Diseño e implementación de la B.D. de un: Sistema de votaciones ciudadano a nivel Europeo, a través de Internet.

TFC. Memoria

Alicia Fernández Martínez
Ingeniería Técnica de Informática de Gestión
afernandezmar@uoc.edu

Manel Rella Ruiz
Consultor

09/10/2011

Fecha de creación: 15/01/2012
Fecha de modificación: 15/01/2012
Fichero: afernandezmarTFC0112memoria

Índice de Contenidos

1.…INTRODUCCIÓN 4
1.1 Justificación del TFC y Contexto en el que se desarrolla 4
1.2 Objetivos del Proyecto 5
1.3 Enfoque y Metodología a seguir 6
1.4 Planificación del Proyecto 6
1.4.1 Definición jornada laboral y Calendario de Fechas 7
1.4.2 Identificación de Actividades/Tareas 7
1.4.3 Planificación Temporal 8
1.4.4 Diagrama de Gantt 11
1.5 Producto Obtenido 12
1.5.1 Entregas Parciales 12
1.5.1.1 Planificación TFC (PAC 1) 12
1.5.1.2 Análisis y Diseño de la BD (PAC 2) 12
1.5.1.3 Implementación de la BD (PAC 3) 12
1.5.1.4 Producto final + Memoria + Presentación 12
1.6 Breve descripción del resto de capítulos 13

2.…REQUISITOS DE LA B.D. VOTACIONES ON-LINE 14
2.1 Iniciales 14
2.2 Funcionales 14
2.3 Nuevas Propuestas Funcionales 17
2.4 No Funcionales 17

3.…DISEÑO DE LA B.D. VOTACIONES ON-LINE 19
3.1 Diseño Conceptual 19
3.1.1 Diagrama UML 19
3.1.2 Identificación de Entidades 20
3.1.3 Atributos de las Entidades 20
3.2 Diseño Lógico 22
3.3 Diseño Físico 23
3.3.1 Crear la Base de Datos 23
3.3.2 Tablespaces 23
3.3.3 Usuarios 23
3.3.4 Tablas 23
3.3.5 Packages 30
3.3.6 Funciones 34
3.3.7 Triggers 37

4.…IMPLEMENTACIÓN 38
4.1 Cr_tablas 39
4.2 Cr_Secuencias 39
4.3 Cr_tipos_objeto 39
4.4 Funciones 39
4.4.1 FLISTA_VOTAPPAIS 39
4.4.2 FTOPTENMASVOTA 40
4.4.3 FTOPTENMENVOTA 40
4.4.4 FCENSOPERSONA 40
4.4.5 FVOTAPERSONA 41
4.5 Procedimientos de Alta, Baja y Modificación 41
4.5.1 PK_CENSOS 41
4.5.1.1 Gestión de Países (INS_PAIS) 41

4.5.1.2 Gestión de Países (DEL_PAIS) ... 42
4.5.1.3 Gestión de Países (UPD_PAIS) ... 42
4.5.1.4 Gestión de Regiones (INS_REGION) ... 43
4.5.1.5 Gestión de Regiones (DEL_REGION) ... 43
4.5.1.6 Gestión de Regiones (UPD_REGION) ... 43
4.5.1.7 Gestión de Localidades (INS_LOCALI) ... 44
4.5.1.8 Gestión de Localidades (DEL_LOCALI) ... 44
4.5.1.9 Gestión de Localidades (UPD_LOCALI) ... 45
4.5.2 PK_PERSONAS ... 45
4.5.2.1 Gestión de Personas (INS_PERSONAS) ... 45
4.5.2.2 Gestión de Personas (DEL_PERSONAS) ... 46
4.5.2.3 Gestión de Personas (UPD_PERSONAS) ... 46
4.5.2.4 Mantenimiento Censo - Personas (INS_PERSOCENSO) ... 47
4.5.2.5 Mantenimiento Censo - Personas (DEL_PERSOCENSO) ... 48
4.5.2.6 Mantenimiento Votaciones - Personas (INS_VOTAPERSONA) ... 48
4.5.3 PK_VOTACION ... 49
4.5.3.1 Gestión de Votación (INS_VOTACION) ... 49
4.5.3.2 Gestión de Votación (DEL_VOTACION) ... 50
4.5.3.3 Gestión de Votación (UPD_VOTACION) ... 50
4.5.3.4 Gestión de Votación (UPD_VOTACION) ... 51
4.5.3.5 Gestión de Votación (UPD_ESTADO_VOTA) ... 51
4.5.3.6 Gestión de Opciones (INS_OPCION) ... 52
4.5.3.7 Gestión de Opciones (DEL_OPCION) ... 53
4.5.3.8 Gestión de Opciones (UPD_OPCION) ... 53
4.5.3.9 Gestión de Censo - Votación (INS_VOTACENSO) ... 54
4.5.3.10 Gestión de Censo - Votación (DEL_VOTACENSO) ... 55
4.6 Módulo Estadístico ... 55
4.6.1 DISPARADORES ... 56
4.6.1.1 TR_INS_VOTAPERSONA ... 56
4.6.1.2 TR_UPD_ESTADO_VOTACION ... 56
4.6.2 Consultas del módulo ... 57

5. ... PLAN DE CONTINGENCIAS ... 58

6. ... PLAN DE PRUEBAS ... 58
6.1 Carga Inicial de datos ... 58
6.2 Pruebas de Procedimientos ... 59
6.3 Pruebas de Consultas y Consultas Estadísticas ... 59
6.3.1 Pruebas de Consultas ... 59
6.3.2 Pruebas Módulo Estadístico ... 60
6.4 Resultado de las pruebas ... 60

7. ... VALORACIÓN ECONÓMICA Y RECURSOS ... 60
7.1 Valoración Económica ... 60
7.2 Recursos Necesarios ... 62

8. ... BIBLIOGRAFÍA ... 62

9. ... COMENTARIOS ... 62
9.1 Revisión de la PAC2 con respecto a la PAC1 ... 62
9.2 Revisión de la PAC3 con respecto a la PAC2 ... 62

10. . CONCLUSIONES ... 63

1. Introducción

El presente Trabajo Final de Carrera, en adelante TFC, tiene como objetivo el diseño e implementación de una base de datos que cubra las necesidades para la futura aplicación de Gestión de Votaciones Ciudadanas a través de Internet, para toda la Comunidad Europea. Esta deberá de cumplir todos los requisitos especificados en el enunciado proporcionado por los profesores de la asignatura.

En líneas generales la base de datos, a de permitir almacenar información de los datos básicos de las votaciones incluyendo un histórico, los censos electorales, los censos que participan en cada votación y datos estadísticos de los resultados de las votaciones.
Con esta información el sistema deberá permitir, a los gestores de la aplicación, definir las propuestas que serán votadas y a que censo irán dirigidas, mantener los censos a los que pertenece cada ciudadano, permitir las votaciones, a las propuestas, de los ciudadanos a través de Internet. Al finalizar y cerrar las votaciones, el sistema deberá realizar una serie de cálculos con los resultados que se obtenga de las votaciones, estos datos serán almacenados para ser consultados y explotados estadísticamente.

Se nos pide realizar:

- Diseño de la base de datos.
- Generar los scripts de creación de B.D., usuarios, tablas, índices, etc..
- Generar los datos mínimos para arrancar la aplicación, definir datos para las tablas maestras.
- Implementar los procedimientos que sean necesarios para su gestión. Los procedimientos almacenados, tienen que tener control de excepciones y dejar registros en una tabla de Logs.
- Generar un juego de pruebas integral.

1.1 Justificación del TFC y Contexto en el que se desarrolla

El TFC es una asignatura que está pensada para realizar un trabajo de síntesis de los conocimientos adquiridos en diversas asignaturas de Ingeniería Técnica de Informática de Gestión y requiere ponerlos en práctica en un proyecto concreto. Normalmente, y como ocurre en este caso, el TFC es un trabajo con aplicación práctica y vinculado al ejercicio profesional de la informática.

En el área de Bases de Datos relacionales el objetivo es consolidar los conocimientos adquiridos en esta área a través de asignaturas como Bases de Datos I, Bases de Datos II y Sistemas de Gestión de Bases de Datos. Adicionalmente también se utilizan conocimientos adquiridos en asignaturas como Estructura de la Información y Gestión de Organización y proyectos informáticos.

En este caso el TFC se trata de un ejemplo práctico de cómo realizar un diseño de una base de datos relacional, a partir de unos requisitos iniciales proporcionados por el cliente. También de diseñar e implementar los

procedimientos almacenados que usará la aplicación para interactuar con ella y de construir un juego de pruebas que permita testear el correcto funcionamiento del sistema con respecto a los requerimientos. El sistema de Gestión de bases de datos será Oracle, lo cual es un valor añadido al trabajo, ya que actualmente goza de muy buena posición a nivel de mercado.

Este proyecto tiene como objetivo dar la oportunidad al estudiante de consolidar y ampliar los conocimientos adquiridos en el área de Base de Datos y ponerlos en práctica mediante un sistema de gestión de base de datos relacional. Además podrá adquirir ciertas habilidades de planificación, gestión, toma de decisiones y resolución de problemas.
En este proyecto, en según en que fase y la tarea que estemos desarrollando en aquel momento, nos tendremos que adaptar a los distintos tipos de perfil que entran en juego en este proyecto, que son:

- Jefe de proyecto
- Analista
- Programador
- Administrador de Base de datos

Los cuales serán de mucha utilidad en nuestra vida profesional, ya que podemos realizar en un mismo proyecto, tareas de perfiles diferentes.

1.2 Objetivos del Proyecto

El objetivo general del proyecto es el de proporcionar a la Comunidad Europea, el diseño e implementación de una base de datos que cubra las necesidades de almacenaje de datos para la futura aplicación, de votaciones ciudadanas a través de Internet, que quieren implementar.

El producto a entregar deberá de cumplir con todos los requisitos especificados en la documentación entregada por el cliente. Donde se entrará mas en detalle en el capitulo de “Requisitos de la B.D.”.

En el ciclo de vida del Proyecto, existen varios hitos que tenemos que cumplir, estos son los siguientes:

- PAC1: Plan de trabajo. Tiene como objetivo llegar a establecer una planificación lo más detallada y realista posible. Así como mostrar el alcance del proyecto.
- PAC2: Análisis y Diseño. Tiene como objetivo realizar el análisis y diseño de la B.D. a través de los requerimientos.
- PAC3: Implementación y Pruebas. El objetivo es realizar todo el desarrollo y la implementación de las tablas, procedimientos, disparadores, etc. que sean necesarios para cumplir con los requisitos del proyecto.
- ENTREGA FINAL: La entrega final se compone de varios documentos:
    - Producto: Scripts que compone todo el proyecto necesarios para su funcionamiento.
    - Memoria: Documento de síntesis final.
    - Presentación: Resumen del trabajo realizado y de los resultados obtenidos.

Un punto importante será la valoración económica del proyecto, al entrar en concurso con otros proveedores, tendrá que tener unos precios competitivos con la situación del mercado actual.

Para acabar el proyecto correctamente y en la fecha señalada, es necesaria la colaboración y disponibilidad del cliente para aclarar las dudas que vayan surgiendo a lo largo del proyecto, y así poder realizar un producto que cubra todas las necesidades que se han requerido.

1.3 Enfoque y Metodología a seguir

Para conseguir una estimación ajustada del volumen de trabajo y una planificación, lo más realista posible, he estimado conveniente usar el ciclo de vida clásico, es decir, el ciclo de vida en cascada el cual es útil en proyectos cuya duración es acotada y se definen claramente los requisitos del proyecto desde un inicio.

De este modo, se puede planificar cual será la duración de las diferentes fases, se puede planificar la asignación de recursos para ajustarse a esta planificación, y se puede calcular el coste total del proyecto.

Las fases que se prevén en este proyecto son las siguientes:
- Planificación sobre los requerimientos iniciales
- Recogida de Requerimientos, Análisis y Diseño.
- Desarrollo e Implementación.
- Pruebas.
- Entrega Total

Para el desarrollo de las fases previstas en el proyecto, utilizaré los métodos, técnicas y herramientas aprendidas en las diferentes asignaturas cursadas en ETIG.
- La Planificación la he realizado, con la metodología aprendida en la asignatura de Gestión de Organización de proyectos informáticos.
- Para el diseño de base de datos relacionales e implementación, aplicaré las técnicas aprendidas en las asignaturas de Base de Datos I, II y Gestión de Base de Datos.

Para realizar la estimación de esfuerzo, realizaré una descomposición en actividades, subsistemas, puntos de función y me basaré en mi experiencia profesional para estimarlo.

1.4 Planificación del Proyecto

Para realizar la planificación del proyecto he procedido de la siguiente manera:
1. Identificar los Hitos del proyecto.
2. Disponibilidad de dedicación al proyecto.

3. Elegir la estructura del proyecto más adecuada para alcanzar los objetivos.
4. Descomponer el proyecto en Actividades/Tareas.
5. Asignar las actividades al perfil que las va a desarrollar.
6. Estimar el esfuerzo de cada Actividad/Tarea.
7. Realizar la planificación Temporal
8. Realizar el Diagrama de Grantt

1.4.1 Definición jornada laboral y Calendario de Fechas

La planificación del Proyecto la he organizado en base a las fecha de entrega de la evaluación continuada propuestas por el cliente.

Fecha	Tarea	Descripción
21/10/2011	Inicio del proyecto	Fecha de inicio del proyecto.
09/10/2011	PAC1	Fecha de entrega de la PAC1 (Plan de Trabajo).
13/11/2011	PAC2	Fecha de entrega de la PAC2. Análisis y Diseño B.D.
14/12/2011	PAC3	Fecha de entrega de la PAC3. Implementación B.D.
15/01/2012	Entrega Final	Fecha de entrega de la Memoria, la presentación y el Producto.

La jornada laboral será de 4 horas laborables durante cinco días a la semana. A priori, no se considera la necesidad de realizar horas extras, ni de trabajar los fines de semana, ni los días festivos.

Se disponen de 82 días laborables, de los que se han de eliminar los días festivos, en Barcelona, que son:

- 12 de Octubre: Fiesta Nacional
- 1 de Noviembre: Todos los Santos
- 6 y 8 de Diciembre: Fiesta de la Constitución y de la Inmaculada.
- 26 de Diciembre: San Esteban.
- 6 de enero: Fiesta de Reyes.

De los 82 hay que restar 6 por lo que tenemos unos 77 días laborables para el desarrollo del proyecto.

1.4.2 Identificación de Actividades/Tareas

De acuerdo con los estándares de proyectos de construcción y la lectura de la documentación de requerimientos, las actividades que compondrán el proyecto serán las siguientes:

Código de la Actividad	Nombre de la actividad de nivel 1	Nombre de la actividad de nivel 2
01	Inicio del proyecto	
02	Recogida y Lectura de la documentación	
03	Instalación Gantt-Project	
04	Planificación del Proyecto	

Código de la Actividad	Nombre de la actividad de nivel 1	Nombre de la actividad de nivel 2
04.01		Descripción Justificación TFC
04.02		Definición Objetivos
04.03		Identificación de Tareas
04.04		Planificación del Proyecto
04.05		Valoración Económica y de Recursos necesarios
04.06		Elaboración Documentación a Entregar
05	Análisis y Diseño	
05.01		Análisis Requisitos Funcionales
05.02		Análisis Requisitos no Funcionales
05.03		Diseño Conceptual B.D.
05.04		Diseño Lógico B.D.
05.05		Diseño Físico B.D.
05.06		Instalación Oracle.
05.07		Elaboración Documentación a Entregar
06	Desarrollo, Implementación y Pruebas	
06.01		Scripts de creación B.D. y Tablas
06.01		Gestión de Censos
06.02		Gestión de Votaciones
06.03		Gestión Personas
06.04		Gestión Estadística
06.05		Gestión Consultas
06.06		Juego de Pruebas Unitarias
06.07		Elaboración Documentación a Entregar
07	Entrega Producto	
07.01		Pruebas Integrales
07.02		Memoria
07.03		Presentación
07.04		Repaso de la Documentación a Entregar

En la entrega del producto se ha puesto una tarea de pruebas integrales, porque en la fase anterior veo muy justo poder realizarla.

1.4.3 Planificación Temporal

Es necesario establecer las precedencias entre las actividades que hay que desarrollar dentro del proyecto, para tener una idea aproximada de la planificación, para ello utilizaré una tabla de recursos y precedencias.
Las tareas identificadas para las cuales se realiza la planificación temporal son las siguientes:

Código Actividad	Nombre Actividad	Estimación (Jornadas)	Recurso	Precedencia
01.	Inicio del proyecto	01	Jefe Proyecto	00
02.	Recogida y Lectura de la documentación	02	Jefe Proyecto Analista	01
03.	Instalación Gantt-Project	02		02
04.	Planificación del Proyecto (PAC1)			
04.01.	Descripción Justificación TFC	01	Jefe Proyecto	03
04.02.	Definición de Objetivos	01	Jefe Proyecto	04.01
04.03.	Identificación de Tareas	01	Jefe Proyecto	04.02

Código Actividad	Nombre Actividad	Estimación (Jornadas)	Recurso	Precedencia
04.04.	Planificar Proyecto	01	Jefe Proyecto	04.03
04.05.	Valoración Económica y Recursos necesarios	02	Jefe Proyecto	04.04
04.06.	Elaboración Documentación a Entregar	02	Jefe Proyecto	04.05
05.	Análisis y Diseño			
05.01.	Análisis			
05.01.01.	Requisitos Funcionales	02	Analista	04.06
05.01.02.	Requisitos no Funcionales	02	Analista	05.01.01
05.02.	Diseño B.D.			
05.02.01.	Conceptual	05	Analista	05.01.02
05.02.02.	Lógico	05	Analista	05.02.01
05.02.03.	Físico	05	Analista	05.02.02
05.03.	Instalación de Oracle	01	Adminis. B.D.	05.02.03
05.04.	Elaboración Documentación a Entregar	03	Analista	05.03
06.	Implementación y Pruebas			
06.01.	Scripts de creación B.D., Tablas, Usuario y datos mínimos.	02	Programador	05.04
06.02.	Desarrollo de Scripts			
06.02.01.	Gestión de Censos	02	Programador	06.01
06.02.02.	Gestión de Votaciones	02	Programador	06.02.01
06.02.03.	Gestión de Personas	02	Programador	06.02.02
06.02.04.	Gestión Estadística	02	Programador	06.02.03
06.02.05.	Gestión de Consultas	02	Programador	06.02.04
06.03.	Pruebas Unitarias			
06.03.01.	Juego de pruebas	02	Analista	06.02.05
06.03.02.	Script de pruebas	02	Analista	06.03.01
06.03.03.	Depuración código	02	Programador	06.03.02
06.04.	Elaboración Documentación a Entregar	03	Analista	06.03.03
07.	Producto + Memoria + Presentación			
07.01.	Pruebas Integrales y Depuración de errores	05	Analista Programador	06.04
07.02.	Memoria	08	Jefe Proyecto, Analista	07.01
07.03.	Presentación	05	Jefe Proyecto	07.02
07.04.	Repaso de la documentación a entregar	02	Jefe Proyecto, Analista, Programador	07.03
TOTAL		**77**		

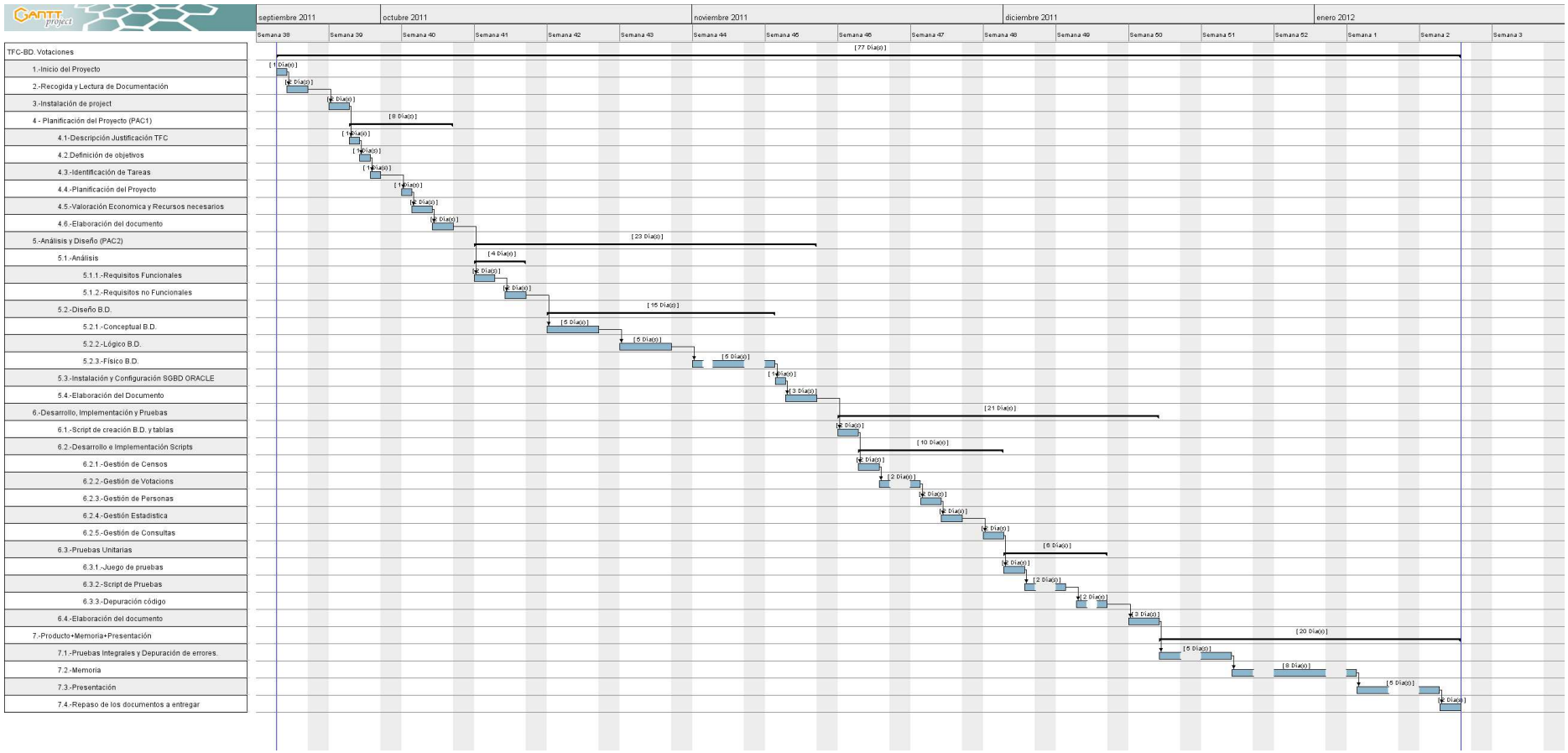
La planificación del proyecto será la siguiente:

Nombre Actividad/Tarea	Comienzo	Fin	Duración
TFC BD	**21/09/2011**	**13/01/2012**	**77**
1.-Inicio del Proyecto	**21/09/2011**	**22/09/2011**	**1**
2.-Recogida y Lectura documentación	**22/09/2011**	**24/09/2011**	**2**
3.-Instalación de software ofimático no instalado.	**26/09/2011**	**28/09/2011**	**2**
4.- Planificación del Proyecto (PAC1)	**28/09/2011**	**08/10/2011**	**8**
4.1.-Descripción Justificación TFC	28/09/2011	29/09/2011	1
4.2.-Definición de objetivos	29/09/2011	30/09/2011	1
4.3.-Identificar Tareas	30/09/2011	01/10/2011	1
4.4.-Planificar Proyecto	03/10/2011	04/10/2011	1
4.5.-Valoración Económica y Recursos necesarios	04/10/2011	06/10/2011	2
4.6.- Elaboración Documentación a Entregar	06/10/2011	08/10/2011	3
5.- Análisis y Diseño (PAC2)	**10/10/2011**	**12/11/2011**	**23**
5.1.-Análisis	10/10/2011	15/10/2011	04
5.1.1.-Requisitos Funcionales	10/10/2011	12/10/2011	2

Nombre Actividad/Tarea	Comienzo	Fin	Duración
5.1.2.- Requisitos no Funcionales	13/10/2011	15/10/2011	2
5.2.- Diseño BD	17/10/2011	08/11/2011	15
5.2.1.-Diseño Conceptual BD	17/10/2011	22/10/2011	5
5.2.2.-Diseño Lógico BD	24/10/2011	29/10/2011	5
5.2.3.-Diseño Físico BD	31/10/2011	08/11/2011	5
5.3.- Instalación y Configuración SGBD ORACLE	08/11/2011	09/11/2011	1
5.4.- Elaboración Documentación a Entregar	09/11/2011	12/11/2011	3
6.- Desarrollo, implementación y Pruebas (PAC3)	**14/11/2011**	**15/12/2011**	**21**
6.1.- Script de creación B.D., Tablas, Usuario y datos	14/11/2011	16/11/2011	2
6.2.- Desarrollo de Scripts	16/11/2011	30/11/2011	10
6.2.1.-Gestión de Censos	16/11/2011	18/11/2011	2
6.2.2.-Gestión de Votaciones	18/11/2011	22/11/2011	2
6.2.3.-Gestión de Personas	22/11/2011	24/11/2011	2
6.2.4.-Gestión Estadística	24/11/2011	26/11/2011	2
6.2.5.-Gestión de Consulta	28/11/2011	30/11/2011	2
6.3.- Pruebas Unitarias	30/11/2011	10/12/2011	6
6.3.1.-Juego de pruebas	30/11/2011	02/12/2011	2
6.3.2.-Script de pruebas	02/12/2011	06/12/2011	2
6.3.3.-Depuración código	07/12/2011	10/12/2011	2
6.4.- Elaboración Documentación a Entregar	12/12/2011	15/12/2011	3
7.- Producto + Memoria + Presentación	**15/12/2011**	**13/01/2012**	**20**
7.1.-Pruebas y depuración de errores	15/12/2011	22/12/2011	5
7.2.-Memoria	22/12/2011	03/01/2012	8
7.3.-Presentación	03/01/2012	11/01/2012	5
7.4.-Repaso de la documentación a entregar	11/01/2012	13/01/2012	2

1.4.4 Diagrama de Gantt

Con la ayuda del Gantt-Project y considerando la fecha de inicio del proyecto el día 21/09/2011, la planificación propuesta es la siguiente:



1.5 Producto Obtenido

Tras la realización del proyecto, obtendremos los siguientes documentos:

- *Plan de Trabajo*: planificación y estimación de las tareas necesarias para llevar a cabo los objetivos previstos.
- *Producto*: desarrollos correspondientes a la Base de Datos.
- *Memoria*: síntesis del trabajo realizado y muestra que se han alcanzado los objetivos propuestos.
- *Presentación*: resumen claro y conciso, en diapositivas, del trabajo realizado y de los resultados obtenidos.

1.5.1 Entregas Parciales

Las entregas parciales que se realizaran en el curso de la evaluación continuada y que se contemplan en la planificación realizada serán las siguientes:

1.5.1.1 Planificación TFC (PAC 1)

En esta primera fase, siguiendo las metodologías de planificación aprendidas, determinamos las tareas necesarias para alcanzar los objetivos del proyecto y estimación del esfuerzo, los recursos necesarios para llevarlas a cabo y la valoración económica.

1.5.1.2 Análisis y Diseño de la BD (PAC 2)

Esta tarea comprende las siguientes actividades:

- Una revisión de los requerimientos proporcionados en el enunciado para la BD, para evaluar si son suficientes para comenzar el diseño e identificar los puntos abiertos que requieran una clarificación.
- Elaboración del diseño conceptual de la BD mediante diagramas E/R y su consecuente traslación a diseño lógico.
- Instalación y configuración del SGBD Oracle en los equipos asignados al proyecto, de modo que esté disponible al inicio del desarrollo.
- Elaboración del diseño físico de la BD.

1.5.1.3 Implementación de la BD (PAC 3)

- Construcción y prueba de los objetos de la BD incluyendo el esquema de tablas, juego de datos, secuencias y disparadores, los procedimientos almacenados y las funciones necesarias.
- Revisión y prueba del código desarrollado.

1.5.1.4 Producto final + Memoria + Presentación

- *producto final*: Scripts necesarios para el funcionamiento del proyecto:
    - Script de creación de las tablas en la Base de Datos.
    - Script de juego de datos para el rellenado de tablas maestras.

  - Script de creación de los procedimientos de Base de Datos.
  - Script de pruebas de la Base de Datos para corroborar su comportamiento según los requerimientos.
- *memoria*: este mismo documento. La memoria debe sintetizar el trabajo realizado y mostrar claramente que se han alcanzado los objetivos propuestos. Formalmente, la memoria debe contener la información relevante que permita entender el problema planteado por el TFC, la metodología utilizada para su resolución y mostrar la resolución del problema planteado.
- *presentación*: es un documento de síntesis que debe sintetizar de forma clara y concisa el trabajo realizado a lo largo del semestre y los resultados obtenidos.

1.6 Breve descripción del resto de capítulos

Los siguientes capítulos del TFC son:

- *Análisis del sistema:* Partiendo de la descripción proporcionada por el cliente sobre lo que desea, se deducen los requisitos, los cuales serán descritos de forma clara y concisa.
- *Diseño del sistema:* A partir de los requisitos, se diseña la Base de Datos necesaria para albergar la información y las funcionalidades que los satisfagan. El diseño de la Base de Datos se realizará en diferentes etapas (diseño conceptual, diseño lógico y diseño físico).
- *Implementación:* Después de diseñar y crear físicamente las tablas de la Base de Datos, se desarrolla toda la funcionalidad que el sistema requiere. Dicha funcionalidad se encontrará en forma de procedimientos almacenados, que tendrán control de excepciones y dejarán registros en una tabla de Logs.
- *Plan de Contingencia:* Análisis de los posibles riesgos que pueden interferir en el desarrollo del proyecto y de las acciones a tomar en caso de que estos sucedan.
- *Plan de Pruebas:* Teniendo en la BD tanto las tablas como los procedimientos, en esta fase se persigue comprobar el correcto funcionamiento del sistema (entendiéndose como tal que se cumplan todos los requisitos detectados).
- *Valoración económica y recursos necesarios:* Valoración del coste del producto a desarrollar y los recursos necesarios para ello.
- *Conclusiones:* conclusiones del trabajo de fin de carrera.

2. Requisitos de la B.D. Votaciones On-Line

Este apartado describe todos los requisitos y funcionalidades que tienen que cumplir el diseño de la BD, a parte también se proponen nuevas funcionalidades que el cliente deberá aprobar. Si se aprueban se realizará un estudio de implementación de estas nuevas funcionalidades, respetando la planificación y el presupuesto inicial. Si no fuera posible llevarlas a cabo en esta primera fase se realizarían en una segunda fase.

2.1 Iniciales

La comunidad Europea, dentro de la partida presupuestaria destinada a fomentar la participación ciudadana dentro del ámbito político Europeo, ha decidido abrir un concurso público para recibir propuestas sobre el diseño de una base de datos, que les sirva para almacenar la información para la futura aplicación de votaciones ciudadanas a través de Internet que quieren implementar. En el marco de colaboración con la UOC se centrará solo en el diseño de la base de datos, ya que la aplicación de gestión se desarrollará en una segunda fase del plan de sistemas de información de la comunidad Europea.

2.2 Funcionales

Los requisitos funcionales establecidos por el cliente para el modelo de datos a diseñar son los siguientes:

R1.- El modelo tiene que permitir guardar todos los datos asociados a una votación, estos serán como mínimo:

- Titulo de la votación, para la identificación rápida de los ciudadanos.
- Descripción de la votación y las diferentes opciones que podrán escoger los ciudadanos (cada votación tendrá un mínimo de dos opciones a escoger y máximo no establecido).
- Intervalo de fechas que estará abierta la votación para los ciudadanos.
- Fecha de publicación de la votación (será siempre un mínimo de una semana antes que se abra la votación a los ciudadanos para dejar un periodo de reflexión)
- Censo o censos electorales habilitados a votar en función del ámbito de la votación. Una votación podrá tener más de un censo asignado. Es decir, podríamos tener una votación que afectase a todos los ciudadanos de dos países de la comunidad Europea o de toda la comunidad o de una o más regiones concretas o de una o más pueblos o ciudades.
- El nombre del presidente de la mesa electoral virtual, será obligatorio en todos los casos.
- La dirección web opcional donde habrá explicaciones más detalladas de la votación.

R2.- El modelo tendrá que permitir guardar en el momento del cierre de la votación:

- El número de personas con capacidad de voto.
- Los votos totales que se han emitido realmente dentro de cada votación,
- El porcentaje de participación
- El numero de votos que ha tenido cada opción
- El porcentaje que representa, el número de votos que ha tenido cada opción, con respecto al total.

R3.- El sistema ha de permitir la gestión de los diferentes censos electorales, concretamente hay cuatro tipos de censos:

- Global. Incluye todos los ciudadanos mayores de 18 años correspondientes a todos los países de la comunidad Europea. Solo existirá una instancia en este censo.
- Estatal. Corresponde a todos los ciudadanos de un determinado país de la Comunidad Europea, en este caso se ha de guardar como meta data el código del país en cuestión según la codificación ISO 3166-1 alfa 2. Existirá una instancia de este censo por cada país de la comunidad.
- Regional. Incluye a todos los ciudadanos de una región concreta de Europa, como pueden ser los landers alemanes o las autonomías dentro de España. En este tipo de censo se guardara el país y el nombre de la región en cuestión.
- Local. Este tipo de censo incluye a todos los ciudadanos de un determinado pueblo o ciudad de Europa. En este tipo de censo se guardara el país, la región y el nombre del pueblo o ciudad a la que hace referencia.

R4.- Una votación puede tener más de un ceso electoral asociado, un ciudadano puede pertenecer a más de un censo, por ejemplo, un ciudadano de Antequera estará como mínimo en cuatro censos:

- El censo europeo de tipo Global.
- El censo de tipo Nacional asignado a España.
- El censo de tipo Regional asignado a Andalucía.
- El censo de tipo Local asignado a la ciudad de Antequera.

Se ha de controlar que una persona física, aunque pueda estar en diferentes censos, solo pueda emitir un único voto dentro de una votación concreta. Una vez emitido el voto no podrá ser modificado ni anulado.

R5.- Nuestra BD no guardará ni el nombre del ciudadano, ni otro dato que pueda violar el derecho a la votación secreta, estos datos se guardaran en otra base de datos separada físicamente y controlada por un departamento de seguridad especial y por lo tanto no entrará dentro del diseño de nuestra BD. Del ciudadano se guardará:

- código aleatorio asociado a cada ciudadano.
- la edad.
- código de país.
- código de región
- código de localidad.

Estos últimos con finalidades estadísticas y también para determinar los censos a los que pertenece el ciudadano.

R6.- La aplicación tendrá que disponer, como mínimo, de las funcionalidades siguientes, cumpliendo los requisitos expresados previamente:

- Procedimiento de ABM (Alta + Baja + Modificación) de las votaciones y de su información asociada: opciones, votos emitidos, censos asociados, etc.
- Procedimiento de ABM de los diferentes censos dentro del ámbito Europeo.
- Procedimiento de ABM de los ciudadanos y de su asignación a los censos.
- Procedimientos de consulta que permitan obtener:
  a) Dado un país por parámetro: el listado de todas las votaciones que se han producido (es decir, que han tenido asignado algún censo que pertenezca aquel país o censo global). En este listado se mostrará la siguiente información:
     - Titulo de la votación.
     - Fecha Inicio y Final de la votación.
     - Numero total de ciudadanos dentro de los censos asociados a la votación.
     - Estado en el que se encuentra la votación (en edición, pendiente, abierta, finalizada).
     - Si la votación esta en estado finalizado mostrar: % de participación total, opción ganadora y % de votos de la opción ganadora. En caso que no este finalizada tendrá que salir "-" en estos campos.

     Todo ordenado cronológicamente de forma ascendente por fecha de inicio de la votación.

  b) Listado de las 10 votaciones donde ha habido más diferencia porcentual de votos entre la opción más votada y la menos votada indicando:
     - Titulo de la votación.
     - Fecha Inicio y Final de la votación.
     - Opción más votada y su % respecto al total
     - Opción menos votada y su % respecto al total
     - Diferencia porcentual entre las dos opciones anteriores.

     Todo ordenado descendentemente por la diferencia porcentual de las dos opciones.

  c) El mismo listado anterior pero en este caso las 10 votaciones que han tenido menos diferencia porcentual de votos entre la opción más votada y la menos.

  d) Dado el código aleatorio que identifica un ciudadano, el listado de todos los censos a los que pertenece. Esta funcionalidad servirá para que un ciudadano pueda consultar a que censos pertenece.

  e) Dado el código aleatorio que identifica un ciudadano, el listado de las votaciones en las que ha participado y en las que no. Esta funcionalidad servirá para que un ciudadano pueda consultar su histórico de participación en votaciones.

R7.- El modulo estadístico tendrá que dar respuesta a las consultas siguientes:

- Dado un país y un año concreto: el número de votaciones finalizadas que se han producido.
- Dado un censo y un año concreto: el valor medio del % de participación, teniendo en cuenta todas las votaciones asociadas a aquel censo electoral que han finalizado durante ese año.
- Dado un censo y un año concreto: el numero de votaciones que ha tenido asociado aquel censo durante ese año.
- Dado un año concreto: la votación que más participación ha tenido.
- Dado un año concreto: la localidad europea que más votos han emitido sus ciudadanos.
- Dado un año concreto: % de ciudadanos que nunca han votado.
- 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.
- Numero 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.

2.3 Nuevas Propuestas Funcionales

Al identificar todas las propuestas funcionales que nos han requerido en el documento inicial, proponemos en este apartado nuevas funcionalidades:

Una propuesta es realizar una BD Multi Idioma. Con el objetivo de mostrar la información al ciudadano en el idioma de su país o región. Tenemos una Unión Europea donde existen diversos idiomas, las votaciones y sus opciones se pueden definir en los idiomas donde se realizará la consulta. Así se mostrará la información al votante en el idioma que tenga definido, si no lo estuviera informado el idioma de la persona se mostraría el de su localidad o su región o su país en el idioma que primero se encontrara. El idioma asignado al país sería obligatorio.

Otra propuesta sería realizar varios listados:

- Dado un país y año por parámetro: Listar todas las votaciones finalizadas, y el % de participación distinguiendo entre sexos.
- Dado un país y año por parámetro: Listar todas las votaciones finalizadas, y el % de participación distinguiendo entre rango de edad.
- Dada una votación finalizada por parámetro, listar el número de votos emitidos de cada opción por rango de edad.

2.4 No Funcionales

Del enunciado se contemplan los siguientes requisitos no funcionales:

- Sistema de Gestión de B.D. El sistema de gestión escogido será Oracle.

- Política de seguridad. Se ha especificado, por una cuestión de privacidad en las votaciones de los ciudadanos que no se guardaría ningún dato que revelase la identidad del votante. A parte de esto no se especifica nada más de la política de seguridad, ante lo cual se deduce que el sistema seguirá una política de permisos por la cual los ciudadanos tendrán acceso a las funcionalidades descritas en los Requisitos Funcionales.
- Sistema Operativo. No se impone restricción sobre el Sistema Operativo.
- Política de backup de datos. No se contempla la realización de tareas periódicas de backup de los datos almacenados en la Base de Datos. El concurso solo se centra en el diseño de la misma, por lo que la política de backup deberá de fijarse cuando se implante este modelo de datos.
- Utilización de procedimientos almacenados:
    - Control de excepciones
    - Devolución de parámetro de salida indicando si ha ido bien o mal el procedimiento.
- Tabla de logs para almacenar resultado, al ejecutar, los procedimientos.
- Carga del sistema. Aunque no se mencione, al llevarse la gestión individual de los censos, se prevé una carga inicial al sistema de algunos de los países, regiones y localidades de la Unión Europea.

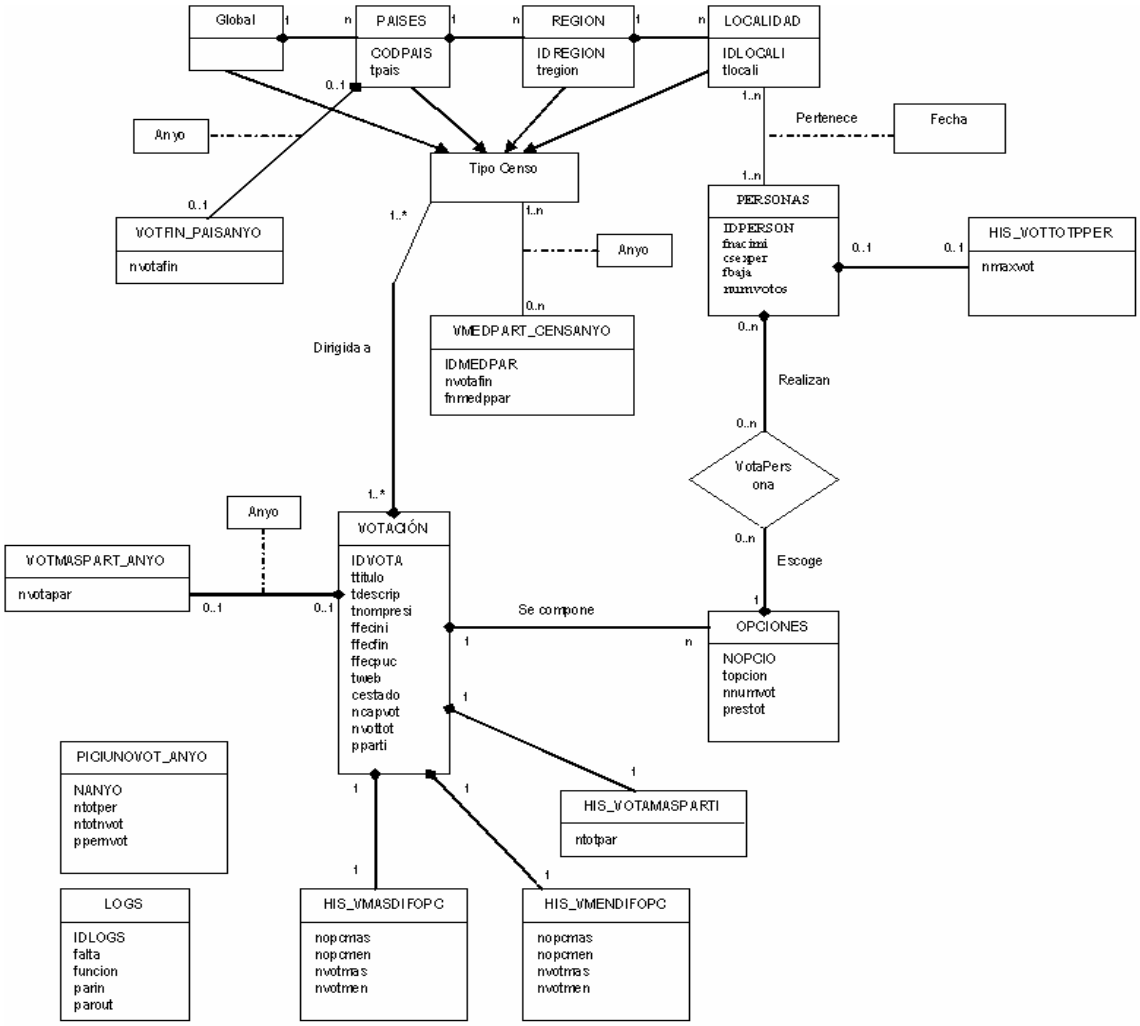
3. Diseño de la B.D. Votaciones On-Line

El diseño del sistema se abordará en tres etapas:

- Diseño conceptual: Buscaremos un diseño independiente de la tecnología que vamos a implementar. Obtendremos un modelo UML como resultado de esta etapa.
- Diseño lógico: Transformaremos el modelo UML en un modelo relacional, ya que utilizaremos como tecnología un SGBD relacional.
- Diseño físico: Buscaremos optimizar la eficiencia y completar el modelo con algunos aspectos físicos dependientes del SGBD utilizado.

3.1 Diseño Conceptual

3.1.1 Diagrama UML



3.1.2 Identificación de Entidades

Tras revisar los requisitos expuestos anteriormente, se identifican tres tipos de entidades:
Principales:
- PAISES
- REGION
- LOCALIDAD
- VOTACION
- OPCIONES
- PERSONAS
- VOTAPERSONA

Intermedias:
- VOTACENSO
- PERSOCENSO

Auxiliares:
- LOGS
- VOTFIN_PAISANYO
- VMEDPART_CENSANYO
- VOTMASPART_ANYO
- VOTMASP_LOCANY
- PCIUNOVOT_ANYO
- HIS_VOTMASPARTI
- HIS_VMASDIFOPC
- HIS_VMENDIFOPC
- HIS_VOTTOPPER

3.1.3 Atributos de las Entidades

A continuación se detallan los atributos de las entidades que se han detectado:

PAISES
<u>codpais</u>, tpais

REGION
<u>codpais, idregion</u>, tregion

LOCALIDAD
<u>codpais, idregion, idlocali</u>, tlocali

VOTACION
<u>idvota</u>, ttitulo, tdescrip, finicio, ffinal, fpubli, tnompresi, tweb, cestado, npervot, ntotvot, pparti

OPCIONES
<u>idvota, nopcio</u>, topcion, nnumvot, prestot

VOTACENSO
idvota, codpais, idregion, idlocali

PERSONAS
idperson, fnacimi, csexper, fbaja, numvotos

PERSOCENSO
idperson, codpais, idregion, idprovin, falta, fbaja

VOTAPERSONA
Idperson, idvota, nopcio, fvota, nedadvota

LOGS
idlogs, falta, funcion, parin, parout

VOTFIN_PAISANYO
codpais, nanyo, nvotafin

VMEDPART_CENSANYO
idvmedpart, nanyo, codpais, idregion, idlocali, nvotafin, nmedppar

VOTMASPART_ANYO
nanyo, idvota, nvotapar

VOTMASP_LOCANY
nanyo, codpais, idregion, idlocali, idvota, nvotmpar

PCIUNOVOT_ANYO
nanyo, ntotper, ntotnvot, ppernvot

HIS_VOTMASPARTI
idvota, ntotpar

HIS_VMASDIFOPC
idvota, idopcmas, idopcmen, nvotmas, nvotmen, pdifer

HIS_VMENDIFOPC
idvota, idopcmas, idopcmen, nvotmas, nvotmen, pdifer

HIS_VOTTOPPER
idperson, nmaxvot

3.2 Diseño Lógico

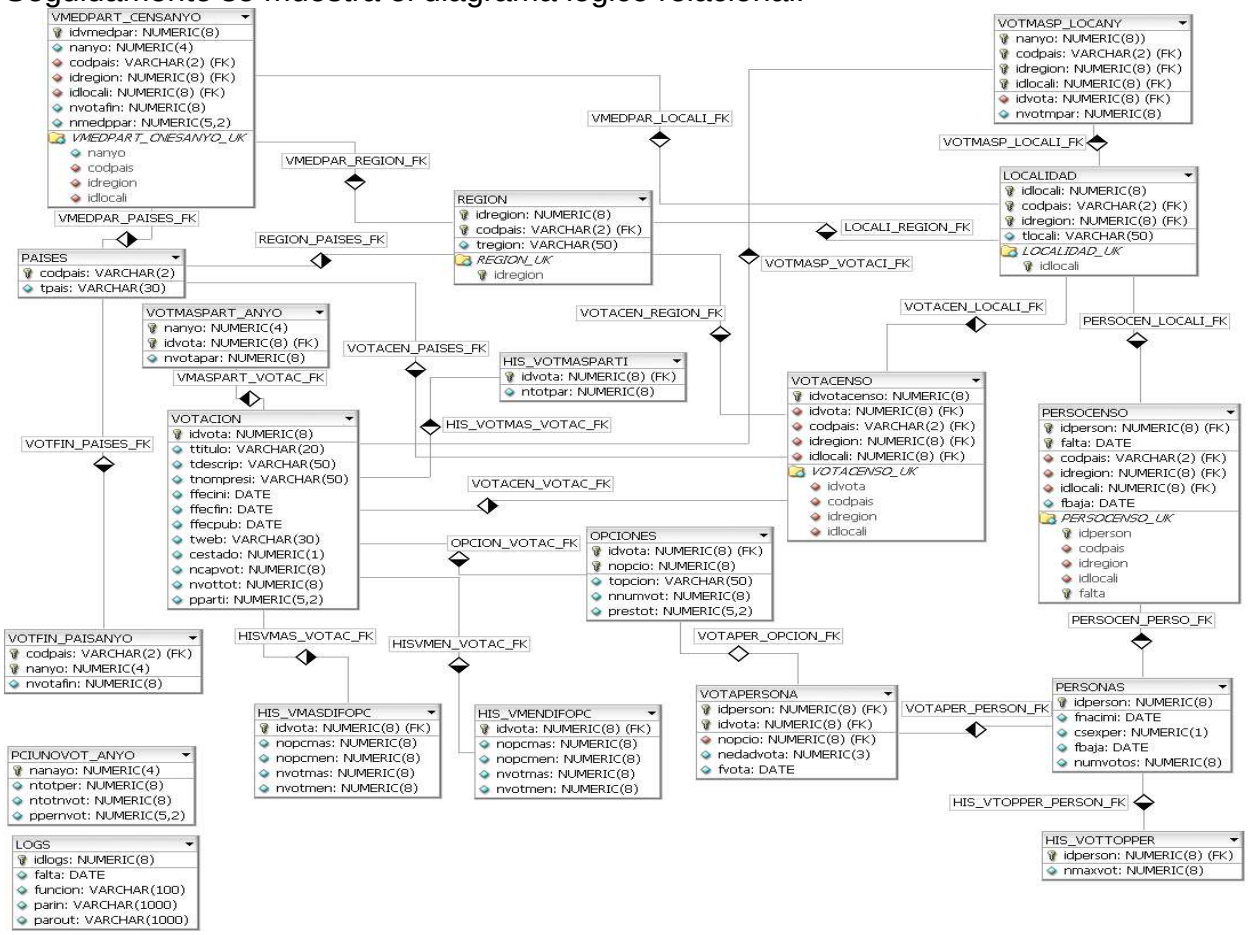
A partir del resultado del paso anterior, se procede a transformarlo en un modelo lógico relacional más afín, siguiendo las siguientes reglas:

- Las entidades originan relaciones.
- Las interrelaciones binarias 1:1 y 1:N originan claves foráneas.
- Las interrelaciones binarias M:N originan nuevas relaciones.

Análogamente al diseño conceptual, contempla entidades con las siguientes diferencias:

- Se indican todas las restricciones (PK=Primary Key=Clave Primaria, FK=Foreign Key=Clave foránea, NOT NULL=no nulo, CH=Check). Existen casos como en las entidades asociativas/intermedias en las que una clave primaria compuesta es a su vez clave externa.
- La relaciones y sus cardinalidades se mantienen iguales aunque se presentan de forma distinta: ya no se presentan las cardinalidades de forma explícita mediante 1 y * sino simplemente mediante una flecha. El lado de la flecha indica el lado uno de la relación o el lado donde está la entidad a consultar.
- Las relaciones expresan la relación clave primaria-externa entre cada lado de la relación.

Seguidamente se muestra el diagrama lógico relacional.



3.3 Diseño Físico

En este apartado realizamos la fase de diseño lógico, donde se debe realizar lo que en los próximos apartados se detalla.

3.3.1 Crear la Base de Datos

Vamos a suponer que donde se implante el modelo de datos ya hay un servidor funcionando, el paquete del SGBD Oracle ya está instalado y la base de datos, en la que hay que implementar el nuevo modelo, ya está creada.
Solo se crearan aquellos objetos que sean necesarios para el modelo a implementar, estos se detallan en los próximos apartados.

3.3.2 Tablespaces

Se crearán dos tablespaces, uno para datos y otro para índices, que actuarán de unidades lógicas de almacenamiento. Cada tabla se almacenará en el tablespace de datos así como cada índice será almacenado en el trablespace de índices.

Los tablespace se crearán gestionados localmente y con gestión automática de almacenamiento, debido a su alto rendimiento y a la sencillez de la asignación de espacio que nos permite “AUTOALLOCATE”. Estos tendrán un tamaño inicial de 10 Mb, que se estima suficiente para los datos iniciales que se prevé que contenga nuestra base de datos, en el caso de que se vea necesario más espacio serán los administradores de la base de datos los encargados de asignarles el espacio necesario.

3.3.3 Usuarios

En los requisitos iniciales no se comenta nada sobre la política de seguridad que se requiere para los usuarios, por lo que no se creará ningún usuario. Se utilizará el usuario SYSTEM. Con el podremos acceder a las diferentes tablas y funcionalidades de la aplicación, será propietario de todos los objetos (índices, tablas, procedimientos, etc.) de nuestro modelo de datos, teniendo los privilegios suficientes para poder crear aquellos objetos necesarios y acceder a los tablespaces de datos e índices.

3.3.4 Tablas

A partir del diseño lógico, crearemos las sentencias de Oracle para crear las tablas con sus correspondientes claves primarias y foráneas.

La nomenclatura utilizada ha sido la siguiente:

- Tablas: Se han escogido nombres que se asemejen a su contenido.
- Atributos:
  - Empiezan por “ID” aquellos campos que son secuencias.
  - Empiezan “C” aquellos que sean un código predefinido.

- Empiezan por la letra “T” aquellos campos que sean un texto.
- Empiezan por la letra “F” las fechas.
- Empiezan por la letra “N” los que son numéricos.
- Empiezan por la letra “P” los que son porcentajes.

- Clave principal: Nombre de la tabla + PK

- Claves foráneas: Nombre tabla origen + tabla destino + FK. Para que el nombre no fuera tan largo en algunos momentos se ha tenido que acotar los nombres de las tablas origen y destino.
- Claves Únicas: Nombre de la tabla + UK
- Checks: CK + Nombre del campo

El tipo y tamaño de los atributos de las tablas se han definido teniendo en cuenta cada caso particular. Se han marcado como NOT NULL los campos que son imprescindibles tener informados para un correcto funcionamiento del sistema.
Se ha utilizado la cláusula CHECK en aquellos campos tipificados para asegurar que su contenido se ajuste a los requisitos solicitados.

Las tablas que se crearán serán las que a continuación se detalla:

PAISES: Esta tabla contiene todos los países que están en la Comunidad Europea.

Atributos	Descripción	Formato	NN	Valores
CODPAIS	Código País	V(2)	S	codificación ISO. 3166-1
TPAIS	Nombre del País	V(30)	S	
FBAJA	Fecha de Baja	D		

Clave Principal	CODPAIS_PK: CODPAIS
Numero registros Insertados:	27 países
Actualización	Excepto la carga inicial no se producirán movimientos en años.

REGION: Esta tabla contiene todas las regiones que pertenecen a cada país

Atributos	Descripción	Formato	NN	Valores
CODPAIS	Código País	V(2)	S	Tabla PAISES
IDREGION	Identificador Región	N(8)	S	Secuencia SEQ_IDREGION
TREGION	Nombre de la Región	V(50)	S	
FBAJA	Fecha de Baja	D		

Clave Principal	REGION_PK : (CODPAIS, IDREGION)
Clave Ajena y Tabla	REGION_PAISES_FK : CODPAIS→ PAISES
Índice único	REGION_UK : IDREGION
Numero Registros Insertados:	Tantos como regiones tenga cada país
Actualización	Excepto la carga inicial no se producirán movimientos en años.

LOCALIDAD: Esta tabla contiene todas las localidades que pertenecen a cada región y país.

Atributos	Descripción	Formato	NN	Valores
CODPAIS	Código País	V(2)	S	Tabla REGION

Atributos	Descripción	Formato	NN	Valores
IDREGION	Identificador Región	N(8)	S	Tabla REGION
IDLOCALI	Identificador de la Localidad	N(8)	S	Secuencia SEQ_IDLOCALI
TLOCALI	Nombre de la Localidad	V(50)	S	
FBAJA	Fecha de Baja	D		

Clave Principal	LOCALIDAD_PK : (CODPAIS, IDREGION, IDLOCALI)
Clave Ajena y Tabla	LOCALI_REGION_FK : (CODPAIS, IDREGION) → REGION
Índice único	LOCALI_UK : IDLOCALI
Numero Registros Insertados:	Tantos como localidades tenga cada región / país
Actualización	Excepto la carga inicial no se producirán movimientos en años.

VOTACION: Contiene todos los datos necesarios para cada votación.

Atributos	Descripción	Formato	NN	Valores
IDVOTA	Identificador Votación	N(8)	S	Secuencia SEQ_IDVOTA
TTITULO	Titulo de la Votación	V(20)	S	
TDESCRIP	Descripción de la Votación	V(50)	S	
TNOMPRESI	Nombre del presidente	V(50)	S	
FFECINI	Fecha Inicio Votación	D	S	
FFECFIN	Fecha que finaliza la Votación	D	S	
FFECPUB	Fecha publicación Votación	D	S	
TWEB	Dirección Web de la votación	V(30)		
CESTADO	Estado que se encuentra la votación	N(2)	S	0-En edición 1-Pendiente 2-Abierta 3-Finalizada
NCAPVOT	Numero personas con capacidad de voto	N(8)		
NVOTTOT	Total de votos realizados	N(8)		
PPARTI	% de participación	N(5,2)		

Clave Principal	VOTACION_PK : IDVOTA
Check:	Ck_cestado in (0,1,2,3)
Numero Registros Insertados:	Tantos como consultas a los ciudadanos se requieran
Actualización	Varias veces al mes.

VOTACENSO: Censo o censos al cual va dirigida la votación.

Atributos	Descripción	Formato	NN	Valores
IDVOTACEN	Identificador del censo al cual va dirigida la votación	N(8)	S	Secuencia SEQ_IDVOTACEN
IDVOTA	Identificador de la votación	N(8)	S	VOTACIÓN
CODPAIS	Código del País	N(8)		PAISES
IDREGION	Identificador de la región	N(8)		REGION
IDLOCALI	Identificador de la localidad	N(8)		LOCALIDAD

Clave Principal	VOTACENSO_PK: IDVOTACEN
Clave Ajena y Tabla	VOTACEN_VOTAC_FK : IDVOTA →VOTACIÓN VOTACEN_PAISES_FK: CODPAIS → PAISES VOTACEN_REGION_FK: (CODPAIS,IDREGION)→REGION VOTACEN_LOCALI_FK: (CODPAIS,IDREGION,IDLOCALI)→LOCALIDAD
Índice único	VOTACENSO_UN: (IDVOTA,CODPAIS,IDREGION,IDLOCALI)
Numero Registros Insertados:	Mínimo uno por votación
Actualización	Varias veces al mes

OPCIONES: Contiene las opciones posibles de cada votación.

Atributos	Descripción	Formato	NN	Valores
IDVOTA	Identificador Votación	N(8)	S	TABLA VOTACION
NOPCIO	Identificador de Opción	N(8)	S	
TOPCION	Descripción de la Opción	V(50)	S	
NNUMVOT	Numero de votos obtenidos	N(8)		
PRESTOT	% respecto al total de participantes	N(5,2)		

Clave Principal	OPCIONES_PK: (IDVOTA, IDOPCIO)
Clave Ajena y Tabla	OPCION_VOTAC_FK: IDVOTA → VOTACION
Numero Registros Insertados:	Mínimo dos por votación
Actualización	Varias veces al mes

PERSONAS: Contiene los ciudadanos que pertenecen a la Comunidad Europea

Atributos	Descripción	Formato	NN	Valores
IDPERSON	Identificador de la persona	N(8)	S	Secuencia SEQ_IDPERSON
FNACIMI	Fecha nacimiento de la persona	D	S	
CSEXPER	Sexo de la persona	N(1)	S	1-Hombre 2-Mujer
FBAJA	Fecha de baja	D		
NUMVOTOS	Número de veces que ha votado la persona.	N(8)		

Clave Principal	PERSONAS_PK: IDPERSON
Check:	Ck_CSEXPER in (1,2)
Numero Registros Insertados:	Tantos como ciudadanos existen en la UE
Actualización	Varias veces al mes. Aunque se realice una carga inicial, siempre habrá nuevas incorporaciones o bajas.

PERSOCENSO: Censos de las personas

Atributos	Descripción	Formato	NN	Valores
IDPERSON	Identificador de la persona	N(8)	S	PERSONAS
FALTA	Fecha de Alta en el censo	D	S	
FBAJA	Fecha de Baja	D		
CODPAIS	Código del país al cual pertenece	V(2)	S	LOCALIDAD
IDREGION	Identificador de la región	N(8)	S	LOCALIDAD
IDLOCALI	Identificador de la Localidad	N(8)	S	LOCALIDAD

Clave Principal	PERSOCENSO_PK: (IDPERSON, FALTA)
Clave Ajena y Tabla	PERSOCEN_LOCALI_FK(CODPAIS, IDREGION, IDLOCALI) → LOCALIDAD PERSOCEN_PERSO_FK: IDPERSON → PERSONAS
Índice único	PERSOCENSO_UK: (IDPERSON, CODPAIS, IDREGION, IDLOCALI,FALTA)
Numero Registros Insertados:	Mínimo uno por persona.
Actualización	Varias veces al mes. Por nuevas incorporaciones o por cambios de censo.

VOTAPERSONA: Votaciones que realizan las persona.

Atributos	Descripción	Formato	NN	Valores

Atributos	Descripción	Formato	NN	Valores
IDPERSON	Identificador de la persona	N(8)	S	PERSONAS
IDVOTA	Identificador de la votación	N(8)	S	VOTACION
NOPCIO	Identificador de la opción	N(8)	S	
FVOTA	Fecha de votación	D	S	
NEDADVOTA	Edad que tiene la persona cuando vota	N(3)	S	

Clave Principal	VOTAPERSONA_PK: IDPERSON, IDVOTA
Clave Ajena y Tabla	VOTAPER_OPCION_FK: (IDVOTA, NOPCIO) → OPCIONES VOTAPER_PERSON_FK: IDPERSON → PERSONAS
Numero Registros Insertados:	Tantos como personas hayan ejercido su derecho al voto.
Actualización	Varias veces, según votaciones abiertas.

LOGS: Contiene todos los registros de las ejecuciones realizadas en la Base de Datos por nuestros procedimientos.

Atributos	Descripción	Formato	NN	Valores
IDLOGS	identificación de LOGS	N(8)	S	Secuencia SEQ_IDLOGS
FALTA	Fecha alta	D		
FUNCION	Función procesada	V(100)		
PARIN	Parámetros de entrada	V(1000)		
PAROUT	Parámetros de salida	V(1000)		

Clave Principal	LOGS_PK: IDLOGS
Numero Registros Insertados:	Tantos como ejecuciones de los procedimientos
Actualización	Frecuente. Cada vez que se utilice un procedimiento, función, etc de la base de datos.

VOTFIN_PAISANYO: Contiene el numero de votaciones finalizadas que se han producido en un país y año.

Atributos	Descripción	Formato	NN	Valores
CODPAIS	Código del país	V(2)	S	PAISES
NANYO	Año	N(4)	S	
NVOTAFIN	Numero de votaciones finalizadas	N(8)	S	

Clave Principal	VOTFIN_PK: (CODPAIS, NANYO)
Clave Ajena y Tabla	VOTFIN_FK: CODPAIS → PAISES
Numero Registros Insertados:	Un registro por país, año
Actualización	Cada vez que se cierra una votación

VMEDPART_CENSANYO: Contiene el porcentaje medio de participación por censo y año y el numero de votaciones asociadas al censo durante el año.
Los censos se distinguirán(por la clave única) de la siguiente forma:
-Solo se informa nanyo: Censo Global
-Solo se informa nanyo, codpais: Censo Estatal
-Solo se informa nanyo, codpais, idregion: Censo Regional
-Se informa nanyo, codpais, idregion, idlocali: Censo Local

Atributos	Descripción	Formato	NN	Valores
IDVMEDPAR	Identificador del registro	N(8)	S	SEQ_IDVMEDPAR
NANYO	Año	N(4)	S	

Atributos	Descripción	Formato	NN	Valores
CODPAIS	Código del país	V(2)		PAISES
IDREGION	Identificador de Región	N(8)		REGION
IDLOCALI	Identificador de localidad	N(8)		LOCALIDAD
NVOTAFIN	Nro de votaciones que han estado asociadas al censo durante el año	N(8)	S	
NMEDPPAR	Valor Medio del porcentaje de participación	N(5,2)	S	

Clave Principal	VMEDPAR_PK: IDVMEDPAR
Clave Ajena y Tabla	VMEDPAR_PAISES_FK: CODPAIS → PAISES VMEDPAR_REGION_FK: (CODPAIS,IDREGION)→REGION VMEDPAR_LOCALI_FK: (CODPAIS,IDREGION,IDLOCALI)→LOCALIDAD
Índice único	VMEDPAR_UK: (NANYO,CODPAIS,IDREGION,IDLOCALI)
Numero Registros Insertados:	Un registro por año, país, región y localidad.
Actualización	Cada vez que se cierra una votación

VOTMASPART_ANYO:

Atributos	Descripción	Formato	NN	Valores
NANYO	Año	N(4)	S	
IDVOTA	Identificador de votación que más participación a tenido.	N(8)	S	VOTACION
NVOTAPAR	Total de votación con más participación.	N(8)	S	

Clave Principal	VMASPART_PK: NANYO, IDVOTA
Clave Ajena y Tabla	VMASPART_VOTAC_FK: IDVOTA → VOTACION
Numero Registros Insertados:	Un registro por año, pero pueden haber más de uno en caso de que se produzcan empates.
Actualización	Cada vez que se cierra una votación

VOTMASP_LOCANY:

Atributos	Descripción	Formato	NN	Valores
NANYO	Año	N(4)	S	
CODPAIS	Código del país	V(2)	S	PAISES
IDREGION	Identificador de Región	N(8)	S	REGION
IDLOCALI	Identificador de localidad	N(8)	S	LOCALIDAD
IDVOTA	Identificador de la votación	N(8)	S	VOTACIÓN
NVOTMPAR	Numero de votos que más han emitido sus ciudadanos	N(8)	S	

Clave Principal	VOTMASP_PK: NANYO, CODPAIS, IDREGION, IDLOCALI, IDVOTA
Clave Ajena y Tabla	VOTMASP_LOCALI_FK: CODPAIS, IDREGION, IDLOCALI → LOCALIDAD VOTMASP_VOTACI_FK: IDVOTA → VOTACION
Numero Registros Insertados:	Un registro por año, pero puede haber más de uno porque pueden producirse empates.
Actualización	Cada vez que se cierra una votación

PCIUNOVOT_ANYO:

Atributos	Descripción	Formato	NN	Valores
NANYO	Año	N(4)	S	
NTOTPER	Numero total de personas que pueden votar.	N(8)	S	

Atributos	Descripción	Formato	NN	Valores
NTOTNVOT	Numero total de personas que nunca han votado.	N(8)	S	
PPERNVOT	% de personas que nunca han votado	N(8)	S	

Clave Principal	PCIUNOVOT_PK: NANYO
Numero Registros Insertados:	Un registro por año
Actualización	Cada vez que se cierra una votación.

HIS_VOTMASPARTI:

Atributos	Descripción	Formato	NN	Valores
IDVOTA	Identificador de la votación	N(8)	S	
NTOTPAR	Numero total de participación	N(8)	S	

Clave Principal	HIS_VOTMAS_PK: IDVOTA
Clave Ajena y Tabla	HIS_VOTMAS_VOTAC_FK: IDVOTA → VOTACIONES
Numero Registros Insertados:	Un registro, pero puede haber más de uno porque pueden producirse empates.
Actualización	Cada vez que se cierra una votación

HIS_VMASDIFOPC: Votación o votaciones que históricamente han tenido menos diferencia porcentual entre la opción mas y menos votadas

Atributos	Descripción	Formato	NN	Valores
IDVOTA	Identificador de la votación	N(8)	S	VOTACION
NOPCMAS	Identificador de la opción más votada	N(8)	S	
NOPCMEN	Identificador de la opción más votada	N(8)	S	
NVOTMAS	Total de votos de la opción más votada	N(8)	S	
NVOTMEN	Total de votos de la opción menos votada	N(8)	S	
PDIFER	% de la diferencia de votos	P(5,2)	S	

Clave Principal	HIS_VMASDIF_PK: IDVOTA
Clave Ajena y Tabla	HISVMAS_VOTAC_FK: IDVOTA → VOTACIONES
Numero Registros Insertados:	Un registro, pero puede haber más de uno porque pueden producirse empates.
Actualización	Cada vez que se cierra una votación

HIS_VMENDIFOPC: Votación o votaciones que históricamente han tenido menos diferencia porcentual entre la opción mas y menos votadas

Atributos	Descripción	Formato	NN	Valores
IDVOTA	Identificador de la votación	N(8)	S	VOTACION
NOPCMAS	Identificador de la opción más votada	N(8)	S	
NOPCMEN	Identificador de la opción más votada	N(8)	S	
NVOTMAS	Total de votos de la opción más votada	N(8)	S	
NVOTMEN	Total de votos de la opción menos votada	N(8)	S	
PDIFER	% de la diferencia de votos	P(5,2)	S	

Clave Principal	HIS_VMENDIF_PK: IDVOTA
Clave Ajena y Tabla	HISVMEN_VOTAC_FK: IDVOTA → VOTACIONES
Numero Registros Insertados:	Un registro, pero puede haber más de uno porque pueden producirse empates.
Actualización	Cada vez que se cierra una votación

HIS_VOTTOPPER: Numero máximo de votaciones en las que ha participado una persona/s en la historia de las votaciones en la UE.

Atributos	Descripción	Formato	NN	Valores
IDPERSON	Identificador de la persona	N(8)	S	PERSONAS
NMAXVOT	Numero máximo de votaciones en las que ha participado una persona	N(8)	S	

Clave Principal	HIS_VTOPPER_PK: idperson
Clave Ajena y Tabla	HIS_VTOPPER_PERSON_FK: idperson → PERSONAS
Numero Registros Insertados:	Un registro, pero puede haber más de uno porque pueden producirse empates.
Actualización	Cada vez que una persona realiza una votación.

3.3.5 Packages

A continuación detallo la especificación de los packages que se deben desarrollar, estos son los siguientes:

- PK_CENSOS: Este paquete contendrá todas las funciones que se requieren para dar de alta un censo:

Funciones	Descripción	Par IN	Par Out	Validaciones
INS_PAISES	Inserta en la tabla PAISES, los parámetros de entrada.	codpais tpais	RSP	- Parámetros informados. - Código país no exista
UPD_PAISES	Modifica la descripción de un país, en la tabla PAISES.	codpais tpais	RSP	- Parámetros informados. - Código país exista
DEL_PAISES	Borra un país, de la tabla PAISES. Física si no hay dependencias, Lógica si las hay informando fecha de baja en la tabla PAISES, en la tabla REGION y LOCALIDAD que dependan del PAIS.	codpais	RSP	- Parámetros informados. - Código país exista - Solo se dará de baja física si el país no tiene dependencias sino se informará la fecha de baja
INS_REGION	Inserta en la tabla REGION, los parámetros de entrada.	codpais tregion	RSP	- Parámetros informados. - Código país exista PAISES - No exista codpais, tregion
UPD_REGION	Modifica la descripción de una región de un país, en la tabla REGION.	codpais idregion tregion	RSP	- Parámetros informados. - Código país, idregion exista - tregión nueva no exista para ese país
DEL_REGION	Borra una región, de la tabla REGION. Física si no hay dependencias, Lógica si las hay informando fecha de baja en la tabla REGION y LOCALIDAD que dependan de la REGION	codpais idregion	RSP	- Parámetros informados. - Código país,idregion exista - Solo se dará de baja física si la región no tiene dependencias sino se informará la fecha de baja
INS_LOCALIDAD	Inserta en la tabla LOCALIDAD, los parámetros de entrada.	codpais idregion tlocali	RSP	- Parámetros informados. - codpais, idregion exista REGION - No exista codpais, idregion,tlocali

Funciones	Descripción	Par IN	Par Out	Validaciones
UPD_LOCALIDAD	Modifica la descripción de una localidad de una región, en la tabla LOCALIDAD.	codpais idregion idlocali tlocali	RSP	- Parámetros informados. - Código país, idregion, idlocali exista - tlocali nueva no exista para ese país, región
DEL_LOCALIDAD	Borra una localidad, de la tabla LOCALIDAD. Física si no hay dependencias, Lógica si hay dependencias modificando la fecha de baja de la LOCALIDAD.	codpais idregion idlocali	RSP	- Parámetros informados. - Código país, idregion, idlocali exista. - Solo se dará de baja física si la región no tiene dependencias sino se informará la fecha de baja

- PK_VOTACION: Contiene todas las funciones que son referente a las votaciones. Estas serán:

Funciones	Descripción	Par IN	Par Out	Validaciones
INS_VOTACION	Inserta una nueva votación, en la tabla VOTACIONES.	ttitulo tdescrip tnompre ffecini ffecfin ffecpub tweb	RSP	- Parámetros obligatorios informados. - fecha publicación + 7 no puede ser mayor que la fecha de inicio - fecha inicio no puede ser mayor que la fecha finalización de la votación
DEL_VOTACION	Borra una votación siempre y cuando esté en edición o pendiente. Borrará también los registros de VOTACENSO y OPCIONES, si existen, de la votación a borrar.	Idvota	RSP	- Exista la votación - Estado en Edición o Pendiente
UPD_VOTACION	Modifica todos los datos de una votación. Siempre y cuando esté en edición o pendiente.	idvota ttitulo tdescrip tnompre ffecini ffecfin ffecpub tweb	RSP	- Parámetros obligatorios informados. - fecha publicación + 7 no puede ser mayor que la fecha de inicio - fecha inicio no puede ser mayor que la fecha finalización de la votación - Se podrán modificar todos los datos si el estado = en edición o pendiente. - Si estado de la votación es abierto o cerrado no se podrá modificar nada.
UPD_VOTACION	Modifica algunos datos de una votación. Siempre y cuando no esté finalizada la votación.	Idvota tnompre ffecfin tweb	RSP	- Parámetros obligatorios informados. - fecha inicio no puede ser mayor que la fecha finalización de la votación - Si estado = cerrado no se podrá modificar nada.

Funciones	Descripción	Par IN	Par Out	Validaciones
UPD_ESTADO_VOTA	Actualiza el estado de una votación. La transición de estados será: 0-en edición -> 1-pendiente -> 2-abierto -> 3-finalizado. Modifica los datos de la votación, numero de personas con capacidad de voto, total de votos realizados y % de participación, cuando la votación se finalice. Los campos a modificar serán los siguientes: - en VOTACION: Calcula y actualiza los campos ncapvot, nvottot, pparti. ncapvot: se actualizará con la suma de todas las personas que están en el censo de la votación PERSOCENSO y VOTACENSO. nvottot: se actualizará con todas las votaciones realizadas a esa votación. pparti: se actuaalizará dividiendo (nvottot / ncapvot) * 100 - en OPCIONES: Calcula y actualiza los campos nnumvot, prestot. nnumvot: se actualizará con el total de votos que ha obtenido esa opción. pparti: se actualizará con (nnumvot / votacion.nvottot) * 100 - en PCIUNOVOT_ANYO Se buscarán todas aquellas personas que pueden ejercer su derecho al voto y no lo han hecho a través de PERSOCENSO, VOTACENSO, asociados a las votaciones VOTACION que se han producido en el año de petición, sin que se repitan los censos ya que se duplicarían las personas. Y no existan en la tabla VOTAPERSONA. Este número lo dividirá el número total de personas que han podido ejercer el voto en ese año (sin repetir persona). Si no existe registro para ese año se dará de alta. Si existe y la nueva es mayor se insertará (borrando las existentes), si es igual se insertará un nuevo registro y si es menor no hará nada.	idvota cestado	RSP	- Exista la votación. - Existan como mínimo dos OPCIONES de esta votación. - cestado existente sea menor que nuevo cestado.
INS_OPCION	Inserta una nueva opción, en la tabla OPCIONES.	Idvota nopcio topcion	RSP	- Parámetros obligatorios informados. - Exista idvota en VOTACIONES - El estado de la votación esté en edición o pendiente. - No exista el registro a insertar
UPD_OPCION	Modifica la descripción de una opción de una votación.	Idvota nopcio topcion	RSP	- Parámetros obligatorios informados - Solo se podrá modificar si el estado = en edición o pendiente. - Exista el registro en OPCIONES
DEL_OPCION	Borra una opción de la votación. Se podrá realizar si el estado de la votación esté en edición o pendiente.	Idvota nopcion	RSP	- Exista el registro en OPCIONES - Estado en Edición o Pendiente

Funciones	Descripción	Par IN	Par Out	Validaciones
INS_VOTACENSO	Inserta los censos a los que va dirigida una votación dada. Si solo viene informado idvota. Significa que va dirigido a toda la comunidad(censo global). Si solo viene informado idvota+codpais. Va dirigido al censo estatal del país. Puede haber más de un país Si viene informado idvota+codpais+region. Va dirigido a esa región Censo regional. Puede haber más de una región. Si viene informado idvota+codpais+region+localidad. Va dirigido a esa localidad. Censo local. Puede haber mas de una localidad.	Idvota codpais idregion idlocali	RSP	- idvota informado. - el idvota exista en la tabla VOTACIONES y cestado en Edición . pendiente o abierta (por si se quiere ampliar a otros lugares de la UE. - No exista el registro en VOTACENSO.
DEL_VOTACENSO	Borra un registro siempre y cuando el estado del votación sea en edición o pendiente.	Idvota codpais idregion idlocali	RSP	- Exista el registro

- PK_PERSONAS: Este paquete contiene todas las funciones que hacen referencia a las personas. Se compone de:

Funciones	Descripción	Par IN	Par Out	Validaciones
INS_PERSONAS	Inserta un nuevo votante, en la tabla PERSONAS.	fnacimi csexper	RSP	- Parámetros obligatorios informados. - csexper sea 1 o 2. - fecha de nacimiento no sea mayor que la fecha del sistema en el momento de dar el alta.
UPD_PERSONAS	Modifica los datos de una persona, así como la fecha de nacimiento y el sexo.	idperson fnacimi csexper	RSP	- Parámetros obligatorios informados - Exista el registro. - csexper sea 1 o 2. - Se puede modificar la fecha de nacimiento y el sexo.
DEL_PERSONAS	Borra una persona de forma lógica, para que no afecte los datos calculados en votaciones, opciones y el modulo estadístico. Se informará la fecha de baja que será la fecha del sistema en el momento que se pida la baja del registro.	idperson	RSP	- Exista el registro.
INS_PERSOCENSO	Inserta la localidad donde vive la persona. La fecha de alta se informará la del sistema en el momento de la inserción. Si existe ya un registro de la persona activo, no se podrá añadir otro, primero se tendrá que dar de baja con el procedimiento DEL_PERSOCENSO.	idperson codpais idregion idlocali	RSP	- Parámetros obligatorios informados. - Exista la persona. - El país esté vigente. - La región esté vigente. - La localidad exista y esté vigente. - No tenga un censo activo asignado, si es así primero tendrá que darse de baja el censo activo.
DEL_PERSOCENSO	Borra una persona del censo, de forma lógica actualizando la fecha de baja con la fecha del sistema.	idperson	RSP	- Persona exista - Exista un registro y esté activo, es decir, no esté informada la fecha de baja.

Funciones	Descripción	Par IN	Par Out	Validaciones
INS_VOTAPERSONA	Inserta un voto de una persona de una votación. La fecha del voto será la fecha del sistema en el momento de la votación, al igual que se calculará la edad de la persona en el momento de la votación.	idvota idperson nopcio	RSP	- Parámetros obligatorios informados. - Votación y opción exista en la tabla OPCIONES. - La persona exista en la tabla PERSONAS. - La persona no puede estar de baja. - La persona tiene que tener 18 años o más. - La persona pertenezca al censo que marca la VOTACIÓN en la fecha que se efectúa el voto.

3.3.6 Funciones

Para desarrollar el módulo de consultas, tendremos que desarrollar una serie de funciones que a continuación detallamos:

FLista_VotapPais

Descripción: Dado un el censo global o un país, listar todas las votaciones que se han producido en el país dado o en todos.
Parámetros:
Input : pcodpais. Código del país, si no se informa todos (significará censo global)
Output : tabla de un objeto
Tablas: VOTACIÓN, VOTCENESTA, OPCIONES
Información a extraer:
VOTACION.ttitulo : Titulo de la votación
VOTACION.ffecini: Fecha inicio de la votación
VOTACION.ffecfin: Fecha final de la votación
VOTACION.nncapvot: Numero de ciudadanos capacitados para votar.
VOTACION.nvottot: Numero de votos emitidos
VOTACION.cestado: Estado de la votación
VOTACION.pparti: Porcentaje de participación
Si VOTACION.cestado = 3 (finalizada)
OPCIONES. topcion: Descripción de la opción ganadora
OPCIONES. prestot: Porcentaje de votos de la opción ganadora
Si VOTACIÓN.cestado <> 3
-
-
Condiciones:
VOTACION.idvota = VOTCENESTA.idvota
VOTCENESTA.codpais BETWEEN nvl(pcodpais,'AA') AND nvl (pcodpais, 'ZZ')
OPCIONES.idvota = VOTACIÓN.idvota. La MAX(pnumvot)

Ordenado:
VOTACIÓN.ffecini

FTopTenMasVota

Descripción: De las votaciones finalizadas, listar 10 votaciones donde la diferencia porcentual, de la más votada a la menos votada, sea mayor.
Parámetros:
Input : ninguno
Output : tabla de un objeto
Tablas: VOTACIÓN, OPCIONES
Información a extraer:
VOTACION.ttitulo : Titulo de la votación
VOTACION.ffecini: Fecha inicio de la votación
VOTACION.ffecfin: Fecha final de la votación
OPCIONES.topcion: Titulo de la opción más votada
OPCIONES. prestot: Porcentaje de votos más votada
OPCIONES.topcion: Titulo de la opción menos votada
OPCIONES. prestot: Porcentaje de votos menos votada
Diferen : diferencia entre prestot(masvotada) – prestot(mensovotada)
Condiciones:
VOTACION.cestado = 3
La MAX(pnumvot) de OPCIONES.idvota = VOTACIÓN.idvota.
La MIN(pnumvot) de OPCIONES.idvota = VOTACIÓN.idvota.
ROWNUM <= 10
Ordenado:
diferen desc

FTopTenMenVota

Descripción: De las votaciones finalizadas, listar 10 votaciones donde la diferencia porcentual, de la más votada a la menos votada, sea menor.
Parámetros:
Input : ninguno
Output : tabla de un objeto
Tablas: VOTACIÓN, OPCIONES
Información a extraer:
VOTACION.ttitulo : Titulo de la votación
VOTACION.ffecini: Fecha inicio de la votación
VOTACION.ffecfin: Fecha final de la votación
OPCIONES.topcion: Titulo de la opción más votada
OPCIONES. prestot: Porcentaje de votos más votada
OPCIONES.topcion: Titulo de la opción menos votada
OPCIONES. prestot: Porcentaje de votos menos votada
Diferen : diferencia entre prestot(masvotada) – prestot(mensovotada)
Condiciones:
VOTACION.cestado = 3
La MAX(pnumvot) de OPCIONES.idvota = VOTACIÓN.idvota.
La MIN(pnumvot) de OPCIONES.idvota = VOTACIÓN.idvota.
ROWNUM <= 10
Ordenado:
diferen asc

FCensoPersona

Descripción: Dado el identificador de la persona, listar el censo al que pertenece y los censos a los que ha pertenecido una persona.
Parámetros:
Input : pidperson. Identificador de la persona
Output : tabla de un objeto
Tablas: PERSONAS, PERSOCENSO, PAISES, REGION, LOCALIDAD
Información a extraer:
PERSONAS.idperson : identificador de la persona
PERSONAS.fnacimi: Fecha de nacimiento
PERSONAS.csexper: Sexo de la persona (1-hombre, 2-mujer)
PERSOCENSO.falta: Fecha de Alta
PERSOCENSO. fbaja: Fecha de Baja
PAISES.codpais: Código del país
PAISES.tpais: Nombre del País
REGION.tregión: Nombre de la Región
LOCALIDAD.tlocali: Nombre de la Localidad

Condiciones:
PERSONAS.idperson = pidperson
PERSONAS.idperson = PERSOCENSO.idperson
PERSOCENSO.idpais = PAISES.idpais
PERSOCENSO.idpais = REGION.idpais
PERSOCENSO.idregio = REGION.idregio
PERSOCENSO.idpais = LOCALIDAD.idpais
PERSOCENSO.idregio = LOCALIDAD.idregio
PERSOCENSO.idlocali = LOCALIDAD.idlocali

Ordenado:
falta

FVotaPersona

Descripción: Dado el identificador de la persona, listar todas las votaciones en las que ha participado y en las que ha podido participar y no lo ha echo.
Parámetros:
Input: pidperson. Identificador de la persona
Output: tabla de un objeto
Tablas: PERSONAS, PERSOCENSO, VOTACION, VOTAPERSONA
Información a extraer:
PERSONAS.idperson : identificador de la persona
PERSONAS.fnacimi: Fecha de nacimiento
PERSONAS.csexper: Sexo de la persona (1-hombre, 2-mujer)
PERSOCENSO.falta: Fecha de Alta
PERSOCENSO. fbaja: Fecha de Baja
VOTACION.ttitulo: Titulo de la votación
VOTACIÓN.ffecini: Fecha Inicio de la votación
VOTACIÓN.ffecfin: Fecha final de la votación
Participo: (‘SI’existe en VOTAPERSONA, ‘No’ si no existe en VOTAPERSONA)

Condiciones:
PERSONAS.idperson = pidperson
PERSONAS.idperson = PERSOCENSO.idperson
PERSOCENSO.codpais = nvl(VOTACENSO.codpais, PERSOCENSO.codpais)
PERSOCENSO.idregion = nvl(VOTACENSO.idregion,PERSOCENSO.idregion)
PERSOCENSO.idlocali = nvl(VOTACENSO.codpais, PERSOCENSO.idlocali)
VOTACION.idvota = VOTACENSO.idvota
VOTACIÓN.ffecini este entre PERSOCENSO.falta y PERSOCENSO.fbaja.

Ordenado:
idperson

3.3.7 Triggers

TR_UPD_ESTADO_VOTACION

Se activa cuando el campo VOTACION.cestado = 3. Es decir, cuando se finaliza una votación.

El propósito de este TRIGGER es actualizar el módulo estadístico y algunos datos de las consultas. Después de finalizar la votación se realizará:
- en VOTFIN_PAISANYO
Si no existe registro para el país, año (año de la fecha inicio de la votación), lo insertará inicializando a 1 el campo nvotafin.
Si existe el registro sumará 1 al campo nvotafin del país y año de la votación finalizada.
El país lo extrae de la tabla VOTACENSO, si el codpais no estuviera informado (censo global), leerá toda la tabla de países insertándolos todos.
El año lo extrae de la fecha inicio de la votación.

- en VMEDPART_CENSANYO
Si no existe registro para el censo, año (año de la fecha inicio de la votación), lo insertará inicializando el campo pparmig con el campo censo.pparti % de participación de la votación que se ha cerrado y el campo nvotafin = 1.
Si existe para el censo, año, actualizará el campo pparmig (sumando todos los % de participación censo.pparti que pertenecen al censo que se ha cerrado y dividirlo con el numero de votaciones que se han dirigido sobre ese censo + 1 vmedpart_censanyo. nvotafin).

- en VOTMASPART_ANYO
Si no existe registro para ese año(año de la fecha inicio de la votación) se insertará la votación y el total de participación de la tabla votaciones.
Si existe se mirará si la votación finalizada sea de más participación que la que ya estaba almacenada, si es así se modificará con la finalizada, sino no se hará nada.

- en VOTMASP_LOCANY

Si no existe registro para ese año(año de la fecha inicio de la votación) se insertará la localidad, votación y el total de participación de la tabla votaciones y votacioncenso finalizada.
Si existe se mirará si la votación finalizada sea de más participación que la que ya estaba almacenada, si es así se modificará con la finalizada, sino no se hará nada.

- en HIS_VOTMASPARTI
Si no existe registro para ese año(año de la fecha inicio de la votación) se insertará la localidad, votación y el total de participación de la tabla votaciones y votacioncenso finalizada.
Si existe y la nueva es mayor se insertará (borrando las existentes), si es igual se insertará un nuevo registro y si es menor no hará nada.

- en HIS_VMASDIFOPC
Si no existe registro se insertará la votación, la opción máxima de votos, la opción mínima de votos y sus correspondientes votos leídos de las tablas OPCIONES de la votación que se ha cerrado. Calculando la diferencia porcentual.

Si existe se mirará si la diferencia es mayor que la existente, si es así borrará los existentes e insertará los datos de la votación que se ha cerrado, si es igual se insertará el registro y si es menor no hará nada.

- en HIS_VMENDIFOPC
Si no existe registro se insertará la votación, la opción máxima de votos, la opción mínima de votos y sus correspondientes votos leídos de las tablas OPCIONES de la votación que se ha cerrado. Calculando la diferencia porcentual.

Si existe se mirará si la diferencia es menor que la existente, si es así borrará los existentes e insertará los datos de la votación que se ha cerrado, si es igual se insertará el registro y si es menor no hará nada.

TR_INS_VOTAPERSONA

Se activa cuando se inserta un nuevo registro en VOTAPERSONA. Se sumará uno en el campo numvotos de la tabla PERSONAS que corresponda a la persona que acaba de votar.
Después realizará una consulta en la tabla HIS_VOTTOPPER, si existen registros mirará si es mayor que las votaciones que lleva la persona que acaba de votar, si es así no hará nada, sino borrará el registro e insertará el nuevo con los datos de la persona que acaba de votar.

4. Implementación

Una vez diseñadas las tablas, índices, claves, secuencias y código en forma de packages, procedimientos y triggers que tendrá nuestro sistema, en esta fase de implementación he desarrollado dicho código para que funcione completamente nuestro sistema.

Los procedimientos almacenados tendrán parámetros de entrada y parámetros de salida. Según los requisitos, todos los procedimientos tendrán un solo parámetro de salida, que será un indicativo de si ha ido todo bien (será un OK) o de si ha habido algún error (será una indicación de qué error se ha producido).

Todos los packages, procedimientos, trigger y secuencias vienen en scripts adjuntos al documento. En estos se podrá ver una explicación mas detallada de los mismos.

A continuación se detallan los scripts realizados.

4.1 Cr_tablas

Este script contiene todas las instrucciones necesarias para definir nuestra base de datos. En ella se definen todas las tablas que se emplearán, más la definición de los índices, claves, etc.

4.2 Cr_Secuencias

En este script definimos las secuencias necesarias que se emplearán en nuestra Base de Datos, más un trigger de inserción en la tabla logs para que actualice la secuencia asociada a ésta tabla.

4.3 Cr_tipos_objeto

En este script definimos los registros que utilizaremos para la realización de las consultas.

4.4 Funciones

Se definen las funciones para realizar las consultas. Estas se detallan a continuación:

4.4.1 FLISTA_VOTAPPAIS

DESCRIPCION: Dado un país por parámetro: listar todas las votaciones que se han producido en ese país o censo global es decir, que han tenido asignado algún censo que pertenezca aquel país o censo global.

PARAMETROS:
INPUT: pcodpais Varchar2(2): Código de País si no se informa se extraerán todos
OUTPUT: RSP VARCHAR2(300)
RETORNA:
Si es O.K. devuelve un Listado con las votaciones del país o censo global
Si no ha ido bien retornará un tipo de error:

'ERROR: (SQLERRM)'

4.4.2 FTOPTENMASVOTA

DESCRIPCION: Muestra un listado con las 10 votaciones con más diferencia porcentual de votos entre la opción más y menos votadas.

PARAMETROS:
INPUT: NINGUNO
OUTPUT: RSP VARCHAR2(300)
RETORNA:
Si ha ido bien devuelve un Listado de las 10 votaciones con MÁS diferencia Porcentual.
Si no ha ido bien retorna el siguiente error:
'ERROR: (SQLERRM)'

4.4.3 FTOPTENMENVOTA

DESCRIPCION: Muestra un listado con las 10 votaciones con menos diferencia porcentual de votos entre la opción más y menos votadas.

PARAMETROS:
INPUT: NINGUNO
OUTPUT: RSP VARCHAR2(300)
RETORNA:
Si ha ido bien devuelve un Listado de las 10 votaciones con MÁS diferencia Porcentual.
Si no ha ido bien retorna el siguiente error:
'ERROR: (SQLERRM)'

4.4.4 FCENSOPERSONA

DESCRIPCION: : Dado el identificador de persona por parámetro: lista todos los censos a los que ha pertenecido o pertenece.

PARAMETROS:
INPUT: pidperson NUMBER(8): identificador de la persona. Obligatorio
OUTPUT: RSP VARCHAR2(300)
RETORNA:
Si ha ido bien devuelve Listado de los censos de la persona
Si no ha ido bien retorna el siguiente error:
'ERROR: Tipo de Error'

4.4.5 FVOTAPERSONA

DESCRIPCION: Dado el identificador de una persona por parámetro: lista todas las votaciones en las que ha participado y las que ha podido participar y no lo ha echo.

PARAMETROS:
INPUT: pidperson NUMBER(8): identificador de la persona. Obligatorio
OUTPUT: RSP VARCHAR2(300)
RETORNA:
Si ha ido bien devuelve un Listado de todas las votaciones que ha participado y las que no.
Si no ha ido bien retorna el siguiente error:
'ERROR: Tipo de Error'

4.5 Procedimientos de Alta, Baja y Modificación

Se han desarrollado un serie de packages que contienen procedimientos almacenados los cuales realizan el mantenimiento del bloque al que van asociados. Cada uno de estos packages se encuentra en el script asociado que se adjunta al documento. Estos son los siguientes:

4.5.1 PK_CENSOS

Contiene la gestión de toda la información relacionada con los censos.
El Packages se compone de los siguientes procedimientos

4.5.1.1 Gestión de Países (INS_PAIS)

DESCRIPCION: Da de alta un nuevo País, si:
- Todos los campos están informados.
- El registro no Exista.

PARAMETROS:
INPUT: Obligatorio: pcodpais: VARCHAR2(02). Código del País.
Ptpais: VARCHAR2(30). Nombre del País.
OUTPUT: RSP VARCHAR2(300)
PRECONDICION: No existe el país a crear
POSTCONDICION: Se da de alta el país
COMENTARIOS: Los parámetros de entrada se convertirán en mayúsculas y sin espacios
RETORNA:
'OK' si la operación se ha realizado con éxito
'ERROR' Si:
'ERROR: El Código del País no puede ser nulo'
'ERROR: La Descripción del País no puede ser nula'
'ERROR: El País ya ha sido dado de alta'
'ERROR: (SQLERRM)'

4.5.1.2 Gestión de Países (DEL_PAIS)

DESCRIPCION: Da de baja un País, si previamente supera los siguientes controles:

- El campo codpais este informado.
- El País exista en la Base de datos.
- El País no tenga dependencias
- Si tiene dependencias se realizará una baja lógica en cascada Informando la fecha de baja en todas sus dependencias, sino se Realizará la baja física.

PARAMETROS:

INPUT: Obligatorio: pcodpais: VARCHAR2(02). Código del País.

OUTPUT: RSP VARCHAR2(300)

PRECONDICION: Existe el país a dar de baja.

POSTCONDICION: Se da de baja el país

COMENTARIOS: Los parámetros de entrada se convertirán en mayúsculas y sin espacios

RETORNA:

'OK' si la operación se ha realizado con éxito

'ERROR' Si:

'ERROR: El Código del País no puede ser nulo'

'ERROR: El País no existe o está dado de baja'

'ERROR: (SQLERRM)'

4.5.1.3 Gestión de Países (UPD_PAIS)

DESCRIPCION: Realiza una modificación en la descripción de un País. Sí:

- El registro Existe.
- La nueva descripción viene informada.

PARAMETROS:

INPUT: Obligatorio: pcodpais: VARCHAR2(02). Código del País.

Ptpais: VARCHAR2(30). Nombre del País.

OUTPUT: RSP VARCHAR2(300)

PRECONDICION: Exista el país a modificar

POSTCONDICION: Se modifica la descripción del país

COMENTARIOS: Los parámetros de entrada se convertirán en mayúsculas y sin espacios

RETORNA:

'OK' si la operación se ha realizado con éxito

'ERROR' Si:

'ERROR: El Código del País no puede ser nulo'

'ERROR: La Descripción del País no puede ser nula'

'ERROR: El País no existe o está dado de baja'

'ERROR: (SQLERRM)'

4.5.1.4 Gestión de Regiones (INS_REGION)

DESCRIPCION: Da de alta una nueva Región a un país, si cumple que:
- Todos los campos están informados.
- El país exista
- El registro no Exista.

PARAMETROS:
INPUT: Obligatorio: pcodpais: VARCHAR2(02). Código del País.
ptregion: VARCHAR2(50). Nombre de la Región.
OUTPUT: RSP VARCHAR2(300)
PRECONDICION: No existe la región del país crear
POSTCONDICION: Se da de alta la región del país
RETORNA:
'OK' si la operación se ha realizado con éxito
'ERROR' Si:
'ERROR: El Código del País no puede ser nulo'
'ERROR: La Región no puede ser nula'
'ERROR: El País no existe o está dado de baja'
'ERROR: La región ya ha sido dada del alta para el país'
'ERROR: (SQLERRM)'

4.5.1.5 Gestión de Regiones (DEL_REGION)

DESCRIPCION: Da de baja una Región de un país, si cumple que:
- El campo idregion esta informado.
- La región exista en la Base de datos
- Si tiene dependencias se realizará una baja lógica en cascada Informando la fecha de baja en todas sus dependencias, sino se Realizará la baja física.

PARAMETROS:
INPUT: Obligatorio: pidregion: NUMBER(08). Identificador de la región.
OUTPUT: RSP VARCHAR2(300)
PRECONDICION: Exista la región a dar de baja
POSTCONDICION: Se da de baja la región
RETORNA:
'OK' si la operación se ha realizado con éxito
'ERROR' Si:
'ERROR: El identificador de Región no puede ser nulo'
'ERROR: La Región no existe o está dada de baja'
'ERROR: (SQLERRM)'

4.5.1.6 Gestión de Regiones (UPD_REGION)

DESCRIPCION: Realiza una modificación en la descripción de una Región, si cumple que:
- El registro Existe.

- La nueva descripción viene informada.

PARAMETROS:
INPUT: Obligatorio: pidregion NUMBER(08): identificador de la región
ptregion VARCHAR2(50): Descripción de la región
OUTPUT: RSP VARCHAR2(300)
PRECONDICION: Exista la región a modificar
POSTCONDICION: Se modifica la descripción de la región
RETORNA:
'OK' si la operación se ha realizado con éxito
'ERROR' Si:
'ERROR: El identificador de la Región no puede ser nulo'
'ERROR: La Descripción de la Región no puede ser nulo'
'ERROR: La Región no existe o está dada de baja '
'ERROR: (SQLERRM)'

4.5.1.7 Gestión de Localidades (INS_LOCALI)

DESCRIPCION: Da de alta una nueva Localidad a un país, si cumple que:
- Todos los campos están informados.
- El país exista
- La región exista
- El registro no Exista.

PARAMETROS:
INPUT: Obligatorio: pcodpais: VARCHAR2(02). Código del País.
pidregion: NUMBER(08). identificador de la Región.
ptlocali: VARCHAR2(50). Nombre Localidad.
OUTPUT: RSP VARCHAR2(300)
PRECONDICION: No existe la localidad para la región/pais
POSTCONDICION: Se da de alta la localidad
RETORNA:
'OK' si la operación se ha realizado con éxito
'ERROR' Si:
'ERROR: El Código del País no puede ser nulo'
'ERROR: La Región no puede ser nula'
'ERROR: La descripción de la localidad no puede ser nula’
‘ERROR: El país no existe o está dado de baja
‘ERROR: La región/país no existe o está dado de baja'
'ERROR: la localidad ya ha sido dada de alta para esta región/país'
'ERROR: (SQLERRM)'

4.5.1.8 Gestión de Localidades (DEL_LOCALI)

DESCRIPCION: Da de baja una localidad, si cumple que:
- El campo idlocali esta informado.
- La localidad existe en la Base de datos
- Si tiene dependencias se realizará una baja lógica en cascada

Informando la fecha de baja en todas sus dependencias, sino se Realizará la baja física.

PARAMETROS:
INPUT: Obligatorio: pidlocali: NUMBER(08). identificador de la Localidad.
OUTPUT: RSP VARCHAR2(300)
PRECONDICION: Exista la localidad a dar de baja
POSTCONDICION: Se da de baja la localidad
RETORNA:
'OK' si la operación se ha realizado con éxito
'ERROR' Si:
'ERROR: El identificador de localidad no puede ser nulo'
'ERROR: La Localidad no existe o está dada de baja'
'ERROR: (SQLERRM)'

4.5.1.9 Gestión de Localidades (UPD_LOCALI)

DESCRIPCION: Realiza una modificación en la descripción de una Localidad, si cumple que:
- El registro Existe.
- La nueva descripción viene informada.

PARAMETROS:
INPUT: Obligatorio: pidlocali NUMBER(08): identificador de la localidad
ptlocaliVARCHAR2(50): Descripción de la localidad
OUTPUT: RSP VARCHAR2(300)
PRECONDICION: Exista la localidad a modificar
POSTCONDICION: Se modifica la descripción de la localidad
RETORNA:
'OK' si la operación se ha realizado con éxito
'ERROR' Si:
'ERROR: El identificador de la Localidad no puede ser nulo'
'ERROR: La Descripción de la Localidad no puede ser nulo'
'ERROR: La Localidad no existe o está dada de baja'
'ERROR: (SQLERRM)'

4.5.2 PK_PERSONAS

Contiene la gestión de toda la información relacionada con los ciudadanos.
El Packages se compone de los siguientes procedimientos

4.5.2.1 Gestión de Personas (INS_PERSONAS)

DESCRIPCION: Da de alta una persona, si cumple que:
- Todos los campos obligatorios están informados.
- las validaciones sean correctas

PARAMETROS:
INPUT: Obligatorio: pfnacimi: DATE. Fecha de nacimiento
pcsexper: NUMBER(01). Sexo.
OUTPUT: RSP VARCHAR2(300)
PRECONDICION: los campos obligatorios estén informados
POSTCONDICION: Se da de alta la persona
RETORNA:
'OK' si la operación se ha realizado con éxito
'ERROR' Si:
'ERROR: La fecha de nacimiento no puede ser nulo'
'ERROR: El sexo no puede ser nulo'
'ERROR: El valor del sexo es incorrecto. Valores correctos son 1-hombre, 2-mujer'
'ERROR: (SQLERRM)'

4.5.2.2 Gestión de Personas (DEL_PERSONAS)

DESCRIPCION: Da de baja una persona, si cumple que:
- El campo idperson este informado.
- La persona exista en la Base de datos
- La baja será lógica, para mantener el histórico

PARAMETROS:
INPUT: Obligatorio: pidperson NUMBER(8): identificador de la persona
OUTPUT: RSP VARCHAR2(300)
PRECONDICION: los campos obligatorios estén informados
POSTCONDICION: Se da de baja lógica la persona, para mantener el histórico
RETORNA:
'OK' si la operación se ha realizado con éxito
'ERROR' Si:
'ERROR: El identificación de la persona no puede ser nulo'
'ERROR: La persona no existe'
'ERROR: (SQLERRM)'

4.5.2.3 Gestión de Personas (UPD_PERSONAS)

DESCRIPCION: Realiza la modificación de los datos de una persona, si cumple que:

PARAMETROS:
INPUT: Obligatorio: pidperson NUMBER(8): identificador de la persona
pfnacimi DATE: Fecha de nacimiento
psexper NUMBER(1): Sexo
OUTPUT: RSP VARCHAR2(300)
PRECONDICION: Exista la persona a modificar
POSTCONDICION: Se realiza la modificación de la persona
RETORNA:
'OK' si la operación se ha realizado con éxito

'ERROR' Si:
'ERROR: El identificación de la persona no puede ser nulo'
'ERROR: La fecha de nacimiento de la persona no puede ser nulo'
'ERROR: El sexo no puede ser nulo'
'ERROR: El valor del sexo es incorrecto. Valores correctos son 1-hombre, 2-mujer'
'ERROR: La persona no existe'
'ERROR: (SQLERRM)'

4.5.2.4 Mantenimiento Censo - Personas (INS_PERSOCENSO)

DESCRIPCION: Inserta el censo de una persona a fecha de inserción: Si se cumplen que:
- Los campos obligatorios estén informados
- Exista la persona en la B.D.
- Exista el país y esté vigente
- Exista la región y esté vigente
- Exista la localidad y esté vigente
- Exista País/región/localidad y esté vigente
- No tenga un censo activo asignado, si es así primero tendrá que dar de baja ese censo activo.

PARAMETROS:
INPUT: Obligatorio: pidperson NUMBER(8): identificador de la persona
pcodpais VARCHAR(2): Código del país.
pidregion NUMBER(8): identificador de la región.
pidlocali NUMBER(8): identificador localidad.

OUTPUT: RSP VARCHAR2(300)
PRECONDICION: No exista el registro
POSTCONDICION: Se da de alta el registro
RETORNA:
'OK' si la operación se ha realizado con éxito
'ERROR' Si:
'ERROR: El identificador de la persona no puede ser nulo'
'ERROR: El Código del país no puede ser nulo'
'ERROR: El identificador de la región no puede ser nulo '
'ERROR: El identificador de la localidad no puede ser nulo'
'ERROR: La persona no existe'
'ERROR: La localidad no existe para la Región/País, o no está vigente'
'ERROR: La región no existe para el País o no está vigente'
'ERROR: El país no existe o no está vigente'
'ERROR: Esta persona ya tiene asignado un censo'
'ERROR: (SQLERRM)'

4.5.2.5 Mantenimiento Censo - Personas (DEL_PERSOCENSO)

DESCRIPCION: Da de baja un censo de una persona, si previamente supera los siguientes controles:

-El campo idpersona este informado
-La persona exista
-La persona tenga un censo asociado activo

PARAMETROS:

INPUT: Obligatorio: pidperson NUMBER(8): identificador de la persona

OUTPUT: RSP VARCHAR2(300)

PRECONDICION: Exista un censo activo para esa persona
POSTCONDICION: Se da de baja (física/lógica) el censo activo para esa persona.

RETORNA:
'OK' si la operación se ha realizado con éxito
'ERROR' Si:

'ERROR: El identificador de la persona no puede ser nulo'
‘ERROR: La persona no existe’
‘ERROR: No existe censo asociado activo para esta persona’
'ERROR: (SQLERRM)'

4.5.2.6 Mantenimiento Votaciones - Personas (INS_VOTAPERSONA)

DESCRIPCION: Inserta la opción escogida, por una persona en una votación, si cumple con los siguientes requisitos.

- Que la persona corresponda al censo de la votación.
- La persona tenga 18 años
- Que la persona no haya votado en la votación.
- Que exista la persona
- Que exista la votación y la opción escogida

PARAMETROS:

INPUT: Obligatorio: pidvota NUMBER(8): identificador de la votación
pidperson NUMBER(8): identificador de la persona
pnopcio NUMBER(8): opción de la votación.

OUTPUT: RSP VARCHAR2(300)
PRECONDICION: No exista el registro
POSTCONDICION: Se da de alta el registro
RETORNA:
'OK' si la operación se ha realizado con éxito
'ERROR' Si:

'ERROR: El Identificador de la votación no puede ser nulo'
'ERROR: El Identificador de la persona no puede ser nulo'
'ERROR: El numero de opción no puede ser nulo'

```
        'ERROR: La persona no existe'
        'ERROR: La votación no existe'
        'ERROR: La votación no está en estado Abierto'
        'ERROR: La opción para esta votación no existe'
        'ERROR: Esta persona no puede votar por no pertenecer al censo
         electoral'
        'ERROR: Esta persona ya ha ejercido el derecho a voto en esta
         votación'
        'ERROR: (SQLERRM)'
```

4.5.3 PK_VOTACION

Contiene la gestión de toda la información relacionada con las votaciones.
El Packages se compone de los siguientes procedimientos

4.5.3.1 Gestión de Votación (INS_VOTACION)

```
DESCRIPCION: Da de alta una votación, si cumple que:
        - Todos los campos obligatorios están informados.
        - las validaciones sean correctas

PARAMETROS:
        INPUT: Obligatorio: pttitulo VARCHAR2(20):  Titulo votación
                            ptdescrip VARCHAR2(50): Descripción votación
                            ptnompresi VARCHAR2(50): Nombre del presidente
                            pffecini DATE: Fecha inicio votación
                            pffecfin DATE: Fecha final votación
                            pffecpub DATE: Fecha de publicación de la votación
                Opcionales: pweb VARCHAR2(30): Nombre de la web
        OUTPUT: RSP VARCHAR2(300)
PRECONDICION: No exista la votación
POSTCONDICION: Se da de alta la votación
RETORNA:
'OK' si la operación se ha realizado con éxito
'ERROR' Si:
        'ERROR: Titulo votación no puede ser nulo'
        'ERROR: La Descripción de la votación no puede ser nula'
        'ERROR: El nombre del presidente no puede ser nulo'
        'ERROR: La fecha de inicio de la votación no puede ser nula'
        'ERROR: La fecha final de la votación no puede ser nula'
        'ERROR: La fecha final no puede ser anterior a la fecha inicio'
        'ERROR: La fecha de publicación no puede ser nula'
        'ERROR: La fecha de publicación tiene que ser como mínimo una
         semana antes a la fecha de inicio'
        'ERROR: (SQLERRM)'
```

4.5.3.2 Gestión de Votación (DEL_VOTACION)

DESCRIPCION: Da de baja una votación, si cumple que:
- El campo idvota este informado.
- las votación exista en la Base de datos
- La votación no esté abierta, ni finalizada

PARAMETROS:
INPUT: Obligatorio: pidvota NUMBER(08): identificador de votación
OUTPUT: RSP VARCHAR2(300)
PRECONDICION: Exista la votación
POSTCONDICION: Se da de baja la votación
RETORNA:
'OK' si la operación se ha realizado con éxito
'ERROR' Si:
'ERROR: El identificación de votación no puede ser nulo'
'ERROR: La votación no existe'
'ERROR: No se puede dar de baja una votación finalizada'
'ERROR: No se puede dar de baja una votación abierta'
'ERROR: (SQLERRM)"ERROR: (SQLERRM)'

4.5.3.3 Gestión de Votación (UPD_VOTACION)

DESCRIPCION: Realiza modificaciones en una votación, si cumple que:
- Todos los campos obligatorios están informados.
- El estado de la votación está en Edición o Pendiente. Se podrá modificar todos los campos (excepto la clave).
- No se podrá modificar ningún dato si el estado esta abierto o finalizado

PARAMETROS:
INPUT: Obligatorio: pidvota NUMBER(08): identificador de la votación
pttitulo VARCHAR2(20): Titulo votación
ptdescrip VARCHAR2(50): Descripción votación
ptnompresi VARCHAR2(50): Nombre del presidente
pffecini DATE: Fecha inicio votación
pffecfin DATE: Fecha final votación
pffecpub DATE: Fecha de publicación de la votación
Opcionales: pweb VARCHAR2(30): Nombre de la web
OUTPUT: RSP VARCHAR2(300)
PRECONDICION: Exista la votación a modificar
POSTCONDICION: Se realiza la modificación de la votación
RETORNA:
'OK' si la operación se ha realizado con éxito
'ERROR' Si:
'ERROR: El Identificador de la votación no puede ser nulo'
'ERROR: La votación no existe'
'ERROR: No se puede modificar una votación finalizada'
'ERROR: No se puede modificar todos los datos de una votación abierta'
'ERROR: Titulo votación no puede ser nulo'

'ERROR: La Descripción de la votación no puede ser nula'
'ERROR: El nombre del presidente no puede ser nulo'
'ERROR: La fecha de inicio de la votación no puede ser nula'
'ERROR: La fecha final de la votación no puede ser nula'
'ERROR: La fecha final no puede ser anterior a la fecha inicio'
'ERROR: La fecha de publicación no puede ser nula'
'ERROR: La fecha de publicación tiene que ser como mínimo una semana antes a la fecha de inicio'
'ERROR: (SQLERRM)'

4.5.3.4 Gestión de Votación (UPD_VOTACION)

DESCRIPCION: Realiza modificaciones en una votación, en la fecha finalización de la votación, nombre del presidente de mesa y dirección web, si cumple que:

- Todos los campos obligatorios están informados.
- El estado de la votación está en Edición, Pendiente o Abierta. Se podrá modificar los campos comentados.
- No se podrá modificar ningún dato si el estado esta finalizado

PARAMETROS:
INPUT: Obligatorio: pidvota NUMBER(08): identificador de la votación
ptnompresi VARCHAR2(50): Nombre del presidente
pffecfin DATE: Fecha final votación
Opcionales: pweb VARCHAR2(30): Nombre de la web
OUTPUT: RSP VARCHAR2(300)
PRECONDICION: Exista la votación a modificar
POSTCONDICION: Se realiza la modificación de la votación
RETORNA:
'OK' si la operación se ha realizado con éxito
'ERROR' Si:
'ERROR: El Identificador de la votación no puede ser nulo'
'ERROR: La votación no existe'
'ERROR: No se puede modificar una votación finalizada'
'ERROR: El nombre del presidente no puede ser nulo'
'ERROR: La fecha final de la votación no puede ser nula'
'ERROR: La fecha final no puede ser anterior a la fecha inicio'
'ERROR: (SQLERRM)'

4.5.3.5 Gestión de Votación (UPD_ESTADO_VOTA)

DESCRIPCION: Realiza modificaciones en el estado de una votación, si cumple que:

- Exista la votación.
- Existan como mínimo dos opciones de la votación a modificar.
- No se podrá modificar si el estado esta finalizado
- El estado modificado no puede ser menor que el actual

Si el estado pasa ha finalizado se actualizarán automáticamente los campos de votaciones:
- ncapvot NUMBER(8): Capacidad de votantes.
- nvottot NUMBER(8): Total de personas que han votado
- pparti NUMBER(5,2): Porcentaje de participación

Y por cada opción asociada a la votación, se actualizarán:
- nnumvot NUMBER(8): Numero de votos
- prestot NUMBER(5,2): Porcentaje respecto al total de votos

PARAMETROS:
INPUT: Obligatorio: pidvota NUMBER(08): identificador de la votación
pcestado NUMBER(02): Estado de la votación
OUTPUT: RSP VARCHAR2(300)
PRECONDICION: Exista la votación a modificar
POSTCONDICION: Se realiza la modificación del estado la votación
RETORNA:
'OK' si la operación se ha realizado con éxito
'ERROR' Si:
'ERROR: El Identificador de la votación no puede ser nulo'
'ERROR: La votación no existe'
'ERROR: No se puede modificar una votación finalizada'
'ERROR: No se puede retroceder el estado de la votación a uno inferior';
'ERROR: (SQLERRM)'

4.5.3.6 Gestión de Opciones (INS_OPCION)

DESCRIPCION: Da de alta una nueva Opción a una votación, si cumple que:
- Todos los campos Obligatorios están informados.
- La votación exista
- La votación no esté abierta ni finalizada

PARAMETROS:
INPUT: Obligatorio: pidvota: NUMBER(08). identificador de votación.
pnopcio: NUMBER(08). Numero de opción.
ptlopcion: VARCHAR2(50). Descripción de la opción.
OUTPUT: RSP VARCHAR2(300)
PRECONDICION: No exista la opción para la votación
POSTCONDICION: Se da de alta la opción
RETORNA:
'OK' si la operación se ha realizado con éxito
'ERROR' Si:
'ERROR: Identificador de votación no puede ser nulo'
'ERROR: El identificador de opción no puede ser nulo'
'ERROR: Descripción de la opción no puede ser nula'
'ERROR: La votación no existe'
'ERROR: No se puede añadir una opción a una votación finalizada'
'ERROR: No se puede añadir una opción a una votación abierta'
'ERROR: (SQLERRM) ‘

4.5.3.7 Gestión de Opciones (DEL_OPCION)

DESCRIPCION: Da de baja una Opción de una votación, si cumple que:
- Todos los campos Obligatorios están informados.
- La votación exista
- La votación no esté abierta ni finalizada

PARAMETROS:
INPUT: Obligatorio: pidvota: NUMBER(08). identificador de votación.
pnopcio: NUMBER(08). Numero de opción.
OUTPUT: RSP VARCHAR2(300)
PRECONDICION: Exista la opción para esa votación
POSTCONDICION: Se da de baja la opción para la votación
RETORNA:
'OK' si la operación se ha realizado con éxito
'ERROR' Si:
'ERROR: El identificación de votación no puede ser nulo'
'ERROR: El identificación de opción no puede ser nulo'
'ERROR: La opción no existe para esta votación'
'ERROR: La votación no existe'
'ERROR: No se puede dar de baja una opción de una votación finalizada'
'ERROR: No se puede dar de baja una opción de una votación abierta'
'ERROR: (SQLERRM) '

4.5.3.8 Gestión de Opciones (UPD_OPCION)

DESCRIPCION: Modifica la descripción de una Opción que pertenece a una votación, si cumple que:
- Todos los campos Obligatorios están informados.
- La opción para a votación exista
- La votación no esté abierta ni finalizada

PARAMETROS:
INPUT: Obligatorio: pidvota: NUMBER(08). identificador de votación.
pnopcio: NUMBER(08). Numero de opción.
ptopcion: VARCHAR2(50). Descripción de la opción
OUTPUT: RSP VARCHAR2(300)
PRECONDICION: Exista la opción para la votación
POSTCONDICION: Se modifica la descripción de la opción para la votación
RETORNA:
'OK' si la operación se ha realizado con éxito
'ERROR' Si:
'ERROR: El identificación de votación no puede ser nulo'
'ERROR: El identificación de opción no puede ser nulo'
'ERROR: Descripción de la opción no puede ser nula'
'ERROR: La opción no existe para esta votación'

'ERROR: La votación no existe'
'ERROR: No se puede modificar una opción de una votación finalizada'
'ERROR: No se puede modificar una opción de una votación abierta'
'ERROR: (SQLERRM) '

4.5.3.9 Gestión de Censo - Votación (INS_VOTACENSO)

DESCRIPCION: Inserta el censo/s al cual va dirigida una votación, si cumple que:

- La votación esté informada y exista.
- La votación no esté abierta ni finalizada
- Si está informado el código de País que exista y esté vigente.
- Si está informada la región que exista y esté vigente.
- Si está informada la localidad que exista y esté vigente.

Si no se informa el código de país, indicará que es para todo el censo global, no se podrá informar ni la región, ni la localidad.
Si no se informa el identificador de región, indicará que es para todas las regiones que pertenezcan al país o censo global y no se podrá informar la localidad.
Si no se informa el identificador de localidad, indicará que es para todas las localidades de la región o país o censo global.

PARAMETROS:
INPUT: Obligatorio: pidvota: NUMBER(08): identificador de votación.
Opcionales: pcodpais VARCHAR2(2): Código del país
pidregion NUMBER (08): identificador de la región.
pidlocali NUMBER (08): identificador de la localidad.
OUTPUT: RSP VARCHAR2(300)
PRECONDICION: No exista el registro
POSTCONDICION: Se da de alta el registro
RETORNA:
'OK' si la operación se ha realizado con éxito
'ERROR' Si:
'ERROR: El Código de la votación no puede ser nulo'
'ERROR: El Identificador de la región no puede ser nulo'
'ERROR: El Código del país no puede ser nulo'
'ERROR: La votación no existe'
'ERROR: No se puede añadir un censo a una votación finalizada'
'ERROR: No se puede añadir un censo a una votación abierta'
'ERROR: La localidad no existe para la Región/País o no está vigente'
'ERROR: La Región no existe para el País o no está vigente'
'ERROR: El País no existe o no está vigente'
'ERROR: El censo para esta votación ya existe o esta contemplado';
'ERROR: El censo para esta votación ya esta contemplado'
'ERROR: El censo es más genérico que los existentes,
si desea uno genérico borre los anteriores'
'ERROR: (SQLERRM) '

4.5.3.10 Gestión de Censo - Votación (DEL_VOTACENSO)

DESCRIPCION: Da de baja un ceso de una votación, si previamente supera los siguientes controles:
- El campo idvotacenso/idvota este informado.
- El censo exista en la Base de datos
- La votación no esté abierta, ni finalizada

PARAMETROS:
INPUT: Obligatorio: pidvota: NUMBER(08): identificador de votación.
pidvotacenso NUMBER (08): identificador del censo del la votación.
OUTPUT: RSP VARCHAR2(200)
PRECONDICION: Exista el censo para esa votación
POSTCONDICION: Se da de baja el censo para la votación
RETORNA:
'OK' si la operación se ha realizado con éxito
'ERROR' Si:
'ERROR: El identificación de votación no puede ser nulo'
'ERROR: El identificación de votación de censo no puede ser nulo'
'ERROR: El censo no existe para esta votación'
'ERROR: La votación no existe'
'ERROR: No se puede dar de baja un censo de una votación finalizada'
'ERROR: No se puede dar de baja un censo de una votación abierta'
'ERROR: (SQLERRM)'

4.6 Módulo Estadístico

El módulo Estadístico es uno de los más importantes de la aplicación, por ese motivo dedico éste apartado.
Éste se alimenta, principalmente de unos disparadores que se activan cuando una votación pasa a estado finalizado o cuando una persona realiza una votación.
El estado se cambia a través del Packages PK_VOTACION en el procedimiento UPD_ESTADO_VOTA. Este procedimiento cuando se pasa a estado finalizado, actualiza los siguientes campos de las tablas, que nos servirá para este módulo:

- en tabla VOTACION:
  - Calcula y actualiza los campos NCAPVOT, NVOTTOT, PPARTI.
  - NCAPVOT: se actualizará con la suma de todas las personas que están en el censo de la votación.
  - NVOTTOT: se actualizará con todas las votaciones realizadas a esa votación.
  - PPARTI: se actualizará dividiendo (NVOTTOT / NCAPVOT) * 100
- en tabla OPCIONES:
  - Calcula y actualiza los campos NNUMVOT, PRESTOT.

- NNUMVOT: se actualizará con el total de votos que ha obtenido esa opción.
- PRESTOT: se actualizará con (NNUMVOT / VOTACION. NVOTTOT) * 100

- en tabla PCIUNOVOT_ANYO:
  - Actualiza la tabla en función de los cálculos obtenidos. Los cálculos que se realizan son los siguientes:
    - Total de Personas que por el censo al que pertenecen han podido votar y no lo han hecho. Es decir, personas que nunca han votado.
    - Total de Personas que por el censo al que pertenecen pueden votar.
    - El % = (Total Personas nunca han votado/ Total de Personas que pueden votar) * 100

A continuación se detalla en más profundidad los disparadores implementados:

4.6.1 DISPARADORES

4.6.1.1 TR_INS_VOTAPERSONA

Este disparador se activa después de cada inserción en la tabla VOTAPERSONA.
El propósito de este TRIGGER es tener un contador de votaciones por persona, campo numvotos de la tabla persona. También actualiza el histórico de votaciones que ha participado una persona, comparando el numero de votos que lleva la persona que vota, con los registro/s existente/s de la tabla HIS_VOTTOPPER, actualizándola si la persona que vota supera o iguala al registro/s existe/s.

4.6.1.2 TR_UPD_ESTADO_VOTACION

En disparador se activa después de actualizar la tabla VOTACION, cuando el campo cestado = 3, que significa que ha finalizado.
El propósito de este TRIGGER es lo siguiente:

- Votaciones finalizadas por PAIS/ANYO. Actualiza la tabla VOTFIN_PAISANYO. Contiene las votaciones finalizadas por cada país y año. Cada vez que se finaliza una votación se busca el censo estatal o censos estatales al cual va dirigido, sumando uno al campo nvotafin si existe el registro, si no existe el registro, genera uno nuevo con nvotafin=1.
- Votaciones finalizadas por CENSO/ANYO. Actualiza la tabla VMEDPART_CENSANYO. Que contiene las votaciones finalizadas
- Votaciones finalizadas con más participación por AÑO. Actualiza la tabla VOTMASPART_ANYO
- Votaciones finalizadas, con más votos emitidos por año/localidad. Actualiza la tabla VOTMASP_LOCANY.
- Votación que históricamente ha tenido más participación. Actualiza la tabla HIS_VOTMASPARTI.

- Votación que históricamente ha tenido más o menos diferencia porcentual entre la opción más y menos votada. Actualiza las tablas HIS_VMASDIFOPC y HIS_VMENDIFOPC

4.6.2 Consultas del módulo

1. Dado un país y año, mostrar el número de votaciones finalizadas que se han producido. Extrae los datos de la tabla VOTFIN_PAISANYO.
2. Dado un censo y año, mostrar el valor medio del % de participación, teniendo en cuenta todas las votaciones asociadas aquel censo electoral que ha finalizado ese año. Extrae los datos de la tabla VMEDPART_CENSANYO.
   Tipos de censo :
   0 - Global.
   1 - Estatal. Si no informa país saldrán todos
   2 - Regional. Si no informa el país ni la región saldrán todos
   3 - Local. Si no informa el país ni la región ni la localidad saldrán todas.

3. Dado un censo y año, mostrar el numero de votaciones que han finalizado ese año que ha tenido asociadas ese censo. Extrae los datos de la tabla VMEDPART_CENSANYO.
   Tipos de censo :
   0 - Global.
   1 - Estatal. Si no informa país saldrán todos
   2 - Regional. Si no informa el país ni la región saldrán todos
   3 - Local. Si no informa el país ni la región ni la localidad saldrán todas.

4. Dado un año, muestra la votación que más participación ha tenido. Extrae los datos de la tabla VOTMASPART_ANYO.
5. Dado un año, mostrar la localidad Europea que más votos han emitido sus ciudadanos. Extrae los datos de la tabla VOTMASP_LOCANY.
6. Dado un año, mostrar el % de ciudadanos que nunca han votado y han estado convocados. Extrae los datos de la tabla PCIUNOVOT_ANYO.
7. Muestra la Votación/es que históricamente han tenido más participación. Extrae los datos de la tabla HIS_VOTMASPARTI.
8. Muestra la Votación/es que históricamente ha tenido MÁS diferencia porcentual entre opciones. Extrae los datos de la tabla HIS_VMASDIFOPC.
9. Muestra la Votación/es que históricamente ha tenido MENOS diferencia porcentual entre opciones. Extrae los datos de la tabla HIS_VMENDIFOPC.
10. Muestra el numero máximo de votaciones en las que ha participado una persona. Extrae los datos de HIS_VOTTOPPER.

5. Plan de Contingencias

Existen diversos riesgos o factores que pueden afectar, de forma imprevista, el desarrollo del proyecto, y no tener tiempo de reacción para cumplir las fechas acordadas. Para prevenir estos contratiempos redacto el siguiente plan de contingencias, con los motivos más usuales que pueden ocurrir en este tipo de proyecto:

Motivos Técnicos:

Descripción	Solución
- Perdida de datos - Error del sistema - Problemas de Hardware. - Errores en la instalación del SGBD	- Copia de seguridad diaria. - Ordenador de Reserva, idéntico a nivel de parámetros y software que se utiliza para el desarrollo.

Motivos Personales:

Descripción	Solución
- Compromisos - Enfermedades	- En la planificación no se ha contado con los días festivos y fines de semana. Por lo que se podría contar con ellos si hubieran contratiempos.

Motivos Laborales:

Descripción	Solución
- Ampliar el horario de la vida Laboral. - Traslado a otra ciudad.	- En la planificación no se ha contado con los días festivos y fines de semana. Por lo que se podría contar con ellos si hubieran contratiempos. - Usar un portátil para seguir con la práctica, en otra ciudad

6. Plan de Pruebas

Para probar el correcto funcionamiento del sistema, se han desarrollado los siguientes scripts:

- Una carga Inicial de datos.
- Pruebas de Procedimientos
- Pruebas de Consultas y Estadísticas

Los cuales contiene una serie de sentencias SQL mediante las cuales se podrá corroborar que las funcionalidades proporcionadas se ajustan a los requerimientos.

6.1 Carga Inicial de datos

La carga inicial de datos, se ha extraído de las pruebas unitarias realizadas en el desarrollo. Proporciono muchos datos para obtener unos datos estadísticos más próximos a la realidad.

Dicha carga de inicialización está dentro de la carpeta que se adjunta, llamada SCRIPTS en una subcarpeta llamada INI_DATOS, esta contiene:

- 10_DAT_PAISES.sql
- 20_DAT_REGION.sql
- 30_DAT_LOCALIDAD.sql
- 40_DAT_PERSONAS.sql:
- 41_DAT_PERSOCENSO.sql
- 50_DAT_VOTACION.sql
- 51_DAT_OPCIONES.sql
- 52_DAT_VOTACENSO.sql
- 60_DAT_VOTAPERSONA.sql

Cada uno de los scripts numerados, se tipifican de la siguiente manera NN_DAT_nombre de la tabla, realizan el llenado de las tablas que su nombre indican.

Estas carga se realizará al ejecutar el script LANZAR. Durante la ejecución se mostrará los progresos realizados en cada ejecución de los scripts detallados anteriormente.

6.2 Pruebas de Procedimientos

Una vez realizada la carga inicial, se procederá a realizar las pruebas de los procedimientos implementados, a través del script 10-Pruebas_Proc_Pk.sql. Se probarán Altas, bajas y modificaciones de datos, comprobando que se cumplan las restricciones especificadas por los requerimientos a la hora de realizar las funcionalidades exigidas.
Hay que resaltar que dicha prueba diseñada no comprueba exhaustivamente todas y cada una de las casuísticas del sistema, es decir, todos los casos de errores posibles. Se ha tratado de hacer una prueba representativa, provocando algunos errores controlados y así mostrar el comportamiento de los mismos. Así mantenemos una prueba sencilla, breve, y que facilite la comprensión de la misma.

6.3 Pruebas de Consultas y Consultas Estadísticas

Se lanza a través de el script 11-Pruebas_consultas. En este script se contemplan los procedimientos de consultas y las consultas del módulo Estadístico.

6.3.1 Pruebas de Consultas

Se proporciona la ejecución de todos los procedimientos de consulta, donde se listarán los datos correspondientes a cada consulta detallada en el apartado de implementación. Verificando su perfecto funcionamiento.

6.3.2 Pruebas Módulo Estadístico

Se proporciona la ejecución de las consultas del módulo estadístico, estas consultas están descritas en el apartado de implementación. Solo cabe destacar que si no sale información en alguna tabla es porque no existen datos de la misma.

6.4 Resultado de las pruebas

Tras la ejecución del script de pruebas, los resultados que más nos interesan son los que se muestran en las tablas de logs y de estadísticas. El resultado esperado tras las pruebas es el siguiente:

En la tabla de Logs, habrá una entrada por cada llamada a un procedimiento almacenado, con los parámetros correspondientes de entrada y de salida. Todas las llamadas tendrán como parámetro de salida 'OK', indicando que las operaciones se han llevado a cabo correctamente o bien un mensaje 'ERROR' + explicación del error, en los casos que se ha producido un error controlado.

El resultado de la ejecución de las pruebas se puede ver en los siguientes .log:

- Pruebas_Proc_Pk.log. Salida de la información referente a todos los procedimientos probados.
- Pruebas_consultas.log. Salida de las consultas de los procedimientos de consulta y el modulo de estadística.

7. Valoración Económica y Recursos

En este punto ya estamos en disposición de elaborar el presupuesto y los recursos necesarios para llevar a cabo proyecto de votaciones. Que a continuación se detalla.

7.1 Valoración Económica

Tras haber detectado los perfiles necesarios, las tareas a desarrollar y programado la planificación estamos en disposición de elaborar la valoración económica. Contando que las jornadas son de lunes a viernes, excepto días festivos. Cada jornada se ha estimado de 4 horas.
Para calcular los costes de personal, disponemos de las siguientes tarifas (con iva incluido) precio/mercado. Costes por hora/jornada trabajada:

Tarifas de Precios		
Recurso	**Coste/Hora**	**Coste/Jornada**
Jefe de Proyecto	40€	160€
Administrador de Bases de Datos	35€	140€
Analista	26€	104€
Programador	20€	80€

En el siguiente cuadro se muestran por cada perfil las tareas, las jornadas previstas y su coste para desarrollarlas y el coste parcial de cada Actividad principal.

Código Actividad	Nombre Actividad	Estimación (Jornadas)	Recurso	Coste	Coste Parcial
01.	**Inicio del proyecto**	**01**	**Jefe Proyecto**	**160€**	**160€**
02.	**Recogida y Lectura de la documentación**	**01**	**Jefe Proyecto**	**160€**	**160€**
		01	**Analista**	**104€**	**104€**
03.	**Instalación Gantt-Project**	**01**	**Analista**	**104€**	**104€**
04.	**Planificación del Proyecto (PAC1)**				**1.280€**
04.01.	Descripción Justificación TFC	01	Jefe Proyecto	160€	
04.02.	Definición de Objetivos	01	Jefe Proyecto	160€	
04.03.	Identificación de Tareas	01	Jefe Proyecto	160€	
04.04.	Planificar Proyecto	01	Jefe Proyecto	160€	
04.05.	Valoración Económica y Recursos necesarios	02	Jefe Proyecto	320€	
04.06.	Elaboración Documentación a Entregar	02	Jefe Proyecto	320€	
05.	**Análisis y Diseño**				**2.428€**
05.01.	Análisis				
05.01.01.	Requisitos Funcionales	02	Analista	208€	
05.01.02.	Requisitos no Funcionales	02	Analista	208€	
05.02.	Diseño B.D.				
05.02.01.	Conceptual	05	Analista	520€	
05.02.02.	Lógico	05	Analista	520€	
05.02.03.	Físico	05	Analista	520€	
05.03.	Instalación de Oracle	01	Adminis. B.D.	140€	
05.04.	Elaboración Documentación a Entregar	03	Analista	312€	
06.	**Implementación y Pruebas**				**1.752€**
06.01.	Scripts de creación B.D., Tablas, Usuario y datos mínimos.	02	Programador	160€	
06.02.	Desarrollo de Scripts				
06.02.01.	Gestión de Censos	02	Programador	160€	
06.02.02.	Gestión de Votaciones	02	Programador	160€	
06.02.03.	Gestión de Personas	02	Programador	160€	
06.02.04.	Gestión Estadística	02	Programador	160€	
06.02.05.	Gestión de Consultas	02	Programador	160€	
06.03.	Pruebas Unitarias				
06.03.01.	Juego de pruebas	02	Programador	160€	
06.03.02.	Script de pruebas	02	Programador	160€	
06.03.03.	Depuración código	02	Programador	160€	
06.04.	Elaboración Documentación a Entregar	03	Analista	312€	
07.	**Producto + Memoria + Presentación**				**2.480€**
07.01.	Pruebas Integrales y Depuración de errores	02 03	Analista Programador	208€ 240€	
07.02.	Memoria	01	Jefe Proyecto	160€	
		07	Analista	728€	
07.03.	Presentación	05	Jefe Proyecto	800€	
07.04.	Repaso de la documentación a entregar	01 01 01	Jefe Proyecto Analista Programador	160€ 104€ 80€	
TOTAL		**77**		**8.468€**	**8.468€**

El coste del desarrollo es el siguiente:

Perfil	Jornadas	Coste Jornada	€
Analista	37	104 €	3.848
Programador	22	80 €	1.760
Jefe de Proyecto	17	160 €	2.720
Administrador Base de Datos	01	140 €	140
		Total	**8.468**

7.2 Recursos Necesarios

Para llevar a cabo el proyecto, en principio, se prevén los siguientes recursos:

Nivel de Hardware

Solo necesitamos un equipo para desarrollar el proyecto, ya que solo existe un recurso, pero como se ha comentado en el apartado "plan de contingencia", se usará otro equipo para prevenir contratiempos de tipo técnico.
Las características de los PC son:

- El de desarrollo: Un Ordenador portátil ACCER Aspire 5735: Intel Core Duo a 2 GHz, 4 Gb de memoria RAM con SO Windows Vista Home Premium.
- El de sustitución: Un Ordenador portátil HP: Intel Core a 2.26 GHz, 4 Gb de memoria RAM con SO Windows Vista Home Premium.

Nivel de Software

Se utilizarán en este proyecto, el siguiente software:

- Gantt – Project 2.0.6. Planificación y seguimiento del proyecto
- Microsoft Word 2000, Microsoft Power Point 2000. Elaboración de la documentación.
- DIA 0.97.1. Diagramas E/R.
- Oracle Express v10.2.0.1 Sistema de Gestión de Bases de Datos Relacionales (SGBDR).
- UltraEdit-32: Desarrollo PL/SQL.

8. Bibliografía

- Practica que elaboré en la asignatura de GOPI.
- Material de la asignatura SGBD
- Material de la asignatura Ingeniería del Software

9. Comentarios

9.1 Revisión de la PAC2 con respecto a la PAC1

En la planificación Gantt, no quedaba muy claro cuales serían los entregables de la PAC3, se han añadido etiquetas como en la descripción de la PAC2.

Se ha añadido un nuevo apartado que se explican nuevas funcionalidades del proyecto.

9.2 Revisión de la PAC3 con respecto a la PAC2

He cambiado la plantilla del documento, salía un logo y el estilo no era homogéneo.

Había planificado generar tablespaces para datos e índices. Al final he decidido no generarlo, ya que se tendrá que gestionar toda la base de datos, en cuanto a permisos, usuarios, etc. Para este tema, quién mejor que la persona que gestionará la base de datos, podrá decidir como y donde los quiere.

A nivel de diseño, había dispuesto generar un packages para las consultas estadísticas, cuando e realizado el desarrollo me he dado cuenta de que no era necesario este packages, ya que se soluciona con simples consultas (SELECT).

10. Conclusiones

En primer lugar quiero destacar, que el trabajo desarrollado en este proyecto ha sido muy enriquecedor, ya que te permite desarrollar distintos roles dentro del mismo proyecto como el de Jefe de Proyecto, Analista, programador y Gestor de base de datos. Además de vivir todas las fases de un proyecto en primera persona.

Para realizar una buena planificación, es importante medir la disponibilidad y las tareas a desarrollar, contando con imprevistos posibles que puedan surgir. Esto nos ayudará a cumplir con los plazos establecidos, sin agobios.

Otra conclusión será realizar un análisis de los requerimientos y tener claro los que se tiene que hacer y como se tiene que hacer, lo antes posible. Aplicando las herramientas y metodología de las que disponemos, para realizar un buen analisis y diseño para llegar a nuestro objetivo.

La documentación es un instrumento que nos permite plasmar el trabajo que vamos a entregar, por tanto tendrá que ser de una calidad elevada. Así como las entregas parciales son primordiales, ya que de este modo en una siguiente fase puedes corregir errores de análisis o diseño que el cliente va detectando.

Otro aspecto a tener en cuenta, es implementar procesos sencillos sin complejidad, la sencillez ayuda a una mejor compresión del código, a la realización de pruebas y de la busqueda de posibles errorres.

La realización de un buen juego de pruebas nos ayudará a depurar el codigo desarrollado. Es importante realizar un juego de pruebas unitario, en cada desarrollo que se realice, así se van modificando el código de una manera sencilla. Posteriormente, cuando el código unitario esté depurado, se realiza una prueba integral de toda la aplicación y así comprobar que los requerimientos se contemplan en nuestra aplicación.