa lanzará un error.

Se comprueba que la votación no ha sido previamente publicada en caso contrario no se permitirá eliminar la votación lanzando el programa un mensaje de error.

Realizar un Delete en la tabla Votacion.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.7.32. ALTA DE UNA OPCION DE VOTACION

#### Funcionalidad:

El procedimiento p\_altaopcionesvotación permite asignar a una votación las distintas opciones que puede votar un elector.

Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro Votacion Opción	<b>Tipo de dato</b> Integer Number	<b>Entrada o Salida</b> E E	Identificación/Mensaje Código Votación Código Opción
RSP	Varchar2	S	Error + Descripción del error

Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobamos que la votación existe, y que el estado de la votación es de edición, si no se cumpliera cualquiera de los dos requisitos el programa lanza un mensaje de error.

Realizar un Insert en la tabla Opciones\_votacion con los datos pasados como parámetros de entrada.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

**3.7.33. BAJA DE UNA OPCIÓN DE VOTACIÓN**Funcionalidad:

El procedimiento p\_bajaopcionvotación permite dar de baja una de las opciones de elección de la votación.

Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro V_votación V_opcion	<b>Tipo de dato</b> Integer Number	<b>Entrada o Salida</b> E E	Identificación/Mensaje Código Votación Código de Opcion
RSP	Varchar2	S	Error + Descripción del error

Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobar que la votación existe, en caso contrario lanzar mensaje de error.

Comprobamos que la opción existe, en caso contrario lanzar mensaje de error.

Se comprueba que la votación no ha sido previamente publicada en caso contrario no se permitirá eliminar la opción de votación lanzando el programa un mensaje de error.

Realizar un Delete en la tabla Opciones\_Votación.

Si el programa no localiza el registro a borrar, se comunica igualmente mediante un mensaje de error.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.7.34. ALTA DE UN CENSO EN UNA VOTACIÓN

#### Funcionalidad:

El procedimiento p\_altacensoshabilitados permite habilitar un censo en una votación siempre y cuando el censo sea del ámbito correspondiente a la votación y la votación esté en estado de edición. Al mismo tiempo se actualiza el número de electores que pueden participar en dicha votación dependiendo de los censos que tenga habilitados la votación.

#### Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro Votación	<b>Tipo de dato</b> Integer	<b>Entrada o Salida</b> E	Identificación/Mensaje Código de la votación
Censo	Integer	E	Código del Censo
RSP	Varchar2	S	Error + Descripción del error

#### Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobar que la votación y el censo han sido dados de alta previamente, ya que de lo contrario se lanzaría un mensaje de error.

Comprobar que la votación está en estado de edición o pendiente ya que una vez iniciada la votación no se debe permitir modificar los censos asociados a la votación, enviando el programa un mensaje de error.

Comprobar que el censo que se va a habilitar corresponde realmente al ámbito de la votación, si no es de ese modo, se envía un mensaje de error.

Realizar un Insert en la tabla Censos\_habilitados con los datos pasados como parámetros de entrada.

Actualizamos el número de electores correspondientes a los censos que tiene habilitados dicha votación mediante el procedimiento p\_insertaelectores. (Al parámetro de entrada operación le pasamos un 0 para indicar que tiene que sumar los electores de dicho censo)

Se actualiza en estadísticas en número de votaciones asociadas a un censo por año.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.7.35. BAJA DE UN CENSO EN UNA VOTACIÓN

#### Funcionalidad:

El procedimiento p\_bajacensohabilitado permite dar de baja un censo en una votación, siempre y cuando no haya sido publicada.

#### Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	<b>Tipo de dato</b>	<b>Entrada o Salida</b>	Identificación/Mensaje
V_votación	Integer	E	Código Votación
V_censo	Integer	E	Código Censo
RSP	Varchar2	S	Error + Descripción del error

#### Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobar los parámetros obligatorios, confirmando que poseen datos en caso contrario el programa lanzará un error.

Comprobar que realmente el censo y la votación han sido datos de alta previamente, en caso contrario lanzar mensaje de error.

Se comprueba que la votación no es de ámbito global, ya que en ese caso tendrá habilitados por defecto todos los censos nacionales y no se le permite al usuario eliminar ninguno de los censos habilitados, lanzando el pertinente mensaje de error.

Realizar un Delete en la tabla Censos\_habilitados

En caso de no localizar el registro en la operación de borrado, se comunica mediante un mensaje de error.

Actualizar el número de electores en la votación mediante el procedimiento p\_insertaelectores. (Al parámetro de entrada operación le pasamos un 1 para indicar que tiene que restar los electores de dicho censo)

Se actualiza en estadísticas en número de votaciones asociadas a un censo por año.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.7.36. ALTA DE UN CANDIDATO A UN PARTIDO POLITICO EN UNA VOTACIÓN

#### Funcionalidad:

El procedimiento p\_altacandidotpartido permite dar de alta a un candidato en un partido político en una votación, siempre y cuando la votación esté en estado de edición.



Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	<b>Tipo de dato</b>	<b>Entrada o Salida</b>	Identificación/Mensaje
Votacion	Integer	E	Código de la votación
Partido	Integer	E	Código del partido
Candidato	Varchar2	E	Código del Candidato
Lista	Number	E	Orden de posición en las listas
RSP	Varchar2	S	Error + Descripción del error

Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobar que la votación y el candidato existen, en caso contrario se lanza un mensaje de error.

Se comprueba que la votación está en estado de edición, ya que de lo contrario no se pueden realizar cambios en la votación enviando el correspondiente mensaje de error.

Comprobar que el partido al que se presenta el candidato participa en dicha votación, en caso contrario lanzar error.

Realizar un Insert en la tabla Candidatos\_Votacion\_Partido con los datos pasados como parámetros de entrada.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### **3.7.37. MODIFICACIÓN DE UN CANDIDATO A UN PARTIDO POLITICO EN UNA VOTACIÓN**

Funcionalidad:

El procedimiento p\_editacandvotpart permite modificar la asignación de un candidato a un partido en una votación, siempre y cuando la votación no haya sido publicada. Para ello hay que pasar como parámetros el código de la votación y el código del candidato y cualquiera de los siguientes campos: partido, lista. Los campos se deben pasar en el orden preestablecido. Si no se quiere modificar un campo se puede enviar vacío si es una cadena o con un cero si es un número.

Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	<b>Tipo de dato</b>	<b>Entrada o Salida</b>	Identificación/Mensaje
Votacion	Integer	E	Código Votación
Partido	Integer	E	Código Partido
Candidato	Varchar2	E	Código Candidato
Lista	Number	E	Orden de posición en las listas
RSP	Varchar2	S	Error + Descripción del error

Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobar los parámetros obligatorios, confirmando que poseen datos en caso contrario el programa lanzará un error.

Comprobar que la votación, el candidato y el partido existen y que la votación está en edición

Comprobar que el partido participa en dicha votación.

En cualquiera de las comprobaciones anteriores sino se cumplen los requisitos el programa envía un mensaje de error.

Realizar un Update en la tabla Candidatos\_Votacion\_Partido con los datos pasados como parámetros de entrada.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.7.38. BAJA DE UN CANDIDATO A UN PARTIDO POLITICO EN UNA VOTACIÓN

Funcionalidad:

El procedimiento p\_bajacandvotpart permite dar de baja la asignación de un candidato a un partido político en una votación, siempre y cuando no haya sido publicada.

Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	<b>Tipo de dato</b>	<b>Entrada o Salida</b>	Identificación/Mensaje
V_votacion	Integer	E	Código Votación
V_partido	Integer	E	Código Partido
V_Candidato	Varchar2	E	Código Candidato
RSP	Varchar2	S	Error + Descripción del error

Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobar los parámetros obligatorios, confirmando que poseen datos en caso contrario el programa lanzará un error.

Comprobar que la votación, el candidato y el partido existen y que la votación está en edición

En cualquiera de las comprobaciones anteriores sino se cumplen los requisitos el programa envía un mensaje de error.

Realizar un Delete en la tabla Candidatos\_Votación\_Partido.

En caso de no localizar el registro en la operación de borrado, se comunica mediante un mensaje de error.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

**3.7.39. ALTA VOTAR**Funcionalidad:

El procedimiento p\_altavotar permite votar a un elector en una votación una opción, siempre y cuando la votación esté en estado abierta.

Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	<b>Tipo de dato</b>	<b>Entrada o Salida</b>	Identificación/Mensaje
Votacion	Integer	E	Código Votación
Opcion	Number	E	Código Opción
Votante	Varchar2	E	Código Votante
RSP	Varchar2	S	Error + Descripción del error

Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobar que la votación, el candidato y la opción de votación existen, en caso contrario se lanza un mensaje de error.

Se comprueba que la votación está en estado abierta para que los electores puedan ejercer su derecho al voto.

Comprobar que los censos en los que está inscrito el elector (regional + local), le habilitan para votar en dicha votación. El ámbito de la votación, habilita al censo regional o local del ciudadano para votar.

Comprobar que la opción de votación que ha elegido el elector corresponde con alguna de las que participan en la votación.

En cualquiera de las comprobaciones anteriores sino se cumplen los requisitos el programa envía un mensaje de error.

Realizar un Insert en la tabla Votar con los datos pasados como parámetros de entrada.

Actualizar el número de votos en la votación sumándole uno.

Actualizar el porcentaje de participación en la votación.

Actualizar el numero de votos de la opción votada en la votación.

Actualizar el porcentaje de participación de todas las opciones que se presentan en dicha votación.

Actualizar la opción más votada y su porcentaje de participación en dicha votación.

Actualizar la opción menos votada y su porcentaje de participación en dicha votación.

Actualizar las estadísticas mediante la ejecución del procedimiento p\_rellenaestadisticas.

\*ver procedimiento más adelante.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### **3.7.40. RELLENAR ESTADISTICAS.**

#### Funcionalidad:

El procedimiento p\_rellenaestadísticas se ejecuta cada vez que un ciudadano emite un voto, actualizando todas aquellas estadísticas cuyo valor se ve alterado al emitir el ciudadano el voto.

Parámetros: No tiene parámetros.

#### Descripción:

Se crea un cursor Anyosvota donde se almacenan los distintos años en los que se han celebrado elecciones.

Se procede a borrar los datos de todas las tablas estadísticas para poder actualizar sus valores.

Se abre el cursor Anyosvota para ir recorriendo los distintos años y por cada año insertamos en la tabla est\_vot\_mas\_parti\_anyo la votacion con mayor participación durante ese año y en la tabla est\_loc\_mas\_votos\_anyo la localidad con mayor número de votos emitidos por sus ciudadanos durante ese año.

Una vez que tenemos actualizadas estas dos estadísticas se procede a actualizar el resto:

Se inserta en la tabla est\_ciudadanos\_sinvotos\_anyo el porcentaje de ciudadanos que no han votado por año.

Se inserta en la tabla est\_historico\_votaciones la votación que históricamente ha tenido mayor participación.

Se inserta en la tabla est\_historico\_mas la votación que históricamente ha tenido mayor diferencia porcentual entre la opción más votada y la menos votada.

Se inserta en la tabla est\_historico\_menos la votación que históricamente ha tenido menor diferencia porcentual entre la opción más votada y la menos votada.

Se inserta en la tabla est\_votac\_pais\_persona el número máximo de votaciones en las que ha participado una misma persona y su país.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS .

### 3.7.41. INSERTAR ELECTORES EN UNA VOTACIÓN

#### Funcionalidad:

El procedimiento p\_insertaelectores permite actualizar el número de electores en una votación al añadir, modificar o eliminar la habilitación de un censo en una votación, siempre que dicha votación esté en estado de edición o publicación.

#### Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	<b>Tipo de dato</b>	<b>Entrada o Salida</b>	Identificación/Mensaje
Votacion	Integer	E	Código Votación
Ambito	Number	E	Código Ambito
Censo	Integer	E	Código Censo
Operación	Number	E	Operación a realizar: 0-Suma 1-Resta
RSP	Varchar2	S	Error + Descripción del error

#### Descripción:

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobamos el ámbito de la votación y dependiendo del mismo realizaremos las siguientes operaciones.

Si el Censo es Global no tenemos que realizar ninguna operación ya que se ha actualizado el número de electores al dar de alta la votación.

Si el censo es Nacional, se obtiene el código del país y mediante un cursor recorremos todas las regiones de ese país para contar el número de electores de todas sus regiones.

Si el censo es regional obtenemos el número de electores inscritos en la región.

Si el censo es local obtenemos el número de electores inscritos en la localidad.

Por último si la operación es 0, añadimos mediante un Update de la tabla votación, el número de electores obtenidos a los que poseía anteriormente la votación, en cambio si la operación es 1, restamos mediante un Update de la tabal votación, al numeró total de electores de la votación el número de electores obtenidos mediante las operaciones anteriores.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.7.42. INSERTAR UN ELECTOR EN UN CENSO

#### Funcionalidad:

El procedimiento p\_inelector permite añadir o eliminar un elector en todas aquellas votaciones en las que un ciudadano tiene permitido participar de acuerdo con los censos regional y local en los que está inscrito. Siempre que la votación esté en estado de edición o publicación.

#### Parámetros:

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro Region	<b>Tipo de dato</b> Integer	<b>Entrada o Salida</b> E	Identificación/Mensaje Código región
Localidad	Integer	E	Código localidad
operacion	Number	E	Operación a realizar: 0-Suma 1-Resta
RSP	Varchar2	S	Error + Descripción del error

#### Descripción:

Se borran los datos contenidos en la tabla auxiliar electores que se ha creado con anterioridad para almacenar los códigos de todas aquellas votaciones que hemos de actualizar el número de electores.

Se guardan los parámetros de entrada del procedimiento para insertarlos en el log.

Comprobar las votaciones en las que debo añadir o eliminar un elector.

#### Ámbito Global

En las votaciones de ámbito global en estado de edición o publicación siempre hay que actualizar el número de electores ya que pueden participar todos los ciudadanos de la unión Europea.

Abrir un cursor con todas las votaciones globales en edición o pendiente. Recorrer cada una de las votaciones insertando en la tabla auxiliar electores el código de la votación para posteriormente actualizarlas.

#### Ambito Nacional

Comprobar todas aquellas votaciones de ámbito nacional correspondientes al país del censo regional donde esté inscrito el ciudadano, es estado de edición o pendiente.

Obtener el código del país al que pertenece el censo regional.

Obtener el código del Censo nacional correspondiente a ese país.

Abrir un cursor con todas las votaciones en estado edición o pendientes del país al que pertenece el censo regional donde está inscrito el ciudadano.

Recorrer cada una de las votaciones insertando en la tabla auxiliar electores el código de la votación para posteriormente actualizarlas.

#### Ambito Regional

Comprobar todas aquellas votaciones de ámbito regional correspondientes al censo regional donde está inscrito el ciudadano, en estado de edición o pendiente.

Abrir un cursor con todas las votaciones en estado edición o pendientes de ámbito regional correspondiente al del ciudadano.

Recorrer cada una de las votaciones insertando en la tabla auxiliar electores el código de la votación para posteriormente actualizarlas.

#### Ambito Local

Comprobar todas aquellas votaciones de ámbito local correspondientes al censo local donde está inscrito el ciudadano, en estado de edición o pendiente.

Abrir un cursor con todas las votaciones en estado edición o pendientes de ámbito local correspondiente al del ciudadano.

Recorrer cada una de las votaciones insertando en la tabla auxiliar electores el código de la votación para posteriormente actualizarlas.

Comprobar el valor del parámetro de entrada operación:

- Si el valor es un cero el valor a añadir en el campo electores de la tabla votación será 1.
- Si el valor es un uno el valor a añadir en el campo electores de la tabla votación será -1

Abrir un cursor con todos los códigos de las votaciones almacenados en la tabla auxiliar electores.

Recorro cada uno de los registros actualizando mediante un update de la tabla votación el campo electores sumando (1,-1) según el caso, al número de electores almacenados en la tabla votación.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### **3.7.43. ACTUALIZAR ESTADO**

#### Funcionalidad:

Job que se ejecuta una vez al día y que nos permite actualizar el estado de una votación dependiendo de la fecha del sistema.

#### Parámetros:

No existen parámetros

#### Descripción:

Job j\_actualizareestado que ejecuta diariamente el procedimiento p\_editaestado.

### **3.7.44. EDITAR ESTADO DE UNA VOTACIÓN**

#### Funcionalidad:

El procedimiento p\_editaestado permite actualizar el estado de una votación a edición, pendiente, abierta y finalizada en relación a la fecha del sistema cuando se ejecute el job j\_actualizareestado.

#### Parámetros:

No posee parámetros.

#### Descripción:

Recorremos todas las votaciones mediante un cursor comparando la fecha del sistema con las distintas fechas de los estados de una votación.

Si la votación esta finalizada no tenemos que realizar ninguna comprobación.

Si la fecha del sistema es menor que la fecha de publicación de la votación actualizamos el estado de la votación a 1 y no seguimos comparando.

Si la fecha del sistema es menor que la fecha de inicio de la votación actualizamos el estado de la votación a 2 y no seguimos comparando.

Si la fecha del sistema es menor que la fecha fin de la votación actualizamos el estado de la votación a 3 y no seguimos comparando.

Finalmente sino se cumple ninguna de las condiciones anteriores actualizamos el estado de la votación a 4.

Cerramos el cursor con las votaciones

Borramos los datos de la tabla Est\_vot\_final\_año para posteriormente actualizar los datos de la tabla estadística, introduciendo el resultado de una sentencia select de las vistas paiscenso y votacenso agrupando los datos por país y año mediante una sentencia insert.

Borramos los datos de la tabla Est\_censo\_anyo\_porct para posteriormente actualizar los datos de la tabla estadística, introduciendo el resultado de una sentencia select de las vista votacenso agrupando los datos por año mediante una sentencia insert.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS.



### 3.8.LISTADOS

En el siguiente apartado se va a proceder a describir cada uno de los listados que ofrece la aplicación de elecciones, obtenidos mediante procedimientos de consulta.

#### 3.8.1. LISTADO DE VOTACIONES DE UN PAÍS

##### Funcionalidad:

El package `pk_consulta_votacionesPaís` permite consultar todas las votaciones que se celebran o se han celebrado en un país determinado. Este package esta formado por una función `f_votacionespais` y un procedimiento `p_listavotpais`.

La función `f_votacionespais` permite obtener un cursor con todas las votaciones de un país.

El procedimiento `p_listavotpais` permite listar todas las votaciones de un país utilizando la función `f_votacionespais`.

##### Parámetros:

`f_votacionespais`

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro Codigo_pais	<b>Tipo de dato</b> Varchar2	<b>Entrada o Salida</b> E	Identificación/Mensaje Código del país
Var_refcursor	Var_refcursor	S	Retorna cursor dinámico

##### Descripción:

Se abre el cursor mediante una consulta combinada de las tablas votación, estado y de los distintos censos que nos devuelve todas aquellas votaciones de un país pasado como parámetro.

Devolvemos el cursor con los datos de las votaciones de un país.

##### Parámetros:

`P_listavotpais`

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro Codigo_pais	<b>Tipo de dato</b> Varchar2	<b>Entrada o Salida</b> E	Identificación/Mensaje Código del país
RSP	Varchar2	S	Error + Descripción del error

##### Descripción:

Declaramos una variable de tipo `var_refcursor`.

Declaramos una variable de tipo registro con los campos que queremos que aparezcan en la consulta.

Guardamos el código del país como parámetro.

Obtenemos un cursor con las votaciones del país pasado como parámetro mediante la llamada a la función `f_votacionespais`, perteneciente al package `pk_consulta_votacionesPaís`.

Imprimimos el encabezado del listado para posteriormente ir recorriendo cada una de las votaciones almacenadas en el cursor e ir mostrando los datos de los campos por pantalla.

Cerramos el cursor.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.8.2. LISTADO DE LAS 10 VOTACIONES CON MAYOR DIFERENCIA PORCENTUAL.

#### Funcionalidad:

El package `pk_consulta_votmasdif` permite consultar las 10 votaciones con mayor diferencia porcentual entre la opción más votada y la menos votada. Este package está formado por una función `f_votacionesmas` y un procedimiento `p_listavotacionesmas`.

La función `f_votacionesmas` permite obtener un cursor con las 10 votaciones con mayor diferencia porcentual entre la opción más votada y la menos votada.

El procedimiento `p_listavotacionesmas` permite listar las 10 votaciones con mayor diferencia porcentual entre la opción más votada y la menos, utilizando la función `f_votacionesmas`.

#### Parámetros:

`f_votacionesmas`

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	Tipo de dato	Entrada o Salida	Identificación/Mensaje
Var_refcursor	Var_refcursor	S	Retorna cursor dinámico

#### Descripción:

Se abre el cursor mediante una consulta de la tabla votación donde se calcula la diferencia entre las opción más votada y menos votada de cada votación ordenadas por la diferencia.

Devolvemos el cursor con los datos de las 10 votaciones con mayor diferencia porcentual entre la opción más votada y la menos votada.

#### Parámetros:

`p_listavotacionesmas`

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	Tipo de dato	Entrada o Salida	Identificación/Mensaje
RSP	Varchar2	S	Error + Descripción del error

Descripción:

Declaramos una variable de tipo var\_refcursor;

Declaramos una variable de tipo registro con los campos que queremos que aparezcan en la consulta.

Actualizamos el parámetro al valor ‘ ‘.

Obtenemos un cursor con las 10 votaciones con mayor diferencia porcentual mediante la llamada a la función f\_votacionesmas, perteneciente al package pk\_consulta\_votmasdif.

Imprimimos el encabezado del listado para posteriormente ir recorriendo cada una de las votaciones almacenadas en el cursor e ir mostrando los datos de los campos por pantalla.

Cerramos el cursor.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### **3.8.3. LISTADO DE LAS 10 VOTACIONES CON MENOR DIFERENCIA PORCENTUAL.**

Funcionalidad:

El package pk\_consulta\_votmenosdif permite consultar las 10 votaciones con menor diferencia porcentual entre la opción más votada y la menos votada. Este package esta formado por una función f\_votacionesmenos y un procedimiento p\_listavotacionesmenos.

La función f\_votacionesmenos permite obtener un cursor con las 10 votaciones con menor diferencia porcentual entre la opción más votada y la menos votada.

El procedimiento p\_listavotacionesmenos permite listar las 10 votaciones con menor diferencia porcentual entre la opción más votada y la menos, utilizando la función f\_votacionesmenos.

Parámetros:

f\_votacionesmenos

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	<b>Tipo de dato</b>	<b>Entrada o Salida</b>	Identificación/Mensaje
Var_refcursor	Var_refcursor	S	Retorna cursor dinámico

Descripción:

Se abre el cursor mediante una consulta de la tabla votación donde se calcula la diferencia entre las opción más votada y menos votada de cada votación ordenadas por la diferencia.

Devolvemos el cursor con los datos de las 10 votaciones con menor diferencia porcentual entre la opción más votada y la menos votada.

Parámetros:

p\_listavotacionesmas

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro	<b>Tipo de dato</b>	<b>Entrada o Salida</b>	Identificación/Mensaje
RSP	Varchar2	S	Error + Descripción del error

Descripción:

Declaramos una variable de tipo var\_refcursor;

Declaramos una variable de tipo registro con los campos que queremos que aparezcan en la consulta.

Actualizamos el parámetro al valor ‘ ‘.

Obtenemos un cursor con las 10 votaciones con menor diferencia porcentual mediante la llamada a la función f\_votacionesmenos, perteneciente al package pk\_consulta\_votmenosdif.

Imprimimos el encabezado del listado para posteriormente ir recorriendo cada una de las votaciones almacenadas en el cursor e ir mostrando los datos de los campos por pantalla.

Cerramos el cursor.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### 3.8.4. LISTADO DE LOS CENSOS EN LOS QUE ESTÁ INSCRITO UN CIUDADANO.

Funcionalidad:

El package pk\_consulta\_censos permite consultar los censos en los que está inscrito un ciudadano. Este package esta formado por una función f\_censos y un procedimiento p\_listacensos.

La función f\_censos permite obtener un cursor con los censos en los que está inscrito un ciudadano.

El procedimiento p\_listacensos permite listar los censos de un ciudadano, utilizando la función f\_censos.

Parámetros:

f\_censos

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro codigo_ciudadano	<b>Tipo de dato</b> <b>Varchar2</b>	<b>Entrada o Salida</b> <b>E</b>	Identificación/Mensaje Codigo aleatorio del ciudadano
Var_refcursor	Var_refcursor	S	Retorna cursor dinámico

Descripción:

Se abre el cursor mediante una consulta combinada de las tablas elector y las tablas de los distintos censos de una votación (global, nacional, regional y local).

Devolvemos el cursor con los datos de los censos en los que está inscrito el ciudadano que se identifica mediante el código aleatorio pasado como parámetro a la función.

Parámetros:

p\_listacensos

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro codigo_ciudadano	Tipo de dato Varchar2	Entrada o Salida E	Identificación/Mensaje Codigo aleatorio del ciudadano
RSP	Varchar2	S	Error + Descripción del error

Descripción:

Declaramos una variable de tipo var\_refcursor;

Declaramos una variable de tipo registro con los campos que queremos que aparezcan en la consulta.

Actualizamos la variable parámetro con el código del ciudadano.

Obtenemos un cursor con los censos en los que esta inscrito el ciudadano pasado como parámetro mediante la llamada a la función f\_censos, perteneciente al package pk\_consulta\_censos .

Imprimimos el encabezado del listado para posteriormente ir recorriendo cada uno de los censos almacenadas en el cursor e ir mostrando los datos de los campos por pantalla.

Cerramos el cursor.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

### **3.8.5. LISTADO DE LAS VOTACIONES EN LAS QUE HA PARTICIPADO UN CIUDADANO.**

Funcionalidad:

El package pk\_consulta\_participacion permite consultar las votaciones en las que ha participado un ciudadano. Este package esta formado por las funciones f\_votaciones, f\_censos y un procedimiento p\_listavotaciones.

La función f\_censos permite obtener los censos en los que está inscrito un ciudadano.

La función f\_votaciones permite obtener un cursor con las votaciones en las que participa y no participa un ciudadano mediante la utilización de la función f\_censos.

El procedimiento p\_listavotaciones permite listar las votaciones en las que participa y no participa un ciudadano, utilizando la función f\_votaciones.

Parámetros:

f\_censos

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro codigo_ciudadano	<b>Tipo de dato</b> Varchar2	<b>Entrada o Salida</b> E	Identificación/Mensaje Codigo aleatorio del ciudadano
Var_refcursor	Var_refcursor	S	Retorna cursor dinámico

Descripción:

Se abre el cursor mediante una consulta combinada de las tablas elector y las tablas de los distintos censos de una votación (global, nacional, regional y local).

Devolvemos el cursor con los datos de los censos en los que está inscrito el ciudadano que se identifica mediante el código aleatorio pasado como parámetro a la función.

Parámetros:

f\_votaciones

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro codigo_ciudadano	<b>Tipo de dato</b> Varchar2	<b>Entrada o Salida</b> E	Identificación/Mensaje Codigo aleatorio del ciudadano
Var_refcursor	Var_refcursor	S	Retorna cursor dinámico

Descripción:

Obtenemos un cursor con los censos en los que esta inscrito el ciudadano pasado como parámetro mediante la llamada a la función f\_censos, perteneciente al package pk\_consulta\_participacion .

Recuperamos del cursor cada uno de los censos en los que está inscrito el ciudadano.

Obtenemos un cursor con las votaciones en las que ha participado y en las que ha podido participar un ciudadano pero no lo ha hecho, correspondientes a los censos en los que está inscrito y que hemos obtenido anteriormente mediante una consulta select de las distintas tablas de censos, votacion y votar.

La consulta select consiste en restar a todas aquellas votaciones que tienen habilitadas los censos correspondientes a los que está inscrito el ciudadano, aquellas votaciones en las que ha participado o votado; De esta manera obtenemos las votaciones en las que no ha participado. PARTICIPA

Al resultado anterior le añadimos las votaciones en las que ha participado el ciudadano. NO PARTICIPA.

Devolvemos el cursor con los datos de las votaciones en las que ha participado y de aquellas en las que no ha participado un ciudadano siempre y cuando esté habilitado para ello.

Parámetros:

p\_listavotacioens

NOMBRE	TIPO	E/S	DESCRIPCION
Parámetro codigo_ciudadano	<b>Tipo de dato</b> Varchar2	<b>Entrada o Salida</b> E	Identificación/Mensaje Codigo aleatorio del ciudadano
RSP	Varchar2	S	Error + Descripción del error

Descripción:

Declaramos una variable de tipo var\_refcursor;

Declaramos una variable de tipo registro con los campos que queremos que aparezcan en la consulta.

Actualizamos la variable parámetro con el código del ciudadano.

Obtenemos un cursor con las votaciones en las que participa el ciudadano pasado como parámetro mediante la llamada a la función f\_votaciones, perteneciente al package pk\_consulta\_participación .

Imprimimos el encabezado del listado para posteriormente ir recorriendo cada una de las votaciones almacenadas en el cursor e ir mostrando los datos de los campos por pantalla.

Cerramos el cursor.

En caso de éxito se devuelve OK y se inserta en la tabla de logs un registro.

En caso de error en cualquier paso se cancelan todos los cambios hechos en la base de datos, se inserta un registro en la tabla de LOGS y se devuelve una descripción del error en el parámetro de salida.

## 4. Bibliografía

<http://mas.laopiniondezamora.es/elecciones2011/preguntas-frecuentes/mesaselectorales.html>

[http://es.wikipedia.org/wiki/Elecciones\\_al\\_Parlamento\\_Europeo\\_de\\_2009](http://es.wikipedia.org/wiki/Elecciones_al_Parlamento_Europeo_de_2009)

<http://www.elecciones.mir.es/europeas2009/candidaturas.html>

Recursos/Materiales y Fuentes/Materiales de la Asignatura/TFC de la UOC

<http://www.mapanet.es/Resources/Countries.asp>

<http://www.quillslanguage.es/recursos/iso3166-1.php>

[vototelematico.diatel.upm.es/articulos/ARTICULO%20AECPA.pdf](http://vototelematico.diatel.upm.es/articulos/ARTICULO%20AECPA.pdf)

Desarrollo de Bases de Datos: Casos prácticos desde el análisis a la implementación.

Editorial Ra-Ma.

[http://download.oracle.com/docs/cd/B19306\\_01/appdev.102/b14258/d\\_sched.htm](http://download.oracle.com/docs/cd/B19306_01/appdev.102/b14258/d_sched.htm)

[http://download.oracle.com/docs/cd/B10501\\_01/appdev.920/a96624/07\\_errs.htm](http://download.oracle.com/docs/cd/B10501_01/appdev.920/a96624/07_errs.htm)

[http://download.oracle.com/docs/cd/B14117\\_01/server.101/b10744/intro.htm](http://download.oracle.com/docs/cd/B14117_01/server.101/b10744/intro.htm)