

Gestión Postventa

Estudiante: Francisco Amador Rísquez Teva
Enginyeria en Informàtica

Consultor: Josep Vallverdú Vergé

20 de Marzo de 2011



DEDICATORIA Y AGRADECIMIENTOS

A mis padres y hermana por haberme apoyado e inculcado el estudio durante toda mi vida.

A mi fiel amigo, mi fiel compañero, mi GRAN perro Zas. Cuantas horas has pasado tumbado a mis pies apoyándome en los momentos en que el estudio sólo era una vía para ausentarme de la realidad. Zas, no te preocupes, el vacío que me has dejado lo llenaré con el recuerdo de los momentos felices. Adiós y hasta siempre.

A Mónica, la persona que me ha hecho ver las estrellas cuando mis lágrimas me impedían ver el sol. La que en un corto espacio de tiempo será la madre de nuestro primer hijo. Gracias mi vida.

A Hugo, que llegará en Agosto y al que esperamos con los brazos abiertos.

A todos ellos GRACIAS.

“Sólo cerrando las puertas detrás de uno se abren las ventanas hacia el porvenir.”

Françoise Sagan

1 Resumen, palabras clave y área del PFC

1.1 Resumen

Nuestra empresa, especializada en el segmento de los electrodomésticos, se ve en la necesidad de cambiar el sistema que tiene para gestionar la post venta de los productos.

Nuestro equipo será el encargado de analizar los requerimientos del nuevo sistema que hagan referencia a la información que hace falta mantener, así como implementar la base de datos que satisfaga estos requerimientos junto con los procedimientos necesarios para la gestión que proporcionaran el único método de acceso.

A parte de la gestión diaria de las incidencias, se solicita la consolidación de datos para generar estadísticas que permitan mejorar la eficiencia de la empresa.

Así, los objetivos serán:

- Detectar cuales son las necesidades básicas del sistema.
- Detectar posibles funcionalidades adicionales de valor añadido.
- Proponer un diseño que se ajuste a los requerimientos expuestos.
- Implementar un sistema que encapsule las funciones de acceso a los datos.

1.2 Palabras Clave

Ingeniería Informática, Base de datos, Almacén de Datos, Data Warehouse, Post venta, Electrodomésticos, Servicios, Técnicos, Teleoperadores.

1.3 Área del TFC

El área del presente PFC es 11.013 PFC-Bases de dades correspondiente a la Ingeniería Informática (EI) de la Universitat Oberta de Catalunya (UOC).

2 Índice

2.1 Índice de Contenidos

1	Resumen, palabras clave y área del PFC.....	1
1.1	<i>Resumen</i>	1
1.2	<i>Palabras Clave</i>	1
1.3	<i>Área del TFC</i>	1
2	Índice.....	2
2.1	<i>Índice de Contenidos</i>	2
2.2	<i>Índice de Figuras</i>	3
3	Introducción	4
3.1	<i>Justificación y contexto del PFC</i>	4
3.2	<i>Objetivo</i>	4
3.3	<i>Alcance del Proyecto</i>	5
3.4	<i>Entregas</i>	6
3.5	<i>Recursos</i>	6
3.5.1	Hardware	6
3.5.2	Software	6
3.6	<i>Enfoque y método seguido</i>	6
4	Planificación PFC.....	7
4.1	<i>Fechas Clave</i>	7
4.2	<i>Fases del proyecto</i>	8
4.2.1	Planificación.....	8
4.2.2	Análisis y diseño	8
4.2.3	Implementación.....	8
4.2.4	Desarrollo de procedimientos almacenados PL/SQL	9
4.2.5	Pruebas	9
4.2.6	Entrega Final	9
4.3	<i>Desglose de actividades</i>	10
4.4	<i>Diagrama de Gantt</i>	11
4.5	<i>Incidencias y riesgos</i>	11
4.6	<i>Productos obtenidos</i>	12
4.7	<i>Breve descripción del resto de capítulos de la memoria</i>	12
5	Análisis	13
5.1	<i>Requerimientos</i>	13
5.1.1	Requerimientos No Funcionales.....	14
5.1.2	Requerimientos Funcionales	17
6	Diseño Base de Datos	65
6.1	<i>Diseño Conceptual</i>	65
6.1.1	Diagrama Entidad-Relación.....	65
6.1.2	Entidades.....	65
6.1.3	Atributos	67
6.2	<i>Diseño Lógico</i>	68
6.2.1	Modelo Lógico.....	68
6.2.2	Entidades.....	69
7	Diseño Almacén de Datos (DW).....	71
7.1	<i>Diseño Conceptual</i>	71
7.1.1	Estadística de Peticiones	71
7.1.2	Estadística de Teleoperadores	71
7.1.3	Estadística de Técnicos	72
7.1.4	Entidades.....	72
7.1.5	Atributos	73
7.2	<i>Diseño Lógico</i>	74
7.2.1	Modelo Lógico.....	74

7.2.2	Entidades.....	74
8	PRODUCTO FINAL	75
8.1	Instalación del producto	76
8.2	Plan de pruebas.....	76
8.2.1	Subsistema Configuración.....	76
8.2.2	Subsistema Personal	79
8.2.3	Subsistema Petición.....	81
8.2.4	Subsistema Atención.....	82
8.2.5	Consultas.....	83
8.3	Carga de Datos.....	84
8.4	ETL.....	85
8.5	Estadísticas	85
9	Valoración Económica	85
10	CONCLUSIONES.....	86
11	GLOSARIO	87
12	BIBLIOGRAFIA	88

2.2 Índice de Figuras

Figura 1:	Ciclo de vida y obtención de documentos.	7
Figura 2:	Desglose de actividades	10
Figura 3:	Diagrama de Gantt.....	11
Figura 4:	Arquitectura lógica de un almacén de datos.	14
Figura 5:	Diagrama de casos de uso del sistema completo.....	17
Figura 6:	Casos de uso del Subsistema Configuración: Servicios.	19
Figura 7:	Casos de uso del Subsistema Configuración: Electrodomésticos.....	22
Figura 8:	Casos de uso del Subsistema Configuración: Distribución Geográfica (Provincias).	25
Figura 9:	Casos de uso del Subsistema Configuración: Distribución Geográfica (Poblaciones).	27
Figura 10:	Casos de uso del Subsistema Configuración: Distribución Geográfica (Zonas).	30
Figura 11:	Casos de uso del Subsistema Personal: Teleoperadores.	35
Figura 12:	Casos de uso del Subsistema Personal: Técnicos.....	38
Figura 13:	Casos de uso del Subsistema Petición: Clientes.	47
Figura 14:	Casos de uso del Subsistema Petición: Productos.	49
Figura 15:	Casos de uso del Subsistema Petición: Peticiones de Servicio.	51
Figura 16:	Casos de uso del Subsistema Atención: Búsquedas y Asignaciones.	53
Figura 17:	Casos de uso del Subsistema Atención: Trabajos.	56
Figura 18:	Casos de uso del Subsistema Atención: Encuesta.	58
Figura 19:	Casos de uso del Subsistema Histórico: Históricos.	60
Figura 20:	Casos de uso del Subsistema Control: Log.	62
Figura 21:	Diagrama Entidad-Relación.	65
Figura 22:	Modelo lógico Base de Datos.	68
Figura 23:	Estrella Estadística Peticiones.	71
Figura 24:	Estrella Estadística Teleoperadores.	72
Figura 25:	Estrella Estadística Técnicos.	72
Figura 26:	Modelo lógico del Almacén de Datos.	74
Figura 27:	Valoración Económica.	86

3 Introducción

3.1 Justificación y contexto del PFC

Según la Wikipedia, una base de datos (BD) es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Hoy en día, casi toda la totalidad de este almacenamiento se realiza en formato digital.

Aparecen los Sistemas de Gestión de Bases de Datos (SGBD), aplicaciones que permiten el almacenamiento y recuperación de los datos de forma rápida y estructurada.

Así en la actualidad, toda aplicación informática que se realice y que necesite una persistencia de datos en el tiempo, entre distintas ejecuciones, utilizarán de forma inevitable las BD y los SGBD para su gestión, programación y administración.

Para el desarrollo del presente Proyecto Final de Carrera (PFC) utilizaremos Oracle. Oracle es uno de los SGBD más completos y más utilizados en los servidores empresariales y, aunque últimamente sufre la competencia de otros SGBD que avanzan de forma rápida en el mercado, sigue siendo un referente en su campo.

En el área de este PFC (Base de datos) vamos a trabajar aspectos relacionados con el diseño y la implementación del sistema planteado en el enunciado del proyecto.

Se pretende que durante la realización del presente proyecto se apliquen los métodos y técnicas que han sido estudiadas a la largo de los estudios de Ingeniería Informática (EI).

En el transcurso del presente proyecto realizaremos el análisis y diseño de la base de datos, los scripts necesarios para crear la base de datos e implementar los procedimientos almacenados necesarios para el correcto funcionamiento del sistema.

3.2 Objetivo

El objetivo de este PFC de Bases de Datos Relacionales pretende:

- poner en práctica y consolidar conocimientos adquiridos en las asignaturas de:
 - Gestión de Proyectos
 - Bases de Datos I
 - Bases de Datos II
 - Sistemas de Gestión de Bases de Datos
 - Modelos multidimensionales y almacenes de datos.
- utilizar el lenguaje PL/SQL y SQL Dinámico.
- practicar con un sistema de gestión de bases de datos de gran impacto en el mercado (Oracle).

En nuestro caso el PFC consiste en diseñar e implementar una BD para la gestión de las incidencias post venta de los productos vendidos por nuestra empresa.

Los principales requisitos que ha de satisfacer la BD son:

- Proporcionar un sistema de gestión y acceso a datos exclusivamente a través de procedimientos almacenados.
- Gestión de incidencias postventa. Manteniendo el estado actual, así como un historial de las acciones realizadas.
- Almacenamiento de los datos necesarios para la correcta gestión (productos, agentes telefónicos, ingenieros de soporte...)
- El sistema debe ser escalable para poder ir incorporando de forma progresiva aquellas necesidades que surjan durante su vida.
- Consolidar datos para poder generar estadísticas útiles para mejorar la eficiencia de la empresa (número de incidencias por tipo de producto, tipos de incidencia, coste de las garantías, etc.). Para ello será necesaria la definición de un almacén de datos (Data Warehouse DW) con los consiguientes procesos de traspaso de información (ETL) y las consultas necesarias para la obtención de la información.
- Dispondrá de mecanismos para permitir resolver posibles problemas de integración con el resto de sistemas:
 - Log de acciones realizadas en la base de datos
 - Test de funcionalidad de la BD.

3.3 Alcance del Proyecto

Para conseguir los objetivos descritos, se realizarán las siguientes tareas:

- Análisis de requerimientos
- Base de Datos
 - Diseño conceptual. Modelo Entidad-Relación
 - Diseño Lógico
 - Diseño Físico. Realización de Scripts para la creación de toda la estructura de la base de datos
 - Implementación de los Procedimientos Almacenados (SP) para la gestión, acceso y control de errores de los datos
- Data Warehouse (DW)
 - Diseño Conceptual Multidimensional
 - Diseño Lógico Multidimensional
 - Diseño Físico Multidimensional
 - Implementación de scripts para extraer, transformar y cargar datos (ETL Extract, Transform and Load)
- Documentación del proyecto

3.4 Entregas

El proyecto constará de los siguientes elementos:

- Plan de Trabajo
- Producto Final que incluirá la BD, así como los scripts SQL para la creación de estructuras y procedimientos almacenados (SP).
- Memoria
- Presentación virtual

3.5 Recursos

3.5.1 Hardware

Para la realización del proyecto se utilizará:

- Ordenador portátil LG R310
Intel Core2 Duo CPU P7350 2Ghz
RAM 4 GB
Sistema Operativo: Windows Vista Home Premium 32bits

3.5.2 Software

Para la realización del proyecto se utilizarán las siguientes aplicaciones:

- SGBD
 - Oracle Database 10g Express Edition
- Desarrollo SQL y Administración BD
 - Toad for Oracle 10
- Diagramas
 - Magic Draw 15
 - Toad Data Modeler 3.4
- Administrador de Proyectos
 - Microsoft Project Professional 2003
- Procesador de Textos
 - Microsoft Office Word 2007

3.6 Enfoque y método seguido

Tras un análisis inicial de la descripción y requerimientos del proyecto se ha elaborado una planificación para poder hacer frente en el periodo de tiempo establecido.

El proyecto se realizará siguiendo el ciclo de vida en Cascada, Iterativo e Incremental por su eficacia y simplicidad. Así, nuestro proyecto estará formado por distintos mini proyectos (iteraciones) que en conjunto formarán el producto final. En cada iteración y a partir de los

resultados anteriores, el producto sufrirá una evolución (incremental) añadiendo o mejorando los objetivos/requisitos anteriores.

En la siguiente figura podemos ver las etapas previstas con los documentos obtenidos y que serán la base de partida para la siguiente etapa:

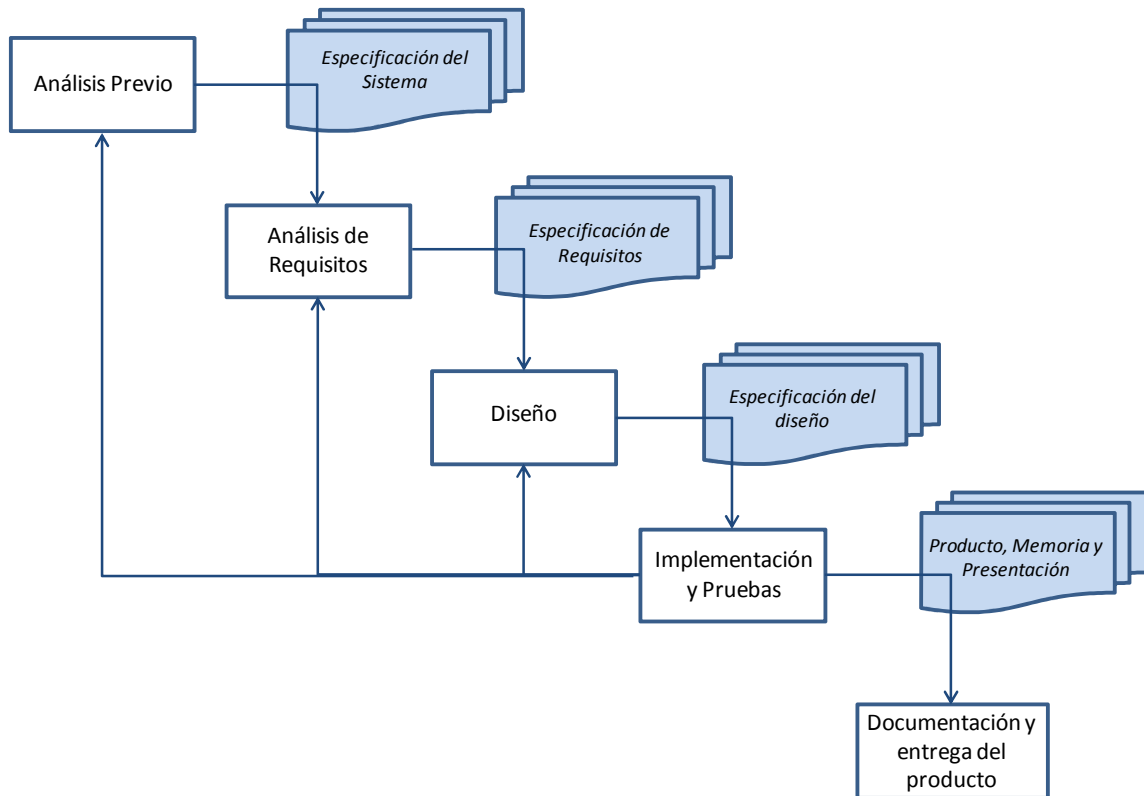


Figura 1: Ciclo de vida y obtención de documentos.

4 Planificación PFC

4.1 Fechas Clave

La fecha de entrega del PFC se fija el día 12 de junio de 2011 y, hasta esta fecha, se realizarán tres entregas que corresponderán con las fechas de cada una de las PAC fijadas en el Plan Docente. La temporización es la siguiente:

20/03/2011: PAC1 - Plan de trabajo.

17/04/2011: PAC2

15/05/2011: PAC3

12/06/2011: ENTREGA FINAL – Memoria, producto y presentación virtual.

4.2 Fases del proyecto

El proyecto constará de las siguientes fases:

- Planificación
- Análisis y diseño
- Implementación
- Desarrollo de procedimientos almacenados PL/SQL
- Pruebas
- Documentación y entrega final

Veamos con más de detalle cada una de las siguientes etapas.

4.2.1 Planificación

En el transcurso de esta fase se realizará una descomposición del trabajo a realizar en tareas, así como su posterior planificación en el tiempo. Se valorarán las posibles incidencias y riesgos elaborando un plan de contingencias para cada caso.

Entrega PAC1

4.2.2 Análisis y diseño

Esta fase constará de:

- Análisis de requisitos. Se analizarán los requerimientos proporcionados, evaluando si éstos son suficientes y examinando que no den lugar a posibles controversias.
- Diseño de BD y DW.
 - Diseño Conceptual mediante diagrama Entidad/Relación (E/R).
 - Diseño Lógico.
 - Diseño Físico.

4.2.3 Implementación

Durante esta fase se realizarán los scripts de creación de tablas e índices y posibles disparadores (triggers), tanto de la base de datos como del almacén de datos.

Se llevaran a cabo pruebas unitarias del funcionamiento.

Entrega PAC2.

4.2.4 Desarrollo de procedimientos almacenados PL/SQL

En esta fase se desarrollarán los procedimientos almacenados que dotarán de funcionalidad a nuestra BD. Destacaremos:

- Procedimientos almacenados para la gestión de la aplicación Gestión Postventa
- Procedimientos de control y registro de errores
- Procedimientos ETL (Transformación, Depuración y Carga)
- Obtención de estadísticas del DW

Se llevaran a cabo pruebas unitarias del funcionamiento.

Durante el transcurso de esta fase puede ser necesaria la modificación de la fase anterior (realimentación correctiva).

Entrega PAC3.

4.2.5 Pruebas

Una vez finalizado el producto se llevarán a cabo pruebas de integración con el fin de verificar la obtención de los datos esperados.

Durante el transcurso de esta fase puede ser necesaria la modificación de la fase anterior (realimentación correctiva).

4.2.6 Entrega Final

Durante esta fase se completará la memoria, así como la creación de una presentación virtual del PFC.

- Memoria. Será el proyecto en sí y deberá quedar reflejado todo el trabajo que se ha llevado a cabo. Aunque su entrega se realiza al final, su desarrollo se irá realizando en el transcurso de todo el proyecto.
- Presentación. Será un resumen del trabajo realizado. Deberá ser clara, concisa y bien estructurada para facilitar el seguimiento por parte de los miembros del tribunal.

4.3 Desglose de actividades

El proyecto quedará desglosado en las diferentes actividades:

CÓDIGO	ACTIVIDADES	PREDECESORAS
01	PAC1. PLAN DE TRABAJO	
01.01	Lectura del PFC	
01.02	Planificación de tareas	01.01
01.03	Creación del documento de entrega	01.02
01.04	Entrega PAC1	01.03
02	Instalación Oracle 10g Express Edition	
03	PAC2. ANÁLISIS Y DISEÑO	01.04
03.01	Análisis de Requerimientos	
03.02	Diseño BD	03.01
03.02.01	Diseño Conceptual	03.01
03.02.02	Diseño Lógico	03.02.01
03.02.03	Diseño Físico	03.02.02
03.03	Implementación BD	02; 03.02
03.03.01	Creación Scripts	03.02.02
03.04	Diseño DW	03.03
03.04.01	Diseño Conceptual	03.03.01
03.04.02	Diseño Lógico	03.04.01
03.04.03	Diseño Físico	03.04.02
03.05	Implementación DW	03.04
03.05.01	Creación Scripts	03.04.03
03.06	Pruebas	03.03.01; 03.05.01
03.07	Documentos entregables	
03.08	Entrega PAC2	03.07
04	PAC3. DESARROLLO DE PROCEDIMIENTOS ALMACENADOS	03.06
04.01	Gestión Base de Datos	
04.01.01	Creación Scripts Subsistema Control	
04.01.02	Pruebas Subsistema Control	04.01.01
04.01.03	Creación Scripts Subsistema Configuración	04.01.02
04.01.04	Pruebas Subsistema Configuración	04.01.03
04.01.05	Creación Scripts Subsistema Personal	04.01.04
04.01.06	Pruebas Subsistema Personal	04.01.05
04.01.07	Creación Subsistema Petición	04.01.06
04.01.08	Pruebas Subsistema Petición	04.01.07
04.01.09	Creación Subsistema Atención	04.01.08
04.01.10	Pruebas Subsistema Atención	04.01.09
04.01.11	Creación Subsistema Historial	04.01.10
04.01.12	Pruebas Subsistema Historial	04.01.11
04.02	ETL	04.01
04.02.01	Creación Scripts Procedimientos	04.02
04.02.02	Pruebas ETL	04.03.01
04.03	Documentos entregables	
04.04	Entrega PAC3	04.05
05	PRUEBAS SISTEMA COMPLETO	04
06	ESTADÍSTICAS DW	05
06.01	Instalación Cognos	
06.02	Generación Estadísticas	06.01
07	ENTREGA FINAL	
07.01	Memoria	
07.02	Presentación Virtual	06.01
07.03	Producto	06.02
07.04	Entrega Final	06.03

Figura 2: Desglose de actividades

4.4 Diagrama de Gantt

La planificación propuesta inicialmente la podemos ver en el siguiente diagrama de Gantt:

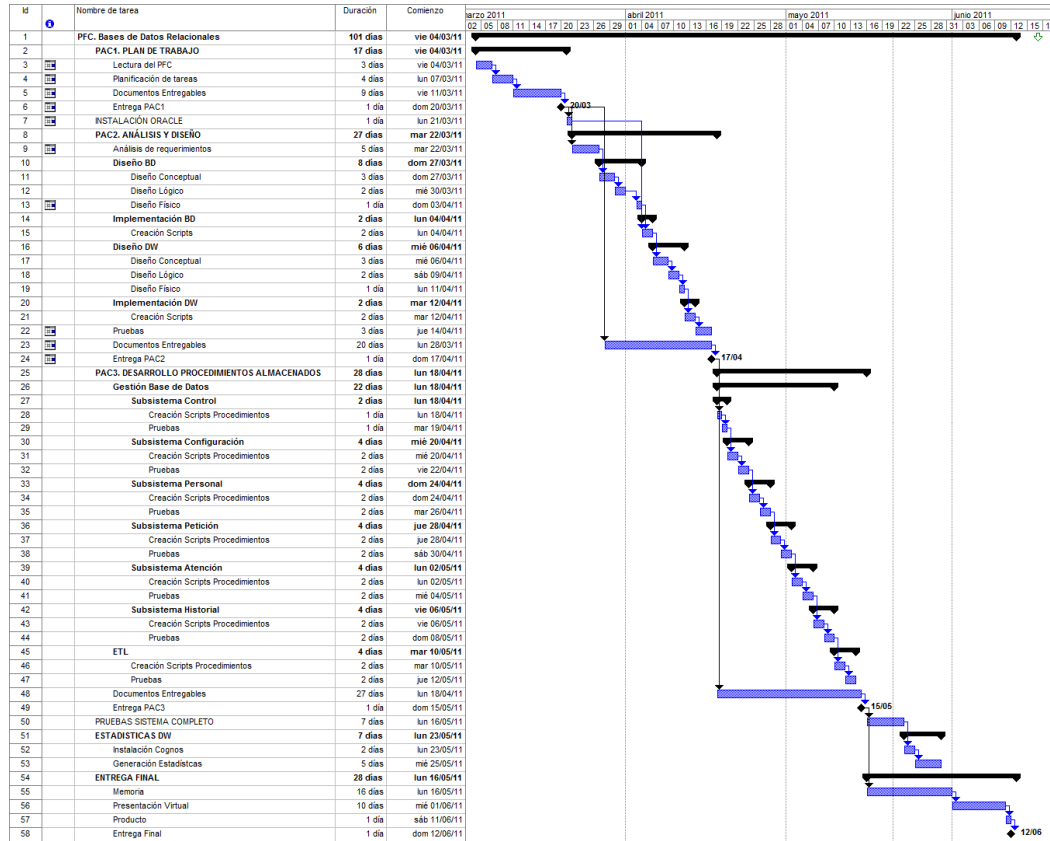


Figura 3: Diagrama de Gantt.

Se ha fijado como día de inicio del proyecto el 02/03/2011 y como fecha de finalización el día 12/06/2011, un total de 101 días para su desarrollo.

Todas las tareas se inician una vez acabada la tarea anterior. Es posible que llegado un punto se tengan que realizar modificaciones en tareas anteriores debido a las realimentaciones que hemos comentado anteriormente.

Esta planificación se ha llevado a cabo contando todos los días de la semana como laborables y sin tener en cuenta las fiestas de Semana Santa, que dedicaré al estudio igual que cualquier otro fin de semana.

4.5 Incidencias y riesgos

No se han valorado incidencias ni riesgos iniciales. Los tiempos dedicados a cada tarea son, en principio, suficientemente amplios como para poder acometerlos sin problemas.

En caso de algún retraso se podría reducir el tiempo dedicado a las pruebas, tanto unitarias como de todo el sistema. Este hecho podría perjudicar en la calidad del producto final pero, a cambio, el proyecto se entregaría en la fecha establecida.

También hay un amplio margen, tanto en la finalización de la memoria como en la creación de la presentación virtual, que se podría utilizar en otras tareas en caso de extrema necesidad.

Serán tratados con el consultor aquellos casos que puedan provocar un retraso en cualquiera de las entregas, o bien que no se puedan realizar las actividades marcadas en la planificación.

4.6 Productos obtenidos

El día fijado para la finalización del PFC se entregarán los siguientes documentos que forman el:

- Plan de trabajo: documento con la definición del PFC, las diferentes tareas y su distribución en el tiempo. Incluido en la primera parte de la memoria.
- Memoria: incluirá una descripción detallada y justificación de las soluciones técnicas adoptadas.
- Presentación virtual: con un resumen esquematizado de todo el trabajo realizado para obtener el producto final.
- Producto: Scripts encargados de la creación de la base de datos (tablas, índices, triggers, procedimientos almacenados, funciones, etc.)
- Plan de pruebas: consistirá en un anexo en el que se detallarán las diferentes pruebas a las que se ha sometido al producto para probar su robustez y funcionalidad. Junto al script de pruebas se presenta un script de carga inicial de estaciones (33 estaciones) y un procedimiento de carga de lecturas configurable (configurando intervalo de tiempo y lecturas por estación).

4.7 Breve descripción del resto de capítulos de la memoria

A continuación detallamos la estructura del resto de la memoria.

- **Diseño de la base de datos:** Realizaremos tanto el diseño conceptual (diagrama E/R) y lógico (tablas, índices, constraints y triggers de la base de datos).
- **Funcionalidades de la base de datos:** Veremos la definición de todos los procedimientos y funciones contenidos en nuestra base de datos.
- **Diseño del Almacén de datos:** También el diseño conceptual (diagrama E/R) y lógico (tablas, índices, constraints).
- **Funcionalidades del Almacén de datos y Estadísticas generadas:** Veremos los procedimientos ETL para alimentar el almacén con los datos de la base de datos operacional y algunas de las estadísticas que se pueden obtener (documento anexo a esta memoria).
- **Producto Final. Instalación y Testing:** Definiremos los scripts que forman nuestro sistema, como instalarlo en un SGBD Oracle, la carga inicial de datos y el plan de pruebas seguido (documento anexo a esta memoria).

5 Análisis

5.1 Requerimientos

Se pretende implementar una base de datos para realizar la gestión del servicio postventa de una empresa especializada en el segmento de los electrodomésticos.

Como modelo seguiremos el común en una empresa de servicios: recepción de la petición de clientes, apertura de un boletín de actuación de servicio y registro de las acciones realizadas así como de su resolución.

En un principio los servicios que realizará la empresa son los de: Instalación, reparación y consulta técnica., pudiendo ser ampliado según las necesidades futuras.

El personal de la empresa se clasificará en Teleoperadores y Técnicos.

Los Teleoperadores serán los encargados de recibir vía telefónica las peticiones de servicio de los clientes, registrarlas en el sistema y realizar la encuesta de satisfacción final.

Los Técnicos serán los encargados de realizar el servicio solicitado por el cliente. Cada técnico tendrá asociado a su perfil uno o más servicios, así como una zona geográfica de trabajo. El sistema proporcionará a cada técnico una lista de trabajos pendientes, permitiéndole su asignación y su posterior cierre.

Con todo esto, el sistema tiene que poder:

- Permitir dar de alta y baja al personal de la empresa, así como modificar datos y configurar perfiles.
- Permitir dar de alta, baja a los clientes, así como modificar sus datos.
- Recoger las peticiones de servicio de los clientes.
- Proporcionar a los técnicos información de los trabajos pendientes de ser atendidos según su perfil.
- Registrar información por parte de los técnicos de la asignación a una petición de servicio y de los trabajos realizados.
- Permitir a los clientes consultar, modificar o anular peticiones de servicio.
- Realizar estadísticas con la información almacenada como: tiempos medios en solucionar las peticiones de servicio, teleoperadores que más llamadas atienden, técnicos más productivos.

Aprovechando los datos operacionales almacenados en el día a día, se creará un Almacén de Datos (Data Warehouse) que proporcionará información necesaria para el conocimiento del negocio y la toma de decisiones.

La arquitectura del almacén de datos (DW) de nuestro sistema, seguirá el siguiente diagrama:

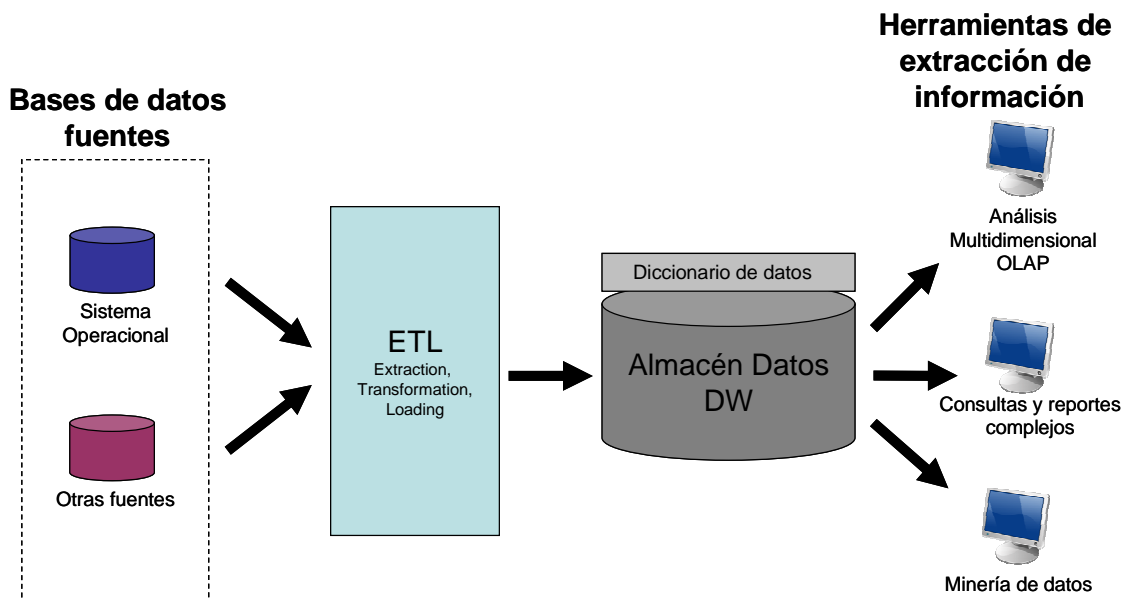


Figura 4: Arquitectura lógica de un almacén de datos.

5.1.1 Requerimientos No Funcionales

La parte primordial de nuestro sistema radicará en el hardware que soportará todo nuestro sistema. La primera pregunta a resolver será si montamos el sistema operacional (OLTP) y el almacén de datos (DW) en una o dos máquinas.

Las ventajas de montar un sistema en dos máquinas son importantes y podemos destacar las siguientes:

- La alta demanda de recursos por el DW puede ralentizar los procesos del OLTP.
- Las configuraciones de ambos sistemas son distintas
- Los datos del DW pueden provenir de distintos OLTP.

En cuanto el optar por la instalación en una única máquina.

- Al encontrarse los dos sistemas en un único equipo, los datos tratados y copiados no necesitarán ser enviados entre máquinas.

En nuestro caso, la estructura del almacén de datos a implementar será bastante sencilla. Nos basaremos en una serie de estructuras OLAP desde las que obtendremos las estadísticas e informes necesarios. Por esta razón, la instalación se realizará en un único equipo.

Una lista de los requerimientos no funcionales del sistema sería:

ID	RNF01
NOMBRE	ORDENADOR
DESCRIPCIÓN	
<p>Por lo expuesto anteriormente el sistema se implantará en un único equipo.</p> <p>Servidor Dell T610 CPU Intel Xeon E5620 2.40GHz 12M Caché RAM 8Gb DDR3-1066MHz</p>	

ID	RNF02
NOMBRE	ALMACENAMIENTO
DESCRIPCIÓN	
<p>RAID-1: 2 discos para Sistema, Oracle, Copias de Seguridad RAID-10: 4 discos para datos (Tablespaces).</p>	

ID	RNF03
NOMBRE	ALIMENTACIÓN
DESCRIPCIÓN	
<p>El equipo dispondrá de dos fuentes de alimentación. Una se conectará a la red general y otra a un SAI. De esta forma se dotará al sistema de una redundancia eléctrica.</p>	

ID	RNF04
NOMBRE	SAI (Sistema de Alimentación Ininterrumpida)
DESCRIPCIÓN	
<p>El SAI mejorará la calidad de la energía eléctrica y deberá poseer la suficiente potencia para poder terminar todas las transacciones en curso y apagar de forma ordenada el sistema.</p> <p>SAI on-line Galleon 2KVA para una carga de hasta 1600W.</p>	

ID	RNF05
NOMBRE	TARJETAS DE RED
DESCRIPCIÓN	
<p>Para proporcionar al sistema de redundancia, el sistema poseerá dos tarjetas de red Gigabyte Ethernet conectadas a dos switch distintos.</p>	

ID	RNF06
NOMBRE	COPIAS DE SEGURIDAD
DESCRIPCIÓN	
<p>Se realizará sobre el disco de sistema (Raid-1) un backup de los datos utilizando el RMAN. Una vez finalizado, se realizará un respaldo del disco de sistema sobre un sistema de backup en cinta.</p>	

ID	RNF07
NOMBRE	SISTEMA DE BACKUP
DESCRIPCIÓN	
<p>Para la realización de copias de seguridad se utilizará una unidad de cinta. Se realizarán copias totales un día a la semana e incrementales el resto.</p> <p>SAS interna HP StorageWorks DAT 320</p> <p>En principio únicamente se realizará copia de seguridad del disco de sistema, ya que contendrá tanto el sistema como la copia de seguridad de los datos realizados con RMAN.</p> <p>Periódicamente se realizará una copia íntegra y las cintas serán depositadas fuera del edificio en que resida el sistema.</p>	

ID	RNF08
NOMBRE	SISTEMA OPERATIVO
DESCRIPCIÓN	
Windows Server 2008 R2	

ID	RNF09
NOMBRE	SISTEMA GESTIÓN BASE DE DATOS
DESCRIPCIÓN	
Oracle 10g Enterprise Edition	

ID	RNF10
NOMBRE	ANÁLISIS
DESCRIPCIÓN	
Herramienta de análisis de DW. IBM Cognos Business Intelligence	

5.1.2 Requerimientos Funcionales

5.1.2.1 Diagrama

El diagrama de casos de uso del sistema completo es el siguiente:

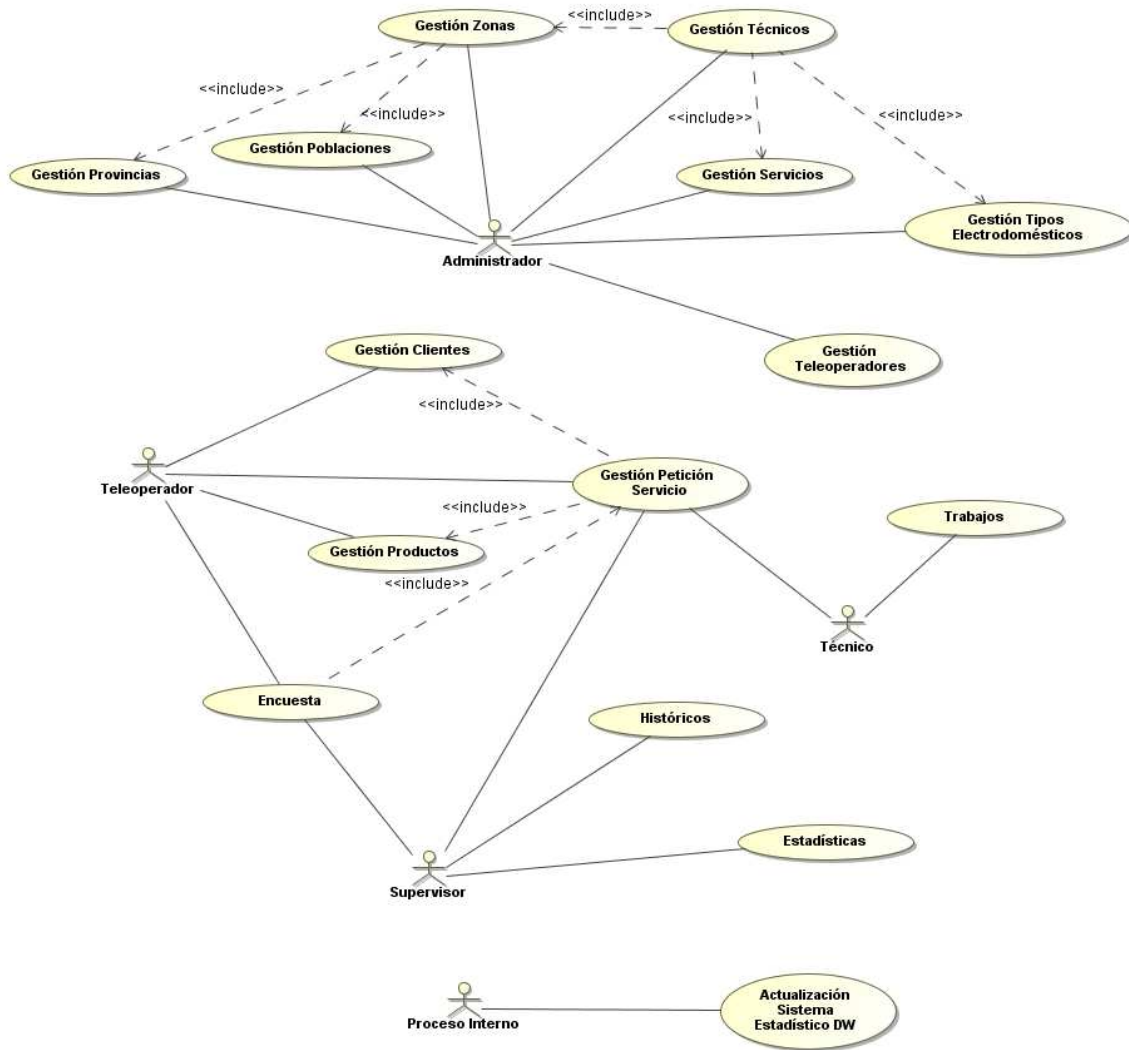


Figura 5: Diagrama de casos de uso del sistema completo.

5.1.2.2 Actores

Los actores representan los distintos roles que interactuarán con el sistema. Así, según el diagrama anterior, podemos identificar a los siguientes actores:

Administrador

Serán los usuarios del sistema encargados de realizar toda la configuración del sistema. Por ejemplo, dar de alta las provincias, poblaciones y configurar las zonas.

Teleoperador

Serán los encargados de atender telefónicamente a los clientes y registrar sus peticiones de servicio. Serán los encargados de realizar la posterior encuesta de satisfacción. También podrán atender telefónicamente a los técnicos y realizar funciones de asignación de las peticiones de servicio.

Técnicos

Serán las personas encargadas de realizar el trabajo solicitado por el cliente. Cada técnico podrá atender diferentes tipos de servicios de unos electrodomésticos determinados y en una zona geográfica.

Supervisor

Su trabajo estará enfocado al análisis de los datos recogidos por el sistema. Se encargará de consultar datos históricos y de satisfacción de cliente. Será también la persona que consultará los datos estadísticos del almacén de datos con los que podrá tomar decisiones o emprender las acciones que considere oportunas para la mejora de objetivos.

Proceso Interno

Corresponde a un proceso que se ejecutará de forma automática cada fin de mes y que actualizará la información del almacén de datos. Estos datos actualizados serán consultados por el Supervisor.

5.1.2.3 Subsistemas

Dividiremos el sistema completo en subsistemas para abordar problemas más pequeños. Así tendremos:

- SubSistema Configuración
 - o Dar de alta los distintos servicios que proporciona la empresa.
 - o Definición de tipos de electrodomésticos.
 - o Distribución geográfica en zonas.
- SubSistema Personal
 - o Dar de alta y baja a los teleoperadores, así como modificar sus datos.
 - o Dar de alta y baja a los técnicos, así como: modificar sus datos, asignar servicios y asignar zona geográfica.
- SubSistema Petición
 - o Dar de alta a clientes, así como modificar sus datos.
 - o Dar de alta productos, así como modificar sus datos.
 - o Registrar la petición de servicio para un electrodoméstico.
 - o Consultar, modificar o anular la petición de un servicio.
 - o Consultar estado de una petición de servicio.
 - o Consultar historial de una petición de servicio.
- SubSistema Atención
 - o Consultar las peticiones por un tipo de servicio y zona.
 - o Consultar las peticiones de servicio para un técnico.
 - o Asignar y desasignar petición de servicio a un técnico.
 - o Introducir trabajo(s) realizados sobre peticiones de servicio.
 - o Cierre de peticiones de servicio.
 - o Introducción encuesta satisfacción.

- SubSistema Histórico
 - o Consultar histórico de peticiones de un cliente.
 - o Consultar histórico de peticiones de un producto determinado.
 - o Consultar actuaciones entre fechas de un teleoperador.
 - o Consultar actuaciones entre fechas de un técnico.
- SubSistema Control
 - o Procedimientos Log con los movimientos efectuados en la BD.
- SubSistema Estadística
 - o Procedimientos ETL para DW.

5.1.2.4 Subsistema Configuración

5.1.2.4.1 Servicios

Los casos de uso para los Tipos de Servicio son:

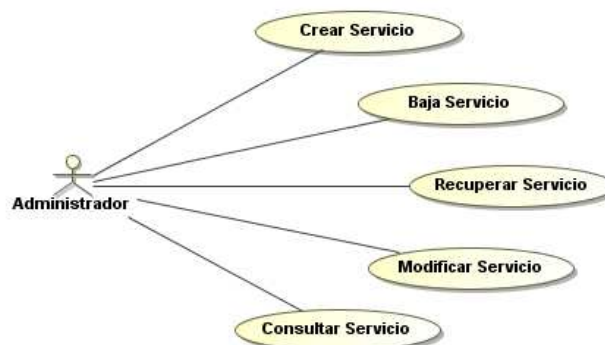


Figura 6: Casos de uso del Subsistema Configuración: Servicios.

5.1.2.4.1.1 CREAR SERVICIO (INS_SERVICIO)

Id: RF01

Función: Este procedimiento inserta un nuevo servicio en el sistema.

Funcionamiento:

- Se introduce el nuevo servicio en el sistema.
- La fecha de alta corresponderá a la fecha del sistema.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- En caso de existir un registro con la misma descripción de servicio se generará un error (*ERROR: La descripción de servicio ya existe*).
- El campo descripción no puede tener valor nulo. Se generará el error (*ERROR: Se ha intentado almacenar un valor nulo*).

Usos: GRABA_LOG.

Parámetros de entrada:

- ds: descripción del tipo de servicio.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.4.1.2 **BAJA SERVICIO (DEL_SERVICIO)**

Id: RF02

Función: Realizamos la baja lógica del servicio. La baja se produce al tener el atributo *fxBaja* un valor distinto de nulo, valor que contendrá la fecha del momento en el que se realiza la baja en el sistema.

Funcionamiento:

- Grabamos la fecha de baja que corresponderá a la fecha del sistema.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- En caso de no existir el identificador de servicio se generará un error (*ERROR: El identificador de servicio no existe*).
- Se generará un error si el servicio ya se encuentra de baja (*ERROR: El servicio ya está de baja*).

Usos: GRABA_LOG.

Parámetros de entrada:

- id: identificador del tipo de servicio.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.4.1.3 **RECUPERAR SERVICIO (RES_SERVICIO)**

Id: RF03

Función: Recuperaremos un registro que se encuentra de baja en el sistema. La recuperación se realizará al dejar de nuevo el atributo *fxBaja* con valor nulo. La fecha de alta (*fxAlta*) no variará.

Funcionamiento:

- Para el identificador de servicio dado, modificamos el valor de la fecha de baja a valor nulo.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- En caso de no existir el identificador de servicio se generará un error (*ERROR: El identificador de servicio no existe*).
- Se generará un error si el servicio ya **no** se encuentra de baja (*ERROR: El servicio no está de baja*).

Usos: GRABA_LOG.

Parámetros de entrada:

- id: identificador del tipo de servicio.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.4.1.4 **MODIFICAR SERVICIO (UPD_SERVICIO)**

Id: RF04

Función: Cambiamos la descripción del servicio.

Funcionamiento:

- Se modifican los campos de la tabla servicios (*dsServicio*) de aquel registro que tenga el *idServicio* dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador de servicio no existe se generará un error (*ERROR: El identificador de servicio no existe*).
- En caso de existir un registro con la misma descripción de servicio se generará un error (*ERROR: La descripción de servicio ya existe*).
- El campo descripción no puede tener valor nulo. Se generará el error (*ERROR: Se ha intentado actualizar a un valor nulo*).
- Si el registro que se intenta modificar se encuentra de baja, se generará el error (*ERROR: El servicio está de baja*).

Usos: GRABA_LOG.

Parámetros de entrada:

- id: identificador del tipo de servicio.
- ds: nueva descripción del tipo de servicio.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.4.1.5 **CONSULTAR SERVICIO (GET_SERVICIO)**

Id: RF05

Función: Obtiene los datos correspondientes a uno o todos los tipos servicios.

Funcionamiento:

- Se comprobará si el valor del parámetro de entrada es nulo, en cuyo caso se seleccionarán todos los tipos servicios.
- Si el valor del parámetro de entrada es distinto de nulo, seleccionaremos el registro cuyo identificador de servicio sea igual al dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador no existe, devolverá un cursor vacío.

Usos: Ninguno.

Parámetros de entrada:

- id: identificador del tipo de servicio. Podrá tener valor nulo para obtener todos los servicios.

Parámetros de salida:

- rs: cursor que podrá contener:
 - Cursor vacío si no existe ningún servicio o el *idServicio* introducido no existe.
 - Cursor con un único servicio con *idServicio* igual al introducido como parámetro.
 - Cursor con todos los servicios si el valor del parámetro *idServicio* es nulo.

5.1.2.4.2 Electrodomésticos

Los casos de uso para los Tipos de Electrodomésticos son:

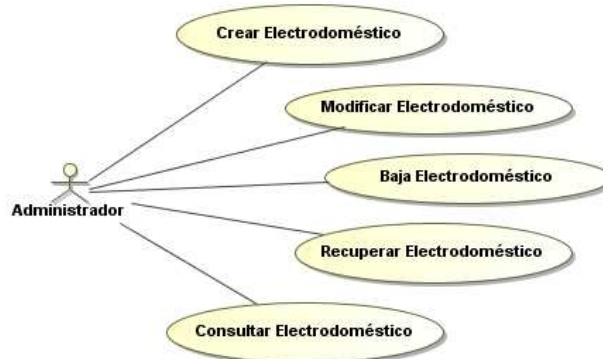


Figura 7: Casos de uso del Subsistema Configuración: Electrodomésticos.

5.1.2.4.2.1 CREAR ELECTRODOMESTICO (INS_ELECTRODOMESTICO)

Id: RF06

Función: Este procedimiento inserta un nuevo electrodoméstico en el sistema.

Funcionamiento:

- Se introduce el nuevo electrodoméstico en el sistema.
- La fecha de alta corresponderá a la fecha del sistema.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- En caso de existir un registro con la misma descripción de electrodoméstico se generará un error (*ERROR: La descripción de electrodoméstico ya existe*).
- El campo descripción no puede tener valor nulo. Se generará el error (*ERROR: Se ha intentado almacenar un valor nulo*).

Usos: GRABA_LOG.

Parámetros de entrada:

- ds: descripción del tipo de electrodoméstico.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.4.2.2 BAJA ELECTRODOMÉSTICO (DEL_ELECTRODOMESTICO)

Id: RF07

Función: Realizamos la baja lógica del electrodoméstico. La baja se produce al tener el atributo *fxBaja* un valor distinto de nulo, valor que contendrá la fecha del momento en el que se realiza la baja en el sistema.

Funcionamiento:

- Grabamos la fecha de baja que corresponderá a la fecha del sistema.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- En caso de no existir el identificador de servicio se generará un error (*ERROR: El identificador de electrodoméstico no existe*).
- Se generará un error si el servicio ya se encuentra de baja (*ERROR: El electrodoméstico ya está de baja*).

Usos: GRABA_LOG.

Parámetros de entrada:

- id: identificador del tipo de electrodoméstico.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.4.2.3 RECUPERAR ELECTRODOMÉSTICO (RES_ELECTRODOMESTICO)

Id: RF08

Función: Recuperaremos un registro que se encuentra de baja en el sistema. La recuperación se realizará al dejar de nuevo al atributo *fxBaja* un valor nulo. La fecha de alta (*fxAlta*) no variará.

Funcionamiento:

- Para el identificador de electrodoméstico dado, modificamos el valor de la fecha de baja a valor nulo.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- En caso de no existir el identificador de electrodoméstico se generará un error (*ERROR: El identificador de electrodoméstico no existe*).
- Se generará un error si el electrodoméstico ya **no** se encuentra de baja (*ERROR: El electrodoméstico no está de baja*).

Usos: GRABA_LOG.

Parámetros de entrada:

- id: identificador del tipo de electrodoméstico.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.4.2.4 MODIFICAR ELECTRODOMÉSTICO (UPD_ELECTRODOMESTICO)

Id: RF09

Función: Cambiamos la descripción del electrodoméstico.

Funcionamiento:

- Se modifican los campos de la tabla electrodomésticos (*dsElectrodomestico*) de aquel registro que tenga el *idElectrodomestico* dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador de electrodoméstico no existe se generará un error (*ERROR: El identificador de electrodoméstico no existe*).
- En caso de existir un registro con la misma descripción de electrodoméstico se generará un error (*ERROR: La descripción de electrodoméstico ya existe*).
- El campo descripción no puede tener valor nulo. Se generará el error (*ERROR: Se ha intentado actualizar a un valor nulo*).
- Si el registro que se intenta modificar se encuentra de baja, se generará el error (*ERROR: El electrodoméstico está de baja*).

Usos: GRABA_LOG.

Parámetros de entrada:

- id: identificador del tipo de electrodoméstico.
- ds: nueva descripción del tipo de electrodoméstico.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.4.2.5 CONSULTAR ELECTRODOMÉSTICO (GET_ELECTRODOMESTICO)

Id: RF10

Función: Obtiene los datos correspondientes a uno o todos los tipos de electrodomésticos.

Funcionamiento:

- Se comprobará si el valor del parámetro de entrada es nulo, en cuyo caso se seleccionarán todos los tipos electrodomésticos.
- Si el valor del parámetro de entrada es distinto de nulo, seleccionaremos el registro cuyo identificador de electrodoméstico sea igual al dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador no existe, devolverá un cursor vacío.

Usos: Ninguno.

Parámetros de entrada:

- id: identificador del tipo de electrodoméstico. Podrá tener valor nulo para obtener todos los electrodomésticos.

Parámetros de salida:

- rs: cursor que podrá contener:
 - o Cursor vacío si no existe ningún electrodoméstico o el *idElectrodomestico* introducido no existe.
 - o Cursor con un único electrodoméstico con *idElectrodomestico* igual al introducido como parámetro.
 - o Cursor con todos los electrodomésticos si el valor del parámetro *idElectrodomestico* es nulo.

5.1.2.4.3 Distribución Geográfica. Provincias

Los casos de uso para los la distribución geográfica y provincias son:

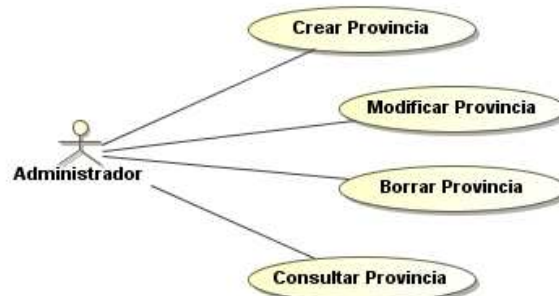


Figura 8: Casos de uso del Subsistema Configuración: Distribución Geográfica (Provincias).

5.1.2.4.3.1 CREAR PROVINCIA (INS_PROVINCIA)

Id: RF11

Función: Este procedimiento inserta una nueva provincia en el sistema.

Funcionamiento:

- Se introduce la nueva provincia en el sistema.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- En caso de existir un registro con el mismo código de provincia se generará un error (*ERROR: El código de provincia ya existe*).
- En caso de existir un registro con el mismo nombre de provincia se generará un error (*ERROR: El nombre de provincia ya existe*).
- El campo nombre de provincia (*dsProvincia*) no puede tener valor nulo. Se generará el error (*ERROR: Se ha intentado almacenar un valor nulo*).

Usos: GRABA_LOG.

Parámetros de entrada:

- cod: código de provincia.
- prov: nombre de provincia.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.4.3.2 **BORRAR PROVINCIA (DEL_PROVINCIA)**

Id: RF12

Función: Realizamos la baja total de la provincia siempre y cuando no tenga ninguna población asignada.

Funcionamiento:

- Borrarnos la provincia del sistema.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- En caso de no existir el código de provincia se generará un error (*ERROR: El código de provincia no existe*).
- Se generará un error si existen poblaciones asignadas a esta provincia (*ERROR: La provincia tiene poblaciones asignadas*).

Usos: GRABA_LOG.

Parámetros de entrada:

- cod: código de la provincia.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.4.3.3 **MODIFICAR PROVINCIA (UPD_PROVINCIA)**

Id: RF13

Función: Cambiamos el nombre de una provincia.

Funcionamiento:

- Se modifican los campos de la tabla provincias (*dsProvincia*) de aquel registro que tenga el *coProvincia* dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador de provincia no existe se generará un error (*ERROR: El código de provincia no existe*).
- En caso de existir un registro con el mismo nombre de provincia se generará un error (*ERROR: El nombre de la provincia ya existe*).
- El campo nombre de provincia no puede tener valor nulo. Se generará el error (*ERROR: Se ha intentado actualizar a un valor nulo*).

Usos: GRABA_LOG.

Parámetros de entrada:

- cod: código de la provincia.
- prov: nuevo nombre de la provincia.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.4.3.4 CONSULTAR PROVINCIA (GET_PROVINCIA)

Id: RF14

Función: Obtiene los datos correspondientes a una o todas las provincias.

Funcionamiento:

- Se comprobará si el valor del parámetro de entrada es nulo, en cuyo caso se seleccionarán todas las provincias.
- Si el valor del parámetro de entrada es distinto de nulo, seleccionaremos el registro cuyo código de provincia sea igual al dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el código de provincia no existe, devolverá un cursor vacío.

Usos: Ninguno.

Parámetros de entrada:

- cod: código de provincia. Podrá tener valor nulo para obtener todas las provincias.

Parámetros de salida:

- rs: cursor que podrá contener:
 - o Cursor vacío si no existe ninguna provincia o el *coProvincia* introducido no existe.
 - o Cursor con una única provincia con *coProvincia* igual al introducido como parámetro.
 - o Cursor con todas las provincias si el valor del parámetro *coProvincia* es nulo.

5.1.2.4.4 Distribución Geográfica. Poblaciones

Los casos de uso para los la distribución geográfica y poblaciones son:

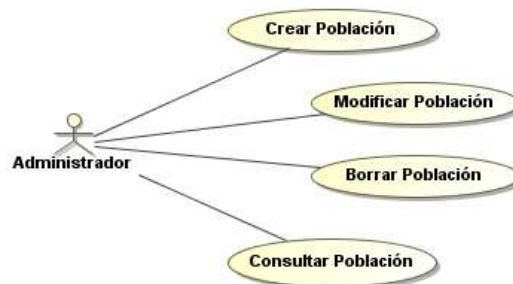


Figura 9: Casos de uso del Subsistema Configuración: Distribución Geográfica (Poblaciones).

5.1.2.4.4.1 **CREAR POBLACION (INS_POBLACION)**

Id: RF15

Función: Este procedimiento inserta una nueva población en el sistema.

Funcionamiento:

- Se introduce la nueva población en el sistema.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- En caso de existir un registro con el mismo código postal se generará un error (*ERROR: El código postal ya existe*).
- En caso de existir un registro con el mismo nombre de población se generará un error (*ERROR: El nombre de la población ya existe*).
- Los campos nombre de población (*dsPoblacion*) y código de provincia (*coProvincia*) no pueden tener valor nulo. Se generará el error (*ERROR: Se ha intentado almacenar un valor nulo*).
- El código de provincia al que pertenece la población deberá existir en el sistema. Si no existe, se generará el error (*ERROR: El código de provincia no existe*).

Usos: GRABA_LOG.

Parámetros de entrada:

- cPostal: código postal de la población.
- población: nombre de la población.
- codProv: código de la provincia.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.4.4.2 **BORRAR POBLACIÓN (DEL_POBLACION)**

Id: RF16

Función: Realizamos la baja total de la población siempre y cuando no esté asociada a ninguna zona.

Funcionamiento:

- Borrarnos la población del sistema.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- En caso de no existir el código postal se generará un error (*ERROR: El código postal no existe*).
- Se generará un error si la población esta asignada a alguna zona (*ERROR: La provincia esta asignada a alguna zona*).

Usos: GRABA_LOG.

Parámetros de entrada:

- cPostal: código postal de la población.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.4.4.3 MODIFICAR POBLACION (UPD_POBLACION)

Id: RF17

Función: Cambiamos el nombre de una población y/o la provincia a la que pertenece.

Funcionamiento:

- Se modifican los campos de la tabla poblaciones (*dsPoblacion*, *coProvincia*) de aquel registro que tenga el *codPostal* dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el código postal no existe se generará un error (*ERROR: El código postal no existe*).
- En caso de existir un registro con el mismo nombre de población se generará un error (*ERROR: El nombre de la población ya existe*).
- Los campos nombre de población y código de provincia no pueden tener valor nulo. Se generará el error (*ERROR: Se ha intentado actualizar a un valor nulo*).
- El código de provincia al que pertenece la población deberá existir en el sistema. Si no existe, se generará el error (*ERROR: El código de provincia no existe*).

Usos: GRABA_LOG.

Parámetros de entrada:

- cPostal: código postal de la población.
- poblacion: nuevo nombre de la población.
- codProv: nuevo código de provincia.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.4.4.4 CONSULTAR POBLACION (GET_POBLACION)

Id: RF18

Función: Obtiene los datos correspondientes a una o todas las poblaciones.

Funcionamiento:

- Se comprobará si el valor del parámetro de entrada es nulo, en cuyo caso se seleccionarán todas las poblaciones.
- Si el valor del parámetro de entrada es distinto de nulo, seleccionaremos el registro cuyo código postal sea igual al dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el código postal no existe, devolverá un cursor vacío.

Usos: Ninguno.

Parámetros de entrada:

- cPostal: código postal de la población. Podrá tener valor nulo para obtener todas las poblaciones.

Parámetros de salida:

- rs: cursor que podrá contener:
 - o Cursor vacío si no existe ninguna población o el *codPostal* introducido no existe.
 - o Cursor con una única población con *codPostal* igual al introducido como parámetro.
 - o Cursor con todas las poblaciones si el valor del parámetro *codPostal* es nulo.

5.1.2.4.5 Distribución Geográfica. Zonas

Los casos de uso para los la distribución geográfica y zonas son:



Figura 10: Casos de uso del Subsistema Configuración: Distribución Geográfica (Zonas).

5.1.2.4.5.1 CREAR ZONA (INS_ZONA)

Id: RF19

Función: Este procedimiento inserta una nueva zona en el sistema.

Funcionamiento:

- Se introduce la nueva zona en el sistema.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- En caso de existir un registro con la misma descripción de zona se generará un error (*ERROR: La descripción de la zona ya existe*).
- El campo descripción no puede tener valor nulo. Se generará el error (*ERROR: Se ha intentado almacenar un valor nulo*).

Usos: GRABA_LOG.

Parámetros de entrada:

- ds: descripción de la zona.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.4.5.2 **BORRAR ZONA (DEL_ZONA)**

Id: RF20

Función: Realizamos la baja total de la zona siempre y cuando no tenga asociada ninguna población.

Funcionamiento:

- Borramos la zona del sistema.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- En caso de no existir el identificador de zona se generará un error (*ERROR: El identificador de zona no existe*).
- Se generará un error si existen poblaciones asignadas a esta zona (*ERROR: La zona tiene poblaciones asignadas*).

Usos: GRABA_LOG.

Parámetros de entrada:

- id: identificador de la zona.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.4.5.3 **MODIFICAR ZONA (UPD_ZONA)**

Id: RF21

Función: Cambiamos la descripción de una zona.

Funcionamiento:

- Se modifican los campos de la tabla zonas (*dsZona*) de aquel registro que tenga el *idZona* dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador de zona no existe se generará un error (*ERROR: El código de zona no existe*).
- En caso de existir un registro con la misma descripción de la zona se generará un error (*ERROR: La descripción de zona ya existe*).
- El campo descripción de zona no puede tener valor nulo. Se generará el error (*ERROR: Se ha intentado actualizar a un valor nulo*).

Usos: GRABA_LOG.

Parámetros de entrada:

- id: identificador de la zona.
- ds: descripción de la zona.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.4.5.4 **CONSULTAR ZONA (GET_ZONA)**

Id: RF22

Función: Obtiene los datos correspondientes a una o todas las zonas.

Funcionamiento:

- Se comprobará si el valor del parámetro de entrada es nulo, en cuyo caso se seleccionarán todas las zonas.
- Si el valor del parámetro de entrada es distinto de nulo, seleccionaremos el registro cuyo identificador de zona sea igual al dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador de zona no existe, devolverá un cursor vacío.

Usos: Ninguno.

Parámetros de entrada:

- id: identificador de zona. Podrá tener valor nulo para obtener todas las zonas.

Parámetros de salida:

- rs: cursor que podrá contener:
 - o Cursor vacío si no existe ninguna zona o el *idZona* introducido no existe.
 - o Cursor con una única zona con *idZona* igual al introducido como parámetro.
 - o Cursor con todas las zonas si el valor del parámetro *idZona* es nulo.

5.1.2.4.5.5 **ASIGNAR PROVINCIA (ASG_PROVINCIAZONA)**

Id: RF23

Función: Asignará todas las poblaciones que pertenecen a una provincia a una zona.

Funcionamiento:

- Se insertan en la tabla ZONASPOBLACION aquellas poblaciones que pertenecen a la provincia dada y siempre que ya no esté asociada a la zona.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el código de la provincia no existe se generará un error (*ERROR: El código de provincia no existe*).
- Si el identificador de zona no existe y se generará un error (*ERROR: El identificador de zona no existe*).

Usos: GRABA_LOG.

Parámetros de entrada:

- coProv: código de la provincia.
- zona: identificador de la zona.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.4.5.6 **ASIGNAR POBLACION (ASG_POBLACIONZONA)**

Id: RF24

Función: Asignará una población a una zona.

Funcionamiento:

- Tras comprobar la existencia del código postal de la población, se realiza la asociación insertando en la tabla ZONASPOBLACION la combinación de identificador de zona y código postal de la población.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el código postal de la población no existe se generará un error (*ERROR: El código postal no existe*).
- Si el identificador de zona no existe y se generará un error (*ERROR: El identificador de zona no existe*).
- Si la población ya se encontraba asociada a la provincia se generará el error (*ERROR: La población ya está asociada a la zona*).

Usos: GRABA_LOG.

Parámetros de entrada:

- cp: código postal de la población.
- zona: identificador de la zona.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.4.5.7 **DESASIGNAR PROVINCIA (DSG_PROVINCIAZONA)**

Id: RF25

Función: Eliminará la asignación de todas las poblaciones que pertenecen a una provincia de una zona dada.

Funcionamiento:

- Se eliminan de la tabla ZONASPOBLACION aquellas poblaciones que pertenecen a la provincia dada y se encuentran asignadas a la zona dada.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el código de la provincia no existe se generará un error (*ERROR: El código de provincia no existe*).
- Si el identificador de zona no existe y se generará un error (*ERROR: El identificador de zona no existe*).

Usos: GRABA_LOG.

Parámetros de entrada:

- coProv: código de la provincia.
- zona: identificador de la zona.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.4.5.8 DESASIGNAR POBLACION (DSG_POBLACIONZONA)

Id: RF26

Función: Eliminará la asignación de una población a una zona.

Funcionamiento:

- Tras comprobar la existencia de la población y la zona, se eliminará de la tabla ZONASPOBLACION la combinación de identificador de zona y código postal de la población.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el código postal de la población no existe se generará un error (*ERROR: El código postal no existe*).
- Si el identificador de zona no existe y se generará un error (*ERROR: El identificador de zona no existe*).

Usos: GRABA_LOG.

Parámetros de entrada:

- cp: código postal de la población.
- zona: identificador de la zona.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.4.5.9 CONSULTAR POBLACIONES DE ZONA (GET_POBLACIONESZONA)

Id: RF27

Función: Obtiene los datos correspondientes a una o todas las poblaciones asignadas a una zona.

Funcionamiento:

- Se comprobará si el valor del parámetro de entrada es nulo, en cuyo caso se seleccionarán todas las zonas.
- Si el valor del parámetro de entrada es distinto de nulo, seleccionaremos el registro cuyo identificador de zona sea igual al dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador de zona no existe, devolverá un cursor vacío.

Usos: Ninguno.

Parámetros de entrada:

- id: identificador de zona. Podrá tener valor nulo para obtener todas las zonas.

Parámetros de salida:

- rs: cursor que podrá contener:
 - o Cursor vacío si no existe ninguna zona o el *idZona* introducido no existe.
 - o Cursor con una única zona con *idZona* igual al introducido como parámetro.
 - o Cursor con todas las zonas si el valor del parámetro *idZona* es nulo.

5.1.2.5 Subsistema Personal

5.1.2.5.1 Teleoperadores

Los casos de uso para los Teleoperadores son:

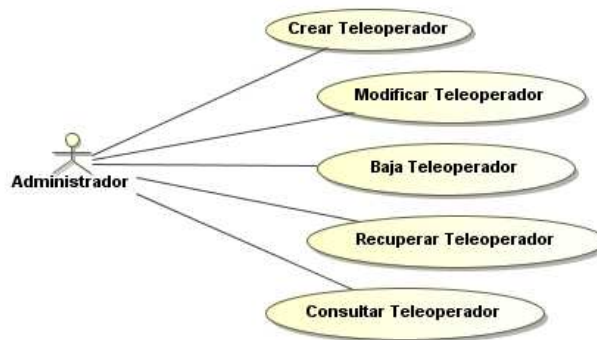


Figura 11: Casos de uso del Subsistema Personal: Teleoperadores.

5.1.2.5.1.1 **CREAR TELEOPERADOR (INS_TELEOPERADOR)**

Id: RF28

Función: Este procedimiento inserta un nuevo teleoperador en el sistema.

Funcionamiento:

- Se introduce el nuevo teleoperador en el sistema.
- La fecha de alta corresponderá a la fecha del sistema.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- En caso de existir un teleoperador con el mismo nif se generará un error (*ERROR: El nif del teleoperador ya existe*).
- Los campos nombre, apellidos y nif no puede tener valor nulo. En caso de que alguno de ellos tenga valor nulo, se generará el error (*ERROR: Se ha intentado almacenar un valor nulo*).

Usos: GRABA_LOG.

Parámetros de entrada:

- pNombre: nombre del teleoperador.
- pApellidos: apellidos del teleoperador.
- pNif: nif del teleoperador.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.5.1.2 BAJA TELEOPERADOR (DEL_TELEOPERADOR)

Id: RF29

Función: Realizamos la baja lógica del teleoperador. La baja se produce al tener el atributo *fxBaja* un valor distinto de nulo, valor que contendrá la fecha del momento en el que se realiza la baja en el sistema.

Funcionamiento:

- Grabamos la fecha de baja que corresponderá a la fecha del sistema.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- En caso de no existir el identificador del teleoperador se generará un error (*ERROR: El identificador del teleoperador no existe*).
- Se generará un error si el teleoperador ya se encuentra de baja (*ERROR: El teleoperador ya está de baja*).

Usos: GRABA_LOG.

Parámetros de entrada:

- pld: identificador del teleoperador.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.5.1.3 RECUPERAR TELEOPERADOR (RES_TELEOPERADOR)

Id: RF30

Función: Recuperaremos un registro que se encuentra de baja en el sistema. La recuperación se realizará al dejar de nuevo al atributo *fxBaja* un valor nulo. La fecha de alta (*fxAlta*) no variará.

Funcionamiento:

- Para el identificador de teleoperador dado, modificamos el valor de la fecha de baja a valor nulo.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- En caso de no existir el identificador del teleoperador se generará un error (*ERROR: El identificador del teleoperador no existe*).
- Se generará un error si el teleoperador ya **no** se encuentra de baja (*ERROR: El teleoperador no está de baja*).

Usos: GRABA_LOG.

Parámetros de entrada:

- pld: identificador del teleoperador.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.5.1.4 MODIFICAR TELEOPERADOR (UPD_TELEOPERADOR)

Id: RF31

Función: Cambiamos los datos (nombre, apellidos y nif) del teleoperador.

Funcionamiento:

- Se modifican los campos de la tabla teleoperadores (*nombre, apellidos y nif*) de aquel registro que tenga el *idTeleoperador* dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador de teleoperador no existe se generará un error (*ERROR: El identificador del teleoperador no existe*).
- En caso de existir un registro con el mismo nif, se generará un error (*ERROR: El nif del teleoperador ya existe*).
- Los campos nombre, apellidos y nif no pueden tener valor nulo. En caso de valor nulo se generará el error (*ERROR: Se ha intentado actualizar a un valor nulo*).
- Si el registro que se intenta modificar se encuentra de baja, se generará el error (*ERROR: El teleoperador está de baja*).

Usos: GRABA_LOG.

Parámetros de entrada:

- pid: identificador del teleoperador.
- pNombre: nuevo nombre del teleoperador.
- pApellidos: nuevos apellidos del teleoperador.
- pNif: nuevo nif del teleoperador.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.5.1.5 CONSULTAR TELEOPERADOR (GET_TELEOPERADOR)

Id: RF32

Función: Obtiene los datos correspondientes a uno o todos los teleoperadores.

Funcionamiento:

- Se comprobará si el valor del parámetro de entrada es nulo, en cuyo caso se seleccionarán todos los teleoperadores.
- Si el valor del parámetro de entrada es distinto de nulo, seleccionaremos el registro cuyo identificador de teleoperador sea igual al dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador no existe, devolverá un cursor vacío.

Usos: Ninguno.

Parámetros de entrada:

- pld: identificador del teleoperador. Podrá tener valor nulo para obtener todos los teleoperadores.

Parámetros de salida:

- rs: cursor que podrá contener:
 - o Cursor vacío si no existe ningún teleoperador o el *idTeleoperador* introducido no existe.
 - o Cursor con un único teleoperador con *idTeleoperador* igual al introducido como parámetro.
 - o Cursor con todos los teleoperadores si el valor del parámetro *idTeleoperador* es nulo.

5.1.2.5.2 Técnicos

Los casos de uso para los Técnicos son:

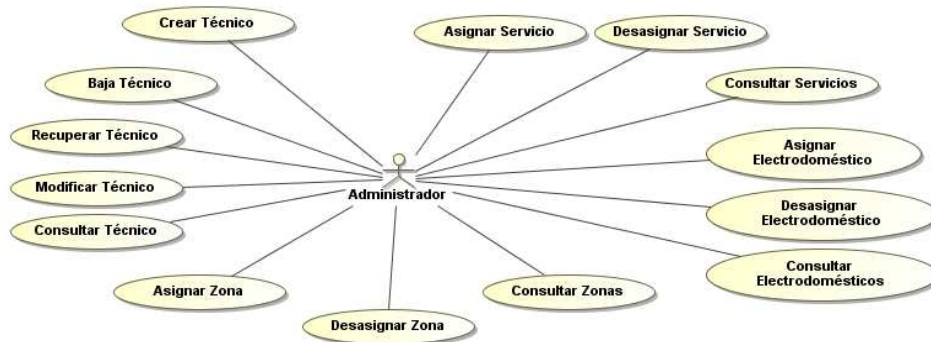


Figura 12: Casos de uso del Subsistema Personal: Técnicos.

5.1.2.5.2.1 CREAR TECNICO (INS_TECNICO)

Id: RF33

Función: Este procedimiento inserta un nuevo técnico en el sistema.

Funcionamiento:

- Se introduce el nuevo técnico en el sistema.
- La fecha de alta corresponderá a la fecha del sistema.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- En caso de existir un técnico con el mismo nif se generará un error (*ERROR: El nif del técnico ya existe*).
- Los campos nombre, apellidos, nif y teléfono móvil no puede tener valor nulo. En caso de que alguno de ellos tenga valor nulo, se generará el error (*ERROR: Se ha intentado almacenar un valor nulo*).

Usos: GRABA_LOG.

Parámetros de entrada:

- pNombre: nombre del técnico.
- pApellidos: apellidos del técnico.
- pNif: nif del técnico.
- pTelMovil: número de teléfono móvil del técnico.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.5.2.2 **BAJA TECNICO (DEL_TECNICO)**

Id: RF34

Función: Realizamos la baja lógica del técnico. La baja se produce al tener el atributo *fxBaja* un valor distinto de nulo, valor que contendrá la fecha del momento en el que se realiza la baja en el sistema.

Funcionamiento:

- Grabamos la fecha de baja que corresponderá a la fecha del sistema.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- En caso de no existir el identificador del técnico se generará un error (*ERROR: El identificador del técnico no existe*).
- Se generará un error si el técnico ya se encuentra de baja (*ERROR: El técnico ya está de baja*).

Usos: GRABA_LOG.

Parámetros de entrada:

- pId: identificador del técnico.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.5.2.3 **RECUPERAR TECNICO (RES_TECNICO)**

Id: RF35

Función: Recuperaremos un registro que se encuentra de baja en el sistema. La recuperación se realizará al dejar de nuevo al atributo *fxBaja* un valor nulo. La fecha de alta (*fxAlta*) no variará.

Funcionamiento:

- Para el identificador de técnico dado, modificamos el valor de la fecha de baja a valor nulo.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- En caso de no existir el identificador del técnico se generará un error (*ERROR: El identificador del técnico no existe*).
- Se generará un error si el técnico ya **no** se encuentra de baja (*ERROR: El técnico no está de baja*).

Usos: GRABA_LOG.

Parámetros de entrada:

- pld: identificador del técnico.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.5.2.4 **MODIFICAR TECNICO (UPD_TECNICO)**

Id: RF36

Función: Cambiamos los datos (nombre, apellidos, nif y teléfono móvil) del técnico.

Funcionamiento:

- Se modifican los campos de la tabla técnicos (*nombre, apellidos, nif y teléfono móvil*) de aquel registro que tenga el *idTecnico* dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador de técnico no existe se generará un error (*ERROR: El identificador del técnico no existe*).
- En caso de existir un registro con el mismo nif, se generará un error (*ERROR: El nif del técnico ya existe*).
- Los campos nombre, apellidos, nif y teléfono móvil no pueden tener valor nulo. En caso de valor nulo se generará el error (*ERROR: Se ha intentado actualizar a un valor nulo*).
- Si el registro que se intenta modificar se encuentra de baja, se generará el error (*ERROR: El técnico está de baja*).

Usos: GRABA_LOG.

Parámetros de entrada:

- pid: identificador del técnico.
- pNombre: nuevo nombre del técnico.
- pApellidos: nuevos apellidos del técnico.
- pNif: nuevo nif del técnico.
- pTelMovil: nuevo número de teléfono móvil del técnico.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.5.2.5 **CONSULTAR TECNICO (GET_TECNICO)**

Id: RF37

Función: Obtiene los datos correspondientes a uno o todos los técnicos.

Funcionamiento:

- Se comprobará si el valor del parámetro de entrada es nulo, en cuyo caso se seleccionarán todos los técnicos.
- Si el valor del parámetro de entrada es distinto de nulo, seleccionaremos el registro cuyo identificador de técnico sea igual al dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador no existe, devolverá un cursor vacío.

Usos: Ninguno.

Parámetros de entrada:

- pld: identificador del técnico. Podrá tener valor nulo para obtener todos los técnicos.

Parámetros de salida:

- rs: cursor que podrá contener:
 - o Cursor vacío si no existe ningún técnico o el *idTecnico* introducido no existe.
 - o Cursor con un único técnico con *idTecnico* igual al introducido como parámetro.
 - o Cursor con todos los técnicos si el valor del parámetro *idTecnico* es nulo.

5.1.2.5.2.6 **ASIGNAR SERVICIO A TECNICO (ASG_SERVICIOTECNICO)**

Id: RF38

Función: Asignará un tipo de servicio a un técnico.

Funcionamiento:

- Se realiza la asociación insertando en la tabla TECNICOSSERVICIOS la combinación de identificador de servicio e identificador de técnico.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador del técnico no existe se generará un error (*ERROR: El identificador de técnico no existe*).
- Si el técnico se encuentra dado de baja se generará un error (*ERROR: El técnico está de baja*).
- Si el identificador de servicio no existe se generará un error (*ERROR: El identificador de servicio no existe*).
- Si el servicio está de baja se generará un error (*ERROR: El servicio está de baja*).
- Si el servicio ya se encontraba asociado al técnico se generará el error (*ERROR: El servicio ya está asociado al técnico*).

Usos: GRABA_LOG.

Parámetros de entrada:

- pldServicio: identificador del tipo de servicio.
- pldTecnico: identificador del técnico.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.5.2.7 **DESASIGNAR SERVICIO A TECNICO (DSG_SERVICIOTECNICO)**

Id: RF39

Función: Eliminará la asignación de un servicio a un técnico.

Funcionamiento:

- Tras comprobar la existencia del servicio y el técnico, se eliminará de la tabla TECNICOSSERVICIOS la combinación de identificador de servicio e identificador de técnico.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador del técnico no existe se generará un error (*ERROR: El identificador técnico no existe*).
- Si el técnico se encuentra dado de baja se generará un error (*ERROR: El técnico está de baja*).
- Si el identificador del servicio no existe se generará un error (*ERROR: El identificador de servicio no existe*).
- Si el servicio está de baja se generará un error (*ERROR: El servicio está de baja*).

Usos: GRABA_LOG.

Parámetros de entrada:

- pIdServicio: Identificador del tipo de servicio.
- pIdTecnico: identificador del técnico.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.5.2.8 **CONSULTAR SERVICIOS DE TECNICO (GET_SERVICIOSTECNICO)**

Id: RF40

Función: Obtiene los datos correspondientes a uno o todos los servicios atendidos por uno o todos los técnicos.

Funcionamiento:

- Se comprobará si el valor de los parámetro de entrada son nulos, en cuyo caso se seleccionarán todos los servicios de todos los técnicos.
- Si el valor del parámetro de entrada es distinto de nulo, seleccionaremos el registro cuyo identificador de servicio y/o técnico sea igual al dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador de técnico no existe, devolverá un cursor vacío.

Usos: Ninguno.

Parámetros de entrada:

- pIdTecnico: identificador del técnico. Podrá tener valor nulo para obtener todos los técnicos.
- pIdServicio: identificador del servicio. Podrá tener valor nulo para obtener todos los servicios.

Parámetros de salida:

- rs: cursor que podrá contener:
 - o Cursor vacío si no existe ningún técnico o el *idTecnico* introducido no existe.

- Cursor con todos los servicio de un técnico con *idTecnico* igual al introducido como parámetro si *idServicio* es nulo.
- Cursos con todos los técnicos que tienen asignado el servicio *idServicio* igual al introducido como parámetro si *idTecnico* es nulo.
- Cursor con todos los técnicos y sus servicios asociados si el valor del parámetro *idTecnico* e *idServicio* son nulos.

5.1.2.5.2.9 **ASIGNAR ELECTRODOMÉSTICO A TECNICO** **(ASG_ELECTRODOMESTICOTECNICO)**

Id: RF41

Función: Asignará un tipo de electrodoméstico a un técnico.

Funcionamiento:

- Se realiza la asociación insertando en la tabla ELECTRODOMESTICOSTECNICO la combinación de identificador de electrodoméstico e identificador de técnico.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador del técnico no existe se generará un error (*ERROR: El identificador de técnico no existe*).
- Si el técnico se encuentra dado de baja se generará un error (*ERROR: El técnico está de baja*).
- Si el identificador de electrodoméstico no existe y se generará un error (*ERROR: El identificador de electrodoméstico no existe*).
- Si el electrodoméstico se encuentra dado de baja se generará un error (*ERROR: El electrodoméstico está de baja*).
- Si el tipo de electrodoméstico ya se encontraba asociado al técnico se generará el error (*ERROR: El servicio ya está asociado al técnico*).

Usos: GRABA_LOG.

Parámetros de entrada:

- pIdElectrodomestico: identificador del tipo de electrodoméstico.
- pIdTecnico: identificador del técnico.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.5.2.10 **DESASIGNAR ELECTRODOMÉSTICO A TECNICO** **(DSG_ELECTRODOMESTICOTECNICO)**

Id: RF42

Función: Eliminará la asignación de un electrodoméstico a un técnico.

Funcionamiento:

- Tras comprobar la existencia del tipo de electrodoméstico y el técnico, se eliminará de la tabla TECNICOSELECTRODOMESTICOS la combinación de identificador de electrodoméstico e identificador de técnico.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador del técnico no existe se generará un error (*ERROR: El identificador técnico no existe*).
- Si el técnico se encuentra dado de baja se generará un error (*ERROR: El técnico está de baja*).
- Si el identificador del electrodoméstico no existe se generará un error (*ERROR: El identificador de electrodoméstico no existe*).
- Si el electrodoméstico se encuentra dado de baja se generará un error (*ERROR: El electrodoméstico está de baja*).

Usos: GRABA_LOG.

Parámetros de entrada:

- pldElectrodomestico: Identificador del tipo de electrodoméstico.
- pldTecnico: identificador del técnico.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

**5.1.2.5.2.11 CONSULTAR ELECTRODOMÉSTICOS DE TECNICO
(GET_ELECTRODOMESTICOSTECNICO)**

Id: RF43

Función: Obtiene los datos correspondientes a uno o todos los electrodomésticos reparados por uno o todos los técnicos.

Funcionamiento:

- Se comprobará si el valor de los parámetro de entrada son nulos, en cuyo caso se seleccionarán todos los electrodomésticos de todos los técnicos.
- Si el valor del parámetro de entrada es distinto de nulo, seleccionaremos el registro cuyo identificador de electrodoméstico y/o técnico sea igual al dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador de técnico no existe, devolverá un cursor vacío.

Usos: Ninguno.

Parámetros de entrada:

- pldTecnico: identificador del técnico. Podrá tener valor nulo para obtener todos los técnicos.
- pldElectrodomestico: identificador del electrodoméstico. Podrá tener valor nulo para obtener todos los electrodomésticos.

Parámetros de salida:

- rs: cursor que podrá contener:
 - Cursor vacío si no existe ningún técnico o el *idTecnico* introducido no existe.
 - Cursor con todos los electrodomésticos de un técnico con *idTecnico* igual al introducido como parámetro si *idElectrodomestico* es nulo.
 - Cursos con todos los técnicos que tienen asignado el electrodoméstico *idElectrodomestico* igual al introducido como parámetro si *idTecnico* es nulo.

- Cursor con todos los técnicos y sus electrodomésticos asociados si el valor del parámetro *idTecnico* e *idElectrodomestico* son nulos.

5.1.2.5.2.12 **ASIGNAR ZONA A TECNICO (ASG_ZONATECNICO)**

Id: RF44

Función: Asignará una zona a un técnico.

Funcionamiento:

- Se realiza la asociación insertando en la tabla ZONASTECNICO la combinación de identificador de zona e identificador de técnico.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador del técnico no existe se generará un error (*ERROR: El identificador de técnico no existe*).
- Si el técnico se encuentra dado de baja se generará un error (*ERROR: El técnico está de baja*).
- Si el identificador de la zona no existe y se generará un error (*ERROR: El identificador de zona no existe*).
- Si la zona ya se encontraba asociado al técnico se generará el error (*ERROR: La zona ya está asociada al técnico*).

Usos: GRABA_LOG.

Parámetros de entrada:

- pIdzona: identificador de la zona.
- pIdTecnico: identificador del técnico.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.5.2.13 **DESASIGNAR ZONA A TECNICO (DSG_ZONATECNICO)**

Id: RF45

Función: Eliminará la asignación de una zona a un técnico.

Funcionamiento:

- Tras comprobar la existencia de la zona y el técnico, se eliminará de la tabla TECNICOSZONAS la combinación de identificador de electrodoméstico e identificador de técnico.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador del técnico no existe se generará un error (*ERROR: El identificador técnico no existe*).
- Si el técnico se encuentra dado de baja se generará un error (*ERROR: El técnico está de baja*).
- Si el identificador de la zona no existe se generará un error (*ERROR: El identificador de zona no existe*).

Usos: GRABA_LOG.

Parámetros de entrada:

- pIdZona: Identificador de zona.
- pIdTecnico: identificador del técnico.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.5.2.14 **CONSULTAR ZONAS DE TECNICO (GET_ZONASTECNICO)**

Id: RF46

Función: Obtiene los datos correspondientes a uno o todas las zonas atendidas por uno o todos los técnicos.

Funcionamiento:

- Se comprobará si el valor de los parámetro de entrada son nulos, en cuyo caso se seleccionarán todas las zonas de todos los técnicos.
- Si el valor del parámetro de entrada es distinto de nulo, seleccionaremos el registro cuyo identificador de zona y/o técnico sea igual al dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador de técnico no existe, devolverá un cursor vacío.

Usos: Ninguno.

Parámetros de entrada:

- pIdTecnico: identificador del técnico. Podrá tener valor nulo para obtener todos los técnicos.
- pIdZona: identificador de la zona. Podrá tener valor nulo para obtener todas las zonas.

Parámetros de salida:

- rs: cursor que podrá contener:
 - o Cursor vacío si no existe ningún técnico o el *idTecnico* introducido no existe.
 - o Cursor con todas las zonas de un técnico con *idTecnico* igual al introducido como parámetro si *idZona* es nulo.
 - o Cursos con todos los técnicos que tienen asignado la zona *idZona* igual al introducido como parámetro si *idTecnico* es nulo.
 - o Cursor con todos los técnicos y sus zonas asociadas si el valor del parámetro *idTecnico* e *idZona* son nulos.

5.1.2.6 Subsistema Petición

5.1.2.6.1 Clientes

Los casos de uso para los Clientes son:

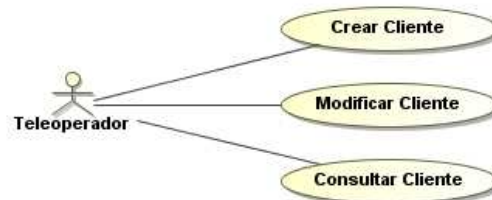


Figura 13: Casos de uso del Subsistema Petición: Clientes.

5.1.2.6.1.1 CREAR CLIENTE (INS_CLIENTE)

Id: RF47

Función: Este procedimiento inserta un nuevo cliente en el sistema.

Funcionamiento:

- Se introduce el nuevo cliente en el sistema.
- La fecha de alta corresponderá a la fecha del sistema.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- En caso de existir un cliente con el mismo nif se generará un error (*ERROR: El nif del cliente ya existe*).
- Los campos nombre, apellidos, nif, dirección, código postal, teléfono fijo, teléfono móvil y email no pueden tener valor nulo. En caso de que alguno de ellos tenga valor nulo, se generará el error (*ERROR: Se ha intentado almacenar un valor nulo*).
- El código postal de la población debe existir en el sistema. En caso de no existir, se generará el error (*ERROR: El código postal no existe*).

Usos: GRABA_LOG.

Parámetros de entrada:

- pNombre: nombre del cliente.
- pApellidos: apellidos del cliente.
- pNif: nif del cliente.
- pDireccion: dirección de cliente.
- pCodPostal: código postal de la población del cliente.
- pTelFijo: teléfono fijo del cliente.
- pTelMovil: número de teléfono móvil del cliente.
- pEMail: dirección de correo electrónico del cliente.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.6.1.2 **MODIFICAR CLIENTE (UPD_CLIENTE)**

Id: RF48

Función: Cambiamos los datos (nombre, apellidos, nif, dirección, código postal, teléfono fijo, teléfono móvil y email) del cliente.

Funcionamiento:

- Se modifican los campos de la tabla clientes (*nombre, apellidos, nif, dirección, código postal, teléfono fijo, teléfono móvil y email*) de aquel registro que tenga el *idCliente* dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador de cliente no existe se generará un error (*ERROR: El identificador del cliente no existe*).
- En caso de existir un registro con el mismo nif, se generará un error (*ERROR: El nif del cliente ya existe*).
- Los campos nombre, apellidos, nif, dirección, código postal, teléfono fijo, teléfono móvil y email no pueden tener valor nulo. En caso de valor nulo se generará el error (*ERROR: Se ha intentado actualizar a un valor nulo*).
- El código postal de la población debe existir en el sistema. En caso de no existir, se generará el error (*ERROR: El código postal no existe*).

Usos: GRABA_LOG.

Parámetros de entrada:

- pid: identificador del cliente.
- pNombre: nombre del cliente.
- pApellidos: apellidos del cliente.
- pNif: nif del cliente.
- pDireccion: dirección de cliente.
- pCodPostal: código postal de la población del cliente.
- pTelFijo: teléfono fijo del cliente.
- pTelMovil: número de teléfono móvil del cliente.
- pEMail: dirección de correo electrónico del cliente.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.6.1.3 **CONSULTAR CLIENTE (GET_CLIENTE)**

Id: RF49

Función: Obtiene los datos correspondientes a uno o todos los clientes.

Funcionamiento:

- Se comprobará si el valor del parámetro de entrada es nulo, en cuyo caso se seleccionarán todos los clientes.
- Si el valor del parámetro de entrada es distinto de nulo, seleccionaremos el registro cuyo identificador de cliente sea igual al dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador no existe, devolverá un cursor vacío.

Usos: Ninguno.

Parámetros de entrada:

- pld: identificador del cliente. Podrá tener valor nulo para obtener todos los clientes.

Parámetros de salida:

- rs: cursor que podrá contener:
 - o Cursor vacío si no existe ningún cliente o el *idCliente* introducido no existe.
 - o Cursor con un único cliente con *idCliente* igual al introducido como parámetro.
 - o Cursor con todos los clientes si el valor del parámetro *idCliente* es nulo.

5.1.2.6.2 Productos

Los casos de uso para los Productos son:

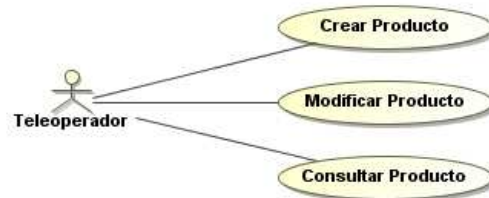


Figura 14: Casos de uso del Subsistema Petición: Productos.

5.1.2.6.2.1 CREAR PRODUCTO (INS_PRODUCTO)

Id: RF50

Función: Este procedimiento inserta un nuevo producto en el sistema.

Funcionamiento:

- Se introduce el nuevo producto en el sistema.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- En caso de existir un producto con el mismo número de serie se generará un error (*ERROR: El número de serie del producto ya existe*).
- Los campos marca, modelo, número de serie, fecha de compra, dirección y código postal no pueden tener valor nulo. En caso de que alguno de ellos tenga valor nulo, se generará el error (*ERROR: Se ha intentado almacenar un valor nulo*).
- El identificador del tipo de electrodoméstico debe existir en el sistema. En caso de no existir, se generará el error (*ERROR: El identificador de electrodoméstico no existe*).
- El electrodoméstico debe estar de alta. En caso contrario, se generará el error (*ERROR: El electrodoméstico está de baja*).
- El código postal de la población debe existir en el sistema. En caso de no existir, se generará el error (*ERROR: El código postal no existe*).
- El identificador de cliente debe existir en el sistema. En caso de no existir, se generará el error (*ERROR: El identificador de cliente no existe*).

Usos: GRABA_LOG.

Parámetros de entrada:

- pIdElectrodomestico: identificador del tipo de electrodoméstico del producto.
- pMarca: marca del producto.
- pModelo: modelo del producto.
- pNumSerie: número de serie del producto.
- pFxCompra: fecha de adquisición del producto.
- pDirección: dirección en la que se encuentra el producto
- pCodPostal: código postal de la población en la que se encuentra el producto.
- pIdCliente: identificador del propietario del producto.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.6.2.2 **MODIFICAR PRODUCTO (UPD_PRODUCTO)**

Id: RF51

Función: Cambiamos los datos (identificador de electrodoméstico, marca, modelo, número de serie, fecha de compra, dirección, código postal, identificador del cliente) del producto.

Funcionamiento:

- Se modifican los campos de la tabla productos (*idElectrodomestico, marca, modelo, número de serie, fecha de compra, dirección, código postal e idCliente*) de aquel registro que tenga el *idProducto* dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador de producto no existe se generará un error (*ERROR: El identificador del producto no existe*).
- En caso de existir un registro con el mismo número de serie, se generará un error (*ERROR: El número de serie del producto ya existe*).
- Los campos marca, modelo, numSerie, fxCompra, dirección y código postal no pueden tener valor nulo. En caso de valor nulo se generará el error (*ERROR: Se ha intentado actualizar a un valor nulo*).
- El identificador del tipo de electrodoméstico debe existir en el sistema. En caso de no existir, se generará el error (*ERROR: El identificador de electrodoméstico no existe*).
- El electrodoméstico debe estar de alta. En caso contrario, se generará el error (*ERROR: El electrodoméstico está de baja*).
- El código postal de la población debe existir en el sistema. En caso de no existir, se generará el error (*ERROR: El código postal no existe*).
- El identificador de cliente debe existir en el sistema. En caso de no existir, se generará el error (*ERROR: El identificador de cliente no existe*).

Usos: GRABA_LOG.

Parámetros de entrada:

- pid: identificador del producto.
- pIdElectrodomestico: identificador del tipo de electrodoméstico del producto.
- pMarca: marca del producto.
- pModelo: modelo del producto.
- pNumSerie: número de serie del producto.

- pFxCompra: fecha de adquisición del producto.
- pDirección: dirección en la que se encuentra el producto
- pCodPostal: código postal de la población en la que se encuentra el producto.
- pIdCliente: identificador del propietario del producto.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.6.2.3 **CONSULTAR PRODUCTO (GET_PRODUCTO)**

Id: RF52

Función: Obtiene los datos correspondientes a uno o todos los productos.

Funcionamiento:

- Se comprobará si el valor del parámetro de entrada es nulo, en cuyo caso se seleccionarán todos los productos.
- Si el valor del parámetro de entrada es distinto de nulo, seleccionaremos el registro cuyo identificador de producto sea igual al dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador no existe, devolverá un cursor vacío.

Usos: Ninguno.

Parámetros de entrada:

- pId: identificador del producto. Podrá tener valor nulo para obtener todos los productos.

Parámetros de salida:

- rs: cursor que podrá contener:
 - o Cursor vacío si no existe ningún producto o el *idProducto* introducido no existe.
 - o Cursor con un único producto con *idProducto* igual al introducido como parámetro.
 - o Cursor con todos los productos si el valor del parámetro *idProducto* es nulo.

5.1.2.6.3 **Peticiones de Servicio**

Los casos de uso para las Peticiones de Servicio son:

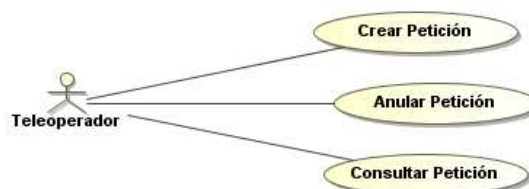


Figura 15: Casos de uso del Subsistema Petición: Peticiones de Servicio.

5.1.2.6.3.1 **CREAR PETICIÓN (INS_PETICION)**

Id: RF53

Función: Este procedimiento inserta una nueva petición en el sistema.

Funcionamiento:

- Se introduce la nueva petición de servicio en el sistema.
- El estado de la petición será *PENDIENTE*.
- La fecha de reclamación corresponderá a la fecha del sistema.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- El identificador del producto debe existir en el sistema. En caso de no existir, se generará el error (*ERROR: El identificador de producto no existe*).
- El identificador del servicio debe existir en el sistema. En caso de no existir, se generará el error (*ERROR: El identificador de servicio no existe*).
- El servicio debe estar de alta. En caso contrario, se generará el error (*ERROR: El servicio está de baja*).
- El identificador del teleoperador debe existir en el sistema. En caso de no existir, se generará el error (*ERROR: El identificador del teleoperador no existe*).
- El teleoperador debe estar de alta. En caso de no existir, se generará el error (*ERROR: El teleoperador está de baja*).

Usos: GRABA_LOG.

Parámetros de entrada:

- pIdProducto: identificador del producto que solicita la petición de servicio.
- pIdServicio: identificador del servicio solicitado en la petición.
- pIdTeleoperador: identificador del teleoperador que realiza la grabación de la petición.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.6.3.2 **ANULAR PETICION (CANCEL_PETICION)**

Id: RF54

Función: Se anula la petición de servicio. Una petición quedará anulada siempre que se encuentre en estado *PENDIENTE*. La anulación conllevará un cambio de estado a *ANULADA* y con *fxCierre* la fecha del sistema.

Funcionamiento:

- Para el identificador de petición dada, modificamos el identificador de estado a *ANULADA* y la fecha de cierre con el valor de la fecha del sistema..
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- En caso de no existir el identificador de la petición se generará un error (*ERROR: El identificador de la petición no existe*).
- Se generará un error si la petición **no** se encuentra en estado pendiente (*ERROR: La petición no se encuentra en estado PENDIENTE*).

Usos: GRABA_LOG.

Parámetros de entrada:

- pld: identificador de la petición.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.6.3.3 CONSULTAR PETICION (GET_PETICION)

Id: RF55

Función: Obtiene los datos correspondientes a una o todas las peticiones.

Funcionamiento:

- Se comprobará si el valor del parámetro de entrada es nulo, en cuyo caso se seleccionarán todas las peticiones.
- Si el valor del parámetro de entrada es distinto de nulo, seleccionaremos el registro cuyo identificador de petición sea igual al dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador no existe, devolverá un cursor vacío.

Usos: Ninguno.

Parámetros de entrada:

- pld: identificador de la petición. Podrá tener valor nulo para obtener todas las peticiones.

Parámetros de salida:

- rs: cursor que podrá contener:
 - o Cursor vacío si no existe ninguna petición o el *idPetición* introducido no existe.
 - o Cursor con una única petición con *idPetición* igual al introducido como parámetro.
 - o Cursor con todas las peticiones si el valor del parámetro *idPetición* es nulo.

5.1.2.7 Subsistema Atención

5.1.2.7.1 Búsquedas y asignaciones

Los casos de uso para las búsquedas y asignaciones son:



Figura 16: Casos de uso del Subsistema Atención: Búsquedas y Asignaciones.

5.1.2.7.1.1 **PETICIONES POR SERVICIO Y ZONA (GET_PETICIONESSERVICIOZONA)**

Id: RF56

Función: Obtiene todas las peticiones pendientes de un servicio o todos y de una zona dada o todas.

Funcionamiento:

- Se comprobará si los valores de los parámetros de entrada son nulos, en cuyo caso se seleccionarán todas las peticiones.
- Si el valor del parámetro de entrada *idServicio* es distinto de nulo, seleccionaremos el registro cuyo identificador de petición sea igual al dado.
- Si el valor del parámetro de entrada *idZona* es distinto de nulo, seleccionaremos el registro cuyo identificador de zona sea igual al dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si alguno de los identificadores no existe, devolverá un cursor vacío.

Usos: Ninguno.

Parámetros de entrada:

- *pldServicio*: identificador del servicio. Podrá tener valor nulo para obtener las peticiones de todos los servicios.
- *pldZona* : identificador de la zona. Podrá tener valor nulo para obtener las peticiones de todas las zonas.

Parámetros de salida:

- *rs*: cursor que podrá contener:
 - Cursor vacío si no existe ninguna petición o el *idServicio* o el *idZona* introducidos no existen.
 - Cursor con las peticiones de todos los servicios y todas las zonas si valor de los parámetros *idServicio* e *idZona* son nulos.
 - Cursor con las peticiones de todos los servicios de una zona dada, si el valor de *idServicio* es nulo.
 - Cursor con las peticiones de todas las zonas de un servicio dado, si el valor de *idZona* es nulo.

5.1.2.7.1.2 **PETICIONES POR TÉCNICO (GET_PETICIONESTECNICO)**

Id: RF57

Función: Obtiene todas las peticiones pendientes para un técnico dado. Si el identificador dado es nulo, se devolverán todas las peticiones.

Funcionamiento:

- Se comprobará si el valor del parámetro de entrada es nulo, en cuyo caso se seleccionarán todas las peticiones.
- Seleccionaremos aquellas peticiones que pertenezcan a la zona, que sean del servicio y de los electrodomésticos que puede atender el técnico dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador de técnico no existe, devolverá un cursor vacío.

Usos: Ninguno.

Parámetros de entrada:

- pIdTecnico: identificador de técnico. Valor nulo para obtener todas las peticiones.

Parámetros de salida:

- rs: cursor que podrá contener:
 - Cursor vacío si no existe ninguna petición o el *idTecnico* introducido no existe.
 - Cursor con las peticiones de la zona, tipo de electrodoméstico y servicio atendido por el técnico con *idTecnico* igual al introducido como parámetro.
 - Cursor con todas las peticiones si el valor del parámetro *idTecnico* es nulo.

5.1.2.7.1.3 **ASIGNAR TÉCNICO (ASG_TECNICO)**

Id: RF58

Función: Asignará un técnico a una petición de servicio pendiente.

Funcionamiento:

- Se realiza la asociación grabado el *idTecnico* en la tabla PETICIONES y modificando el estado a ASIGNADA (T).
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si la petición no existe se generará un error (*ERROR: El identificador de petición no existe*).
- Si la petición se encuentra en un estado diferente a PENDIENTE, se generará un error (*ERROR: La petición no se encuentra en estado PENDIENTE*).
- Si el identificador del técnico no existe se generará un error (*ERROR: El identificador de técnico no existe*).
- Si el técnico se encuentra de baja se generará el error (*ERROR: El técnico está de baja*).

Usos: GRABA_LOG.

Parámetros de entrada:

- pIdPetición: identificador de la petición de servicio.
- pIdTecnico: identificador del técnico.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.7.1.4 **DESASIGNAR TÉCNICO (DES_TECNICO)**

Id: RF59

Función: Desasignará la petición del técnico asociado.

Funcionamiento:

- Se pondrá con valor nulo el campo *idTecnico* y el estado pasará a valor PENDIENTE (P).
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si la petición no existe se generará un error (*ERROR: El identificador de petición no existe*).
- Si la petición se encuentra en un estado diferente a ASIGNADA, se generará un error (*ERROR: La petición no se encuentra en estado ASIGNADA*).

Usos: GRABA_LOG.

Parámetros de entrada:

- pldPetición: identificador de la petición de servicio.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.7.1.5 CONSULTAR TECNICO ASIGNADO (GET_TECNICOASIGNADO)

Id: RF60

Función: Obtiene el técnico asignado a una petición.

Funcionamiento:

- Se seleccionan los datos del técnico que está asociado a la petición dada.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador no existe, devolverá un cursor vacío.
- Si la petición no está asignada, devolverá un cursor vacío.

Usos: Ninguno.

Parámetros de entrada:

- pld: identificador de la petición.

Parámetros de salida:

- rs: cursor que podrá contener:
 - o Cursor vacío si el *idPetición* introducido no existe.
 - o Cursor con los datos del técnico asociado al *idPetición* dado.
 - o Cursor vacío si la petición no se encuentra asignada.

5.1.2.7.2 Trabajos

Los casos de uso para los trabajos son:

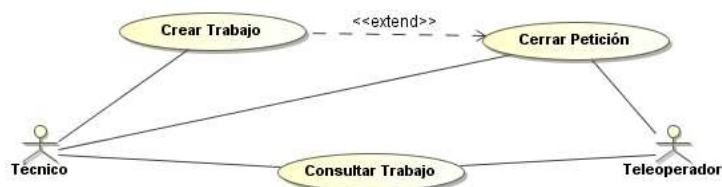


Figura 17: Casos de uso del Subsistema Atención: Trabajos.

5.1.2.7.2.1 **CREAR TRABAJO (INS_TRABAJO)**

Id: RF61

Función: Este procedimiento inserta los datos del trabajo realizado en el sistema.

Funcionamiento:

- Se introduce el nuevo trabajo en el sistema.
- La fecha del trabajo corresponderá a la fecha del sistema.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Los campos dsTrabajo, importe y garantía no pueden tener valor nulo. En caso de que alguno de ellos tenga valor nulo, se generará el error (*ERROR: Se ha intentado almacenar un valor nulo*).
- El identificador de petición debe existir en el sistema. En caso de no existir, se generará el error (*ERROR: El identificador de petición no existe*).
- La petición debe encontrarse en estado ASIGNADA. En caso contrario, se generará el error (*ERROR: La petición no se encuentra en estado ASIGNADA*).
- La petición debe encontrarse asignada al técnico que realiza el trabajo. En caso contrario, se generará el error (*ERROR: La petición está asignada a otro técnico*).
- El identificador del técnico debe existir en el sistema. En caso de no existir, se generará el error (*ERROR: El identificador de técnico no existe*).
- El técnico no puede estar de baja. En caso contrario, se generará el error (*ERROR: El técnico está de baja*).

Usos: GRABA_LOG.

Parámetros de entrada:

- pIdPetición: identificador de la petición sobre la que se realiza el trabajo.
- pIdTecnico: identificador del técnico que realiza el trabajo.
- pDsTrabajo: descripción del trabajo realizado.
- pImporte: importe del trabajo realizado.
- pGarantia: indica con un valor 0 o 1 si la garantía ampara el trabajo.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.7.2.2 **CONSULTAR TRABAJO (GET_TRABAJO)**

Id: RF62

Función: Obtiene los datos correspondientes a uno o todos los trabajos.

Funcionamiento:

- Se comprobará si el valor del parámetro de entrada es nulo, en cuyo caso se seleccionarán todos los trabajos.
- Si el valor del parámetro de entrada es distinto de nulo, seleccionaremos el registro cuyo identificador de trabajo sea igual al dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador no existe, devolverá un cursor vacío.

Usos: Ninguno.

Parámetros de entrada:

- pld: identificador del trabajo. Podrá tener valor nulo para obtener todos los trabajos.

Parámetros de salida:

- rs: cursor que podrá contener:
 - Cursor vacío si no existe ningún trabajo o el *idTrabajo* introducido no existe.
 - Cursor con un único trabajo con *idTrabajo* igual al introducido como parámetro.
 - Cursor con todos los trabajos si el valor del parámetro *idTrabajo* es nulo.

5.1.2.7.2.3 CERRAR PETICION (CLOSE_PETICION)

Id: RF63

Función: Este procedimiento cierra la petición de servicio.

Funcionamiento:

- Se modifica la fecha de cierre y se cambia el estado a CERRADA.
- La fecha de cierre corresponderá a la fecha del sistema.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- El identificador de petición debe existir en el sistema. En caso de no existir, se generará el error (*ERROR: El identificador de petición no existe*).
- La petición debe encontrarse en estado ASIGNADA. En caso contrario, se generará el error (*ERROR: La petición no se encuentra en estado ASIGNADA*).
- La petición debe tener como mínimo un trabajo asociado. En caso contrario, se generará el error (*ERROR: La petición no tiene trabajos*).

Usos: GRABA_LOG.

Parámetros de entrada:

- pldPetición: identificador de la petición que desea cerrar.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.7.3 Encuesta

Los casos de uso para las encuestas son:

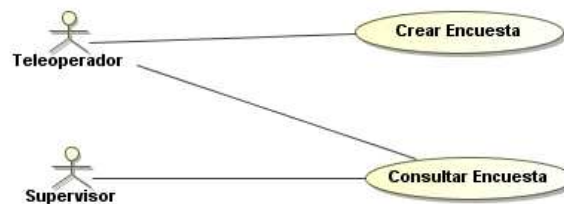


Figura 18: Casos de uso del Subsistema Atención: Encuesta.

5.1.2.7.3.1 **CREAR ENCUESTA (INS_ENCUESTA)**

Id: RF64

Función: Este procedimiento inserta los datos de la encuesta de satisfacción en el sistema.

Funcionamiento:

- Se introduce la encuesta en el sistema.
- La fecha de la encuesta corresponderá a la fecha del sistema.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- El identificador de petición debe existir en el sistema. En caso de no existir, se generará el error (*ERROR: El identificador de petición no existe*).
- La petición debe encontrarse en estado CERRADA. En caso contrario, se generará el error (*ERROR: La petición no se encuentra en estado CERRADA*).
- El identificador del teleoperador debe existir en el sistema. En caso de no existir, se generará el error (*ERROR: El identificador de teleoperador no existe*).
- El teleoperador no puede estar de baja. En caso contrario, se generará el error (*ERROR: El teleoperador está de baja*).

Usos: GRABA_LOG.

Parámetros de entrada:

- pIdPetición: identificador de la petición sobre la que se realiza la encuesta.
- pIdTeleoperador: identificador del teleoperador que realiza la encuesta.
- pValoración: valoración de la satisfacción del cliente. Valor entre 0 y 9.
- pComentario: comentario realizados por el cliente sobre el servicio recibido y/o mejora.

Parámetros de salida:

- resultado: contendrá información de la finalización del procedimiento.

5.1.2.7.3.2 **CONSULTAR ENCUESTA (GET_ENCUESTA)**

Id: RF65

Función: Obtiene los datos correspondientes a una o todas las encuestas.

Funcionamiento:

- Se comprobará si el valor del parámetro de entrada es nulo, en cuyo caso se seleccionarán todas las encuestas.
- Si el valor del parámetro de entrada es distinto de nulo, seleccionaremos el registro cuyo identificador de encuesta sea igual al dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador no existe, devolverá un cursor vacío.

Usos: Ninguno.

Parámetros de entrada:

- pId: identificador de la encuesta. Podrá tener valor nulo para obtener todas las encuestas.

Parámetros de salida:

- rs: cursor que podrá contener:
 - o Cursor vacío si no existe ninguna encuesta o el *idEncuesta* introducido no existe.
 - o Cursor con una única encuesta con *idEncuesta* igual al introducido como parámetro.
 - o Cursor con todas las encuestas si el valor del parámetro *idEncuesta* es nulo.

5.1.2.8 Subsistema Histórico

5.1.2.8.1 Histórico

Los casos de uso para los históricos son:

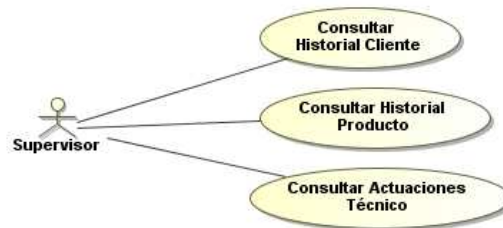


Figura 19: Casos de uso del Subsistema Histórico: Históricos.

5.1.2.8.1.1 CONSULTAR HISTORIAL CLIENTE (GET_HCLIENTE)

Id: RF66

Función: Obtiene todas las peticiones solicitadas por un cliente.

Funcionamiento:

- Seleccionaremos aquellas peticiones de los productos cuyo identificador de cliente es igual al cliente dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador de cliente no existe, devolverá un cursor vacío.

Usos: Ninguno.

Parámetros de entrada:

- pIdCliente: identificador del cliente.

Parámetros de salida:

- rs: cursor que podrá contener:
 - o Cursor vacío si no existe ninguna petición del cliente o el *idCliente* introducido no existe.
 - o Cursor con las peticiones registradas en los productos del cliente.

5.1.2.8.1.2 **CONSULTAR HISTORIAL PRODUCTO (GET_HPRODUCTO)**

Id: RF67

Función: Obtiene todas las peticiones solicitadas sobre un producto.

Funcionamiento:

- Seleccionaremos aquellas peticiones del producto solicitado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador de producto no existe, devolverá un cursor vacío.

Usos: Ninguno.

Parámetros de entrada:

- pIdProducto: identificador del producto.

Parámetros de salida:

- rs: cursor que podrá contener:
 - Cursor vacío si no existe ninguna petición del producto o el *idProducto* introducido no existe.
 - Cursor con las peticiones registradas para el producto.

5.1.2.8.1.3 **CONSULTAR HISTORIAL TECNICO (GET_HTECNICO)**

Id: RF68

Función: Obtiene todos los trabajos realizados por un técnico.

Funcionamiento:

- Seleccionaremos aquellos trabajos cuyo identificador de técnico es igual al técnico dado.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si el identificador de técnico no existe, devolverá un cursor vacío.

Usos: Ninguno.

Parámetros de entrada:

- pIdTecnico: identificador del técnico.

Parámetros de salida:

- rs: cursor que podrá contener:
 - Cursor vacío si no existe ningún trabajo del técnico o el *idTecnico* introducido no existe.
 - Cursor con los trabajos realizados por el técnico.

5.1.2.9 Subsistema Control

5.1.2.9.1 Log

Los casos de uso para los ficheros de log son:

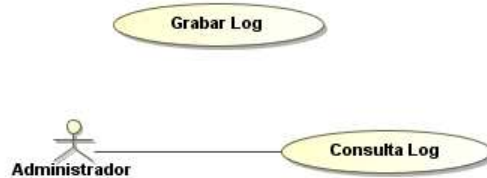


Figura 20: Casos de uso del Subsistema Control: Log.

5.1.2.9.1.1 GRABAR LOG (GRABA_LOG)

Id: RF69

Función: Graba en la tabla correspondiente (LOG) los datos correspondientes a la ejecución de un procedimiento (procedimiento, parámetros, fecha y hora de entrada y salida y resultado). Este procedimiento se declara como una transacción autónoma (PRAGMA AUTONOMOUS_TRANSACTION) para que el COMMIT efectuado no afecte al resto de transacciones.

Funcionamiento:

- Grabamos en la tabla LOG los datos correspondientes a la ejecución de un procedimiento.

Usos: ninguno

Parámetros de entrada:

- procedimiento: nombre del procedimiento ejecutado.
- paramIN: contendrá los parámetros de entrada del procedimiento.
- paramOUT: tiene la información sobre la finalización del procedimiento. Tendrá el siguiente formato:
- fxInicio: fecha de entrada en el procedimiento.
- fxFin: fecha de salida del procedimiento.

Parámetros de salida: ninguno

5.1.2.9.1.2 CONSULTA LOG (GET_LOG)

Id: RF70

Función: Retorna de la tabla correspondiente (LOG) todos los datos correspondientes a la ejecución de procedimientos (procedimiento, parámetros, fecha y hora de entrada y salida, parámetros y resultado) entre las fechas solicitadas.

Funcionamiento:

- Seleccionaremos aquellos movimientos de Log realizados entre las fechas dadas.
- La fecha de inicio debe ser mayor que la fecha final.
- Grabamos en el Log OK si el procedimiento se ha completado con éxito o el ERROR ocurrido (GRABA_LOG).

Restricciones:

- Si la fecha de inicio es mayor que la fecha de fin se generará el error (*ERROR: La fecha de inicio es mayor que la fecha de fin*).
- Si no hay movimientos entre las fechas se devolverá un cursor vacío.

Usos: Ninguno.

Parámetros de entrada:

- pFxInicio: fecha de inicio.
- pFxFin: fecha de fin.

Parámetros de salida:

- rs: cursor que podrá contener:
 - o Cursor con los atributos del log generado entre fechas.
 - o Cursor vacío si no existe ningún movimiento entre fechas.

5.1.2.10 Subsistema Estadísticas

5.1.2.10.1 Estadísticas

Estos requerimientos serán realizados a través de herramientas OLAP (IBM Cognos).

5.1.2.10.1.1 **PETICIONES**

Obtendremos informes relacionados con las peticiones de servicio del sistema aplicando distintas perspectivas.

- Temporal.
- Localización.
- Servicio solicitado.
- Electrodomésticos.

Algunos ejemplos de peticiones de servicio en un periodo temporal podrían ser:

- Número de peticiones registradas.
- Importe de las peticiones de servicio.
- Número de peticiones por servicio y electrodoméstico.
- Número de peticiones por servicio y zona.
- Valoración de las peticiones de servicio en garantía.
- Grado de Satisfacción de clientes.

5.1.2.10.1.2 **TELEOPERADORES**

Obtendremos informes relacionados con el trabajo realizado por los teleoperadores aplicando distintas perspectivas.

- Temporal.
- Localización.
- Tipo de petición de servicio.
- Electrodomésticos.

Algunos ejemplos podrían ser:

- Número de peticiones de servicio registradas.
- Número de encuestas de satisfacción realizadas.

5.1.2.10.1.3 **TÉCNICOS**

Obtendremos informes relacionados con el trabajo realizado por los técnicos aplicando distintas perspectivas.

- Temporal.
- Tipo de petición de servicio.
- Electrodomésticos.

Algunos ejemplos podrían ser:

- Número de trabajos realizados.
- Número de trabajos realizados por tipo.
- Número de trabajos realizados por electrodoméstico.
- Importe de los trabajos realizados.
- Valoración de los trabajos en garantía.

6 Diseño Base de Datos

6.1 Diseño Conceptual

6.1.1 Diagrama Entidad-Relación

En el siguiente diagrama se representan las entidades de la base de datos operacional y las relaciones existentes entre ellas.

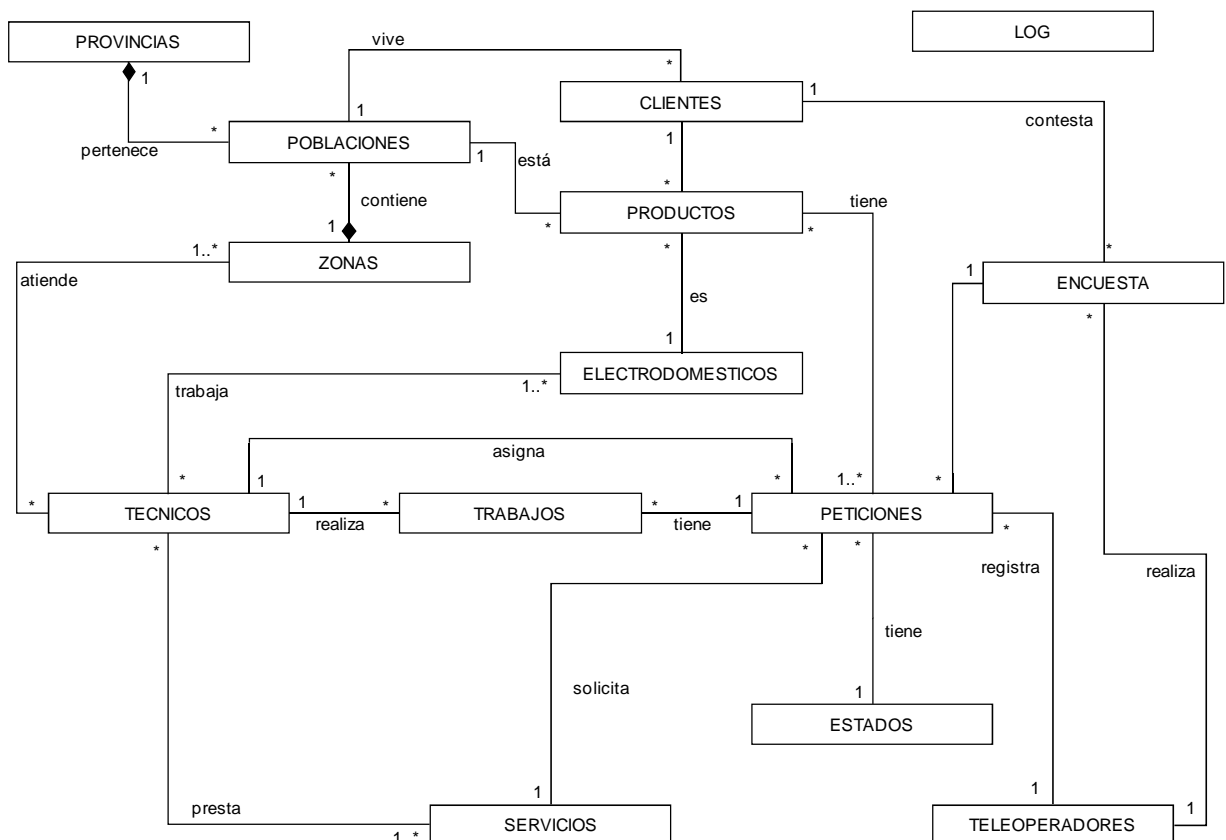


Figura 21: Diagrama Entidad-Relación.

6.1.2 Entidades

Veamos cada una de las entidades representadas en el diagrama anterior.

CLIENTES

En esta entidad se registrarán los datos de identificación y localización del cliente.

ELECTRODOMESTICOS

Contendrá los tipos de electrodomésticos que contempla el sistema.

ENCUESTAS

Esta entidad contendrá una valoración de la satisfacción del cliente con respecto al servicio solicitado.

ESTADOS

Contendrá los diferentes estados en los que puede encontrarse una petición.

LOG

Registrará todos los movimientos que se realizan en el sistema. Identificando al procedimiento, los parámetros, la fecha en que se realiza la acción, la fecha de finalización y el resultado de la operación.

PETICIONES

Esta entidad almacenará los datos de las peticiones de servicio que realizan los clientes sobre los productos.

POBLACIONES

Contendrá las poblaciones almacenadas en el sistema. Las poblaciones nos servirán para establecer las distintas zonas y poder ubicar tanto a los clientes como a sus productos.

PRODUCTOS

Esta entidad almacenará los datos de los productos propiedad de clientes sobre los que se ha solicitado algún servicio. Se deberá especificar la fecha de compra del producto así como su ubicación.

PROVINCIAS

Contendrá las provincias almacenadas en el sistema. Una provincia estará formada por una o varias poblaciones.

SERVICIOS

La entidad Servicios almacenará los distintos tipos de servicio que la empresa ofrece.

TECNICOS

En esta entidad se registrarán los datos de los técnicos que realizarán los servicios ofrecidos por la empresa y solicitados por los clientes. Estarán especializados en uno o varios electrodomésticos, serán capaces de atender uno o más tipos de servicios y todo esto para una o más zonas.

TELEOPERADORES

Contiene la información de los teleoperadores que atenderán a los clientes, registrarán las peticiones de servicio y realizarán las encuestas de satisfacción.

TRABAJOS

La entidad Trabajos almacena todos los trabajos que se realizan los técnicos para atender las peticiones de servicio de los clientes. Será cada trabajo el que contendrá información de la valoración del servicio realizado y si este entra o no en la garantía del producto.

ZONAS

Contendrá las zonas almacenadas en el sistema. Una zona estará formada por una o varias poblaciones. Las zonas indicarán una distribución del área geográfica en la que actuarán determinados técnicos.

6.1.3 Atributos

Vamos a mostrar los atributos que tendrán las entidades del sistema

La clave primaria de cada entidad se muestra subrayada y la **clave alternativa** se muestra en negrita.

CLIENTES

idCliente, nombre, apellidos, **nif**, direccion, telFijo, telMovil, eMail, fxAlta

ELECTRODOMESTICOS

idElectrodomestico, **dsElectrodomestico**, fxAlta, fxBaja

ENCUESTAS

idEncuesta, fxEncuesta, valoracion, comentario

ESTADOS

idEstado, **dsEstado**

LOG

idLog, procedimiento, paramIN, paramOUT, fxIN, fxOUT, resultado

PETICIONES

idPetition, fxReclamacion, fxCierre

POBLACIONES

codPostal, dsPoblacion

PRODUCTOS

idProducto, marca, modelo, **numSerie**, fxCompra, direccion

PROVINCIAS

coProvincia, **dsProvincia**

SERVICIOS

idServicio, **dsServicio**, fxAlta, fxBaja

TELEOPERADORES

idTeleoperador, nombre, apellidos, **nif**, fxAlta, fxBaja

TECNICOS

idTecnico, nombre, apellidos, **nif**, telMovil, fxAlta, fxBaja

TRABAJOS

idTrabajo, fxTrabajo, dsTrabajo, importe, garantia

ZONAS

idZona, **dsZona**

6.2 Diseño Lógico

6.2.1 Modelo Lógico

Mediante la herramienta Toad Data Modeler versión 3.4 creamos el siguiente modelo lógico.

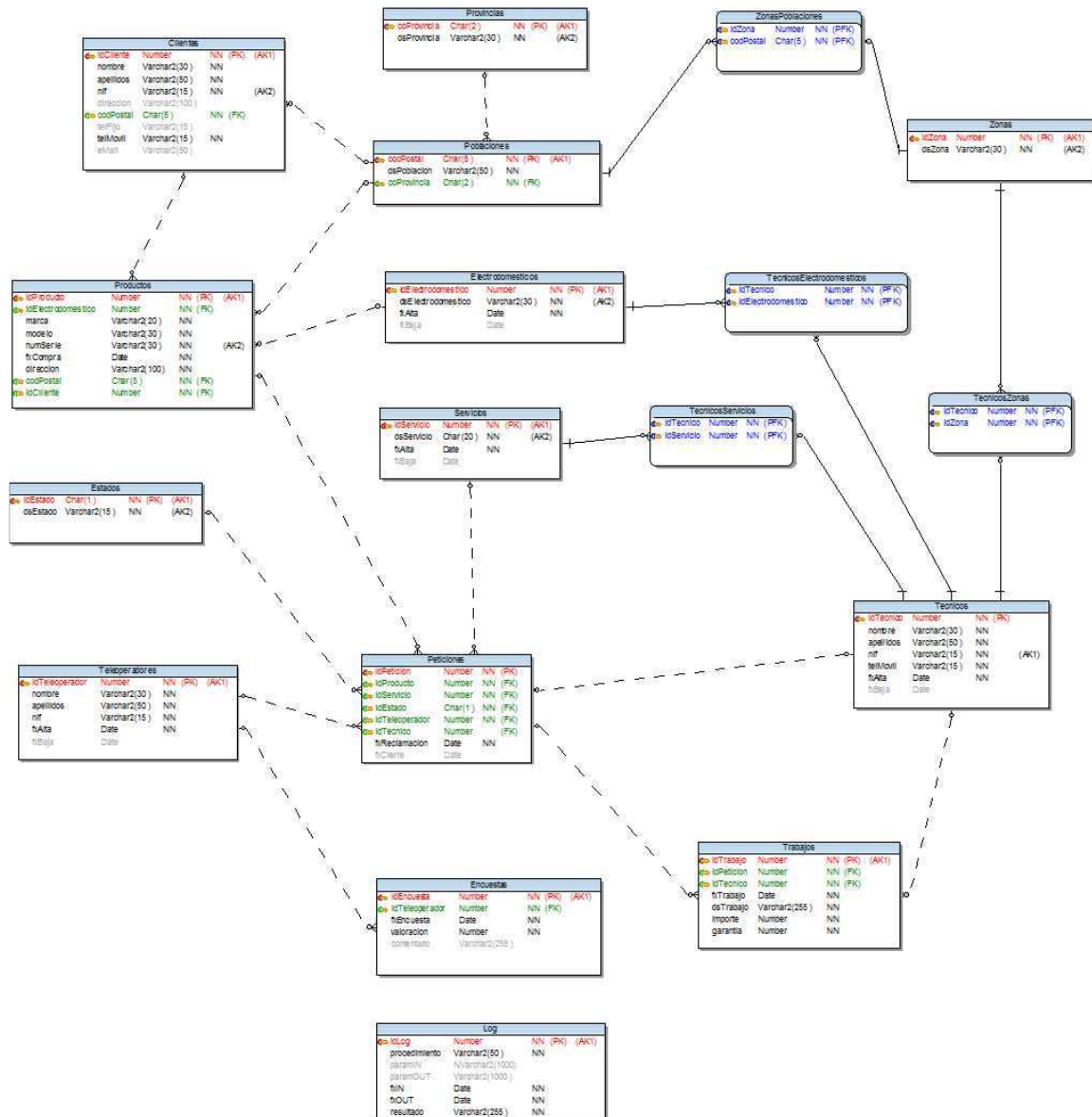


Figura 22: Modelo lógico Base de Datos.

Las siglas utilizadas en el diagrama anterior son:

- **PK**: Clave Primaria
- **AKn**: Clave única (alternativa)
- **FK**: Clave Foránea
- **PFK**: Clave Primaria y Clave Foránea
- **NN**: El atributo de la entidad no admite valores nulos.

6.2.2 Entidades

A continuación se detallan los campos de las tablas, índices y claves foráneas.

Claves principales subrayadas, **claves alternativas** en negrita y valores *nulos no admitidos* en cursiva.

CLIENTES

(idCliente, *nombre*, *apellidos*, **nif**, *direccion*, *codPostal*, *telFijo*, *telMovil*, *eMail*, *fxAlta*)

[nif] clave única

[codPostal] clave foránea Poblaciones(codPostal)

Valores por defecto:

[fxAlta] fecha del sistema.

ELECTRODOMESTICOS

(idElectrodomestico, **dsElectrodomestico**, *fxAlta*, *fxBaja*)

[dsElectrodomestico] clave única

Valores por defecto:

[fxAlta] fecha del sistema.

ENCUESTAS

(idEncuesta, *idPetición*, *idTeleoperador*, *fxEncuesta*, *valoracion*, *comentario*)

[idPetición] clave foránea Peticiones(idPetición)

[idTeleoperador] clave foránea Teleoperadores(idTeleoperador)

Valores por defecto:

[fxEncuesta] fecha del sistema.

Restricciones:

(valoración: valores comprendidos entre 0 y 9)

ESTADOS

(idEstado, **dsEstado**)

[dsEstado] clave única

LOG

(idLog, *procedimiento*, *paramIN*, *paramOUT*, *fxIN*, *fxOUT*, *resultado*)

PETICIONES

(idPetición, *idProducto*, *idServicio*, *idEstado*, *idTeleoperador*, *idTecnico*, *fxReclamacion*, *fxCierre*)

[idProducto] clave foránea Productos(idProducto)

[idServicio] clave foránea Servicios(idServicio)

[idEstado] clave foránea Estados(idEstado)

[idTeleoperador] clave foránea Teleoperadores(idTeleoperador)

[idTecnico] clave foránea Tecnicos(idTecnico)

Valores por defecto:

[fxReclamacion] fecha del sistema.

POBLACIONES

(codPostal, **dsPoblacion**, *coProvincia*)

[coProvincia] clave foránea Provincias(coProvincia)

PRODUCTOS

(idProducto, *idElectrodomestico*, *marca*, *modelo*, **numSerie**, *fxCompra*, *direccion*, *codPostal*, *idCliente*)

[idElectrodomestico] clave foránea Electrodomesticos(idElectrodomestico)

[numSerie] clave única

[codPostal] clave foránea Poblaciones(codPostal)
[idCliente] clave foránea Clientes(idCliente)

PROVINCIAS

(coProvincia, **dsProvincia**)
[dsProvincia] clave única

SERVICIOS

(idServicio, **dsServicio**, *fxAlta*, *fxBaja*)
[dsServicio] clave única
Valores por defecto:
[fxAlta] fecha del sistema.

TECNICOS

(idTecnico, *nombre*, *apellidos*, **nif**, *telMovil*, *fxAlta*, *fxBaja*)
[nif] clave única
Valores por defecto:
[fxAlta] fecha del sistema.

TECNICOSELECTRODOMESTICOS

(idTecnico, idElectrodomestico)
[idTecnico] clave foránea Tecnicos(idTecnico)
[idElectrodomestico] clave foránea Electrodomesticos(idElectrodomestico)

TECNICOSSERVICIOS

(idTecnico, idServicio)
[idTecnico] clave foránea Tecnicos(idTecnico)
[idServicio] clave foránea Servicios(idServicio)

TECNICOSZONAS

(idTecnico, idZona)
[idTecnico] clave foránea Tecnicos(idTecnico)
[idZona] clave foránea Zonas(idZona)

TELEOPERADORES

(idTeleoperador, *nombre*, *apellidos*, **nif**, *fxAlta*, *fxBaja*)
[nif] clave única
Valores por defecto:
[fxAlta] fecha del sistema.

TRABAJOS

(idTrabajo, *idPetición*, *idTecnico*, *fxTrabajo*, *dsTrabajo*, *importe*, *garantia*)
[idPetición] clave foránea Peticiones(idPetición)
[idTecnico] clave foránea Tecnicos(idTecnico)
Restricciones:
(importe >= 0)
(garantia admitirá valor 0 para el trabajo no sujeto a garantía y valor 1 cuando está en garantía)

ZONAS

(idZona, **dsZona**)
[dsZona] clave única

ZONASPOBLACIONES

(idZona, codPostal)
[idZona] clave foránea Zonas(idZona)
[codPostal] clave foránea Poblaciones(codPostal)

7 Diseño Almacén de Datos (DW)

7.1 Diseño Conceptual

Realizaremos un diseño por cada uno de las estadísticas comentadas anteriormente

7.1.1 Estadística de Peticiones

Según hemos visto en los requerimientos, los resultados serán:

- Número de peticiones.
- Importe
- Importe de garantías
- Valoración cliente

Estas estadísticas se llevarán a cabo desde los siguientes puntos de vista:

- Temporal.
- Localización.
- Servicio solicitado.
- Electrodomésticos.

El diagrama correspondiente a una estrella formada por el Hecho H_Peticiones y las dimensiones por las que se podrá realizar la selección y agrupación.



Figura 23: Estrella Estadística Peticiones.

7.1.2 Estadística de Teleoperadores

Según hemos visto en los requerimientos, los resultados serán:

- Número de peticiones de servicio registradas.
- Número de encuestas de satisfacción realizadas.

Estas estadísticas se llevarán a cabo desde los siguientes puntos de vista:

- Temporal.
- Servicio solicitado.
- Electrodomésticos.
- Teleoperadores

El diagrama correspondiente a una estrella formada por el Hecho H_Teleoperadores y las dimensiones por las que se podrá realizar la selección y agrupación.



Figura 24: Estrella Estadística Teleoperadores.

7.1.3 Estadística de Técnicos

Según hemos visto en los requerimientos, los resultados serán:

- Número de trabajos
- Importe
- Importe de garantías

Estas estadísticas se llevarán a cabo desde los siguientes puntos de vista:

- Temporal.
- Tipo de petición de servicio.
- Electrodomésticos.
- Técnicos

El diagrama correspondiente a una estrella formada por el Hecho H_Teleoperadores y las dimensiones por las que se podrá realizar la selección y agrupación.

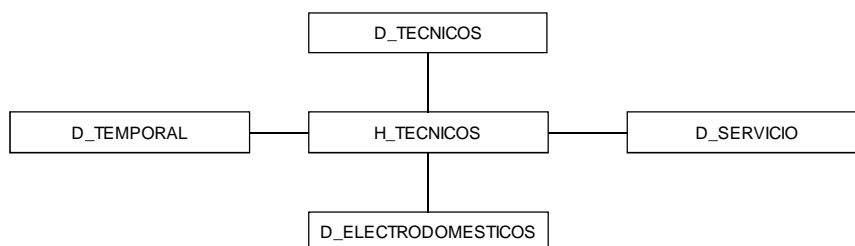


Figura 25: Estrella Estadística Técnicos.

7.1.4 Entidades

Veamos cada una de las entidades representadas en el diagrama anterior.

D ELECTRODOMESTICOS

Dimensión de tipos de electrodomésticos.

D LOCALIZACION

Dimensión de ubicaciones que nos permitirá realizar filtrados por poblaciones, provincias y zonas.

D SERVICIOS

Dimensión de tipos de servicio.

D TECNICOS

Dimensión de técnicos.

D TEMPORAL

Dimensión temporal que nos permitirá realizar filtrados por meses y años.

D TELEOPERADORES

Dimensión de teleoperadores.

H PETICIONES

Tabla de hechos de estadística de peticiones de servicio.

H TELEOPERADORES

Tabla de hechos de estadística de teleoperadores.

H TECNICOS

Tabla de hechos de estadística de técnicos.

7.1.5 Atributos

Vamos a mostrar los atributos que tendrán las entidades del sistema

D ELECTRODOMESTICOS

idElectrodomestico, dsElectrodomestico

D LOCALIZACION

idLocalizacion, dsPoblacion, dsProvincia, dsZona

D SERVICIOS

idServicio, dsServicio

D TECNICOS

idTecnico, nombre

D TEMPORAL

idTemporal, mes, año

D TELEOPERADORES

idTeleoperador, nombre

H PETICIONES

numPeticiones, importe, impGarantia, valoracionMedia

H TELEOPERADORES

numPeticiones, numEncuestas

H TECNICOS

numTrabajos, importe, impGarantia

7.2 Diseño Lógico

7.2.1 Modelo Lógico

Mediante la herramienta Toad Data Modeler versión 3.4 creamos el siguiente modelo lógico.

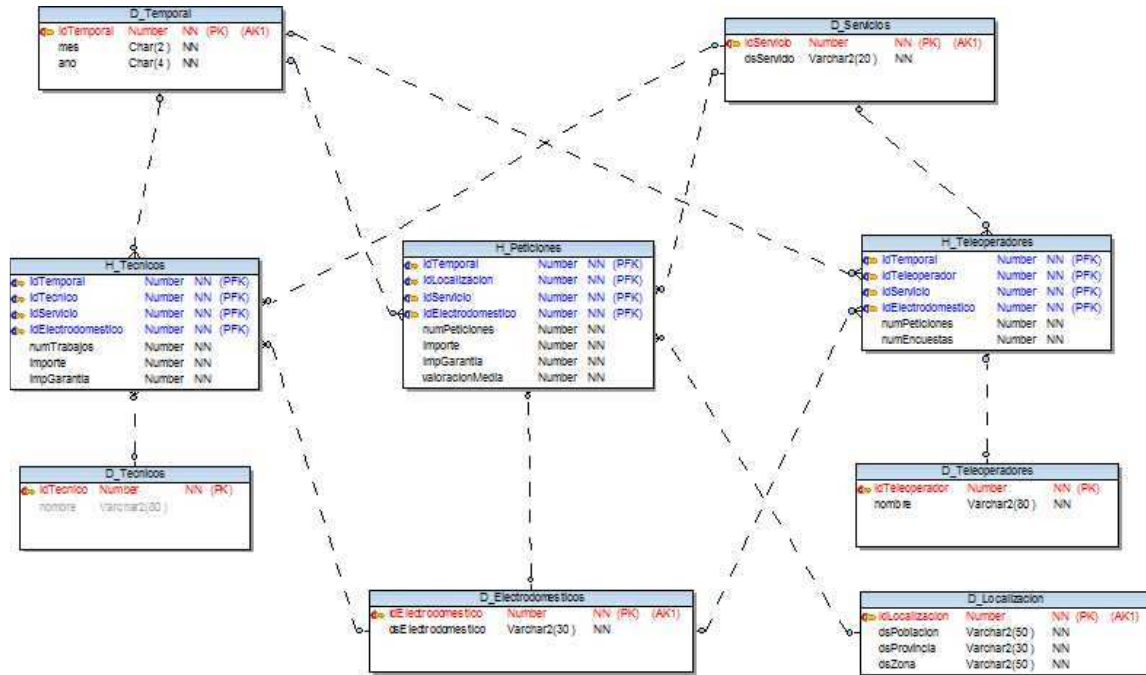


Figura 26: Modelo lógico del Almacén de Datos.

Las siglas utilizadas en el diagrama anterior son:

- **PK**: Clave Primaria
- **AKn**: Clave única (alternativa)
- **FK**: Clave Foránea
- **PFK**: Clave Primaria y Clave Foránea
- **NN**: El atributo de la entidad no admite valores nulos.

7.2.2 Entidades

A continuación se detallan los campos de las tablas, índices y claves foráneas.

Claves principales subrayadas, **claves alternativas** en negrita y valores *nulos no admitidos* en cursiva.

D_ELECTRODOMESTICOS

(idElectrodomestico, **dsElectrodomestico**)

D_LOCALIZACIONES

(idLocalizacion, *dsPoblacion*, *dsProvincia*, *dsZona*)

D_SERVICIOS

(idServicio, *dsServicio*)

D_TECNICOS

(idTecnico, *nombre*)

D_TEMPORAL

(idTemporal, *mes*, *ano*)

Mes en formato: mm

Año en formato: aaaa

D_TELEOPERADORES

(idTeleoperador, *nombre*)

H_PETICIONES

(idTemporal, idLocalizacion, idServicio, idElectrodomestico, *numPeticiones*, *importe*, *impGarantia*, *valoracionMedia*)

H_TECNICOS

(idTemporal, idTecnico, idServicio, idElectrodomestico, *numTrabajos*, *importe*, *impGarantia*)

H_TELEOPERADORES

(idTemporal, idTeleoperador, idServicio, idElectrodomestico, *numPeticiones*, *numEncuestas*)

8 PRODUCTO FINAL

El producto consta de los siguientes scripts ubicados dentro de la carpeta sql.

- *00-GPV.sql*: Crea todos los objetos del sistema, llamando al resto de scripts.
- *01-tablespaces.sql*: Crea los tablespaces.
- *02-usuarios.sql*: Crea los usuarios.
- *03-tablas_bd.sql*: Crea las tablas, índices, constraints y valores iniciales necesarios.
- *04-tablas_dw.sql*: Crea las tablas, índices, constraints y valores iniciales necesarios.
- *05-triggers_bd.sql*: Crea los triggers del esquema BD.
- *06-triggers_dw.sql*: Crea los triggers del esquema DW.
- *07-01-control.sql*: Crea el paquete con los procedimientos del subsistema control.
- *07-02-configuracion.sql*: Crea el paquete con los procedimientos del subsistema configuración.
- *07-03-personal.sql*: Crea el paquete con los procedimientos del subsistema personal.
- *07-04-peticion.sql*: Crea el paquete con los procedimientos del subsistema petición.
- *07-05-atencion.sql*: Crea el paquete con los procedimientos del subsistema atención.
- *07-06-historial.sql*: Crea el paquete con los procedimientos del subsistema historial.
- *08-etl.sql*: Crea el paquete con los procedimientos de ETL.
- *09-01-test_configuracion.sql*: Crea el paquete con los procedimientos del test de configuración.
- *09-02-test_personal.sql*: Crea el paquete con los procedimientos del test de personal.
- *09-03-test_peticion.sql*: Crea el paquete con los procedimientos del test de petición.
- *09-04-test_atencion.sql*: Crea el paquete con los procedimientos del test de atención.
- *10-carga_inicial.sql*: Crea el paquete con los procedimientos para una carga inicial del sistema.

8.1 Instalación del producto

Partimos de una instancia de Oracle Express v10 y una carpeta en C:\ llamada GPV:

Con el usuario SYSTEM o SYS ejecutamos el siguiente script: *00-GPV.sql*

Este script creará:

- Dos tablespaces de nombre BD y DW con un tamaño fijo de 100Mb ubicados en C:\GPV.
- Dos usuarios BD con contraseña BD y DW con contraseña DW que serán los propietarios de los objetos que forman el sistema (BD objetos de la Base de Datos Operacional y DW del Almacén de Datos).
- Objetos necesarios para nuestro sistema: tablas, índices, packages, procedimientos, triggers, etc. Así como rellenar las tablas necesarias con valores iniciales.

8.2 Plan de pruebas

Para poder determinar el correcto funcionamiento del sistema, se realizarán pruebas unitarias de cada uno de los subsistemas.

En el documento Anexo1 - Test del Sistema se podrá observar la ejecución y resultado de las pruebas realizadas.

8.2.1 Subsistema Configuración

8.2.1.1 TEST SERVICIOS

Id: TEST01

Pruebas realizadas:

INS_SERVICIO

- Insertar servicios de forma correcta. Resultado: OK.
- Insertar servicio con una descripción existente. Resultado: ERROR.
- Insertar servicio con descripción nula. Resultado: ERROR.

DEL_SERVICIO

- Baja de servicio de forma correcta. Resultado: OK
- Baja de servicio no existente. Resultado: ERROR.
- Baja de servicio dado de baja. Resultado: ERROR.

RES_SERVICIO

- Recuperar servicio de forma correcta. Resultado: OK.
- Recuperar servicio no existente. Resultado: ERROR.
- Recuperar servicio no dado de baja. Resultado: ERROR.

UPD_SERVICIO

- Modificar servicio de forma correcta. Resultado: OK.
- Modificar servicio no existente. Resultado: ERROR.
- Modificar servicio con una descripción que ya existe. Resultado: ERROR.
- Modificar servicio con valor nulo. Resultado: ERROR.
- Modificar servicio dado de baja. Resultado: ERROR.

8.2.1.2 TEST ELECTRODOMÉSTICOS

Id: TEST02

Pruebas realizadas:

INS_ELECTRODOMESTICO

- Insertar electrodoméstico de forma correcta. Resultado: OK.
- Insertar electrodoméstico con una descripción existente. Resultado: ERROR.
- Insertar electrodoméstico con descripción nula. Resultado: ERROR.

DEL_ELECTRODOMESTICO

- Baja de electrodoméstico de forma correcta. Resultado: OK
- Baja de electrodoméstico no existente. Resultado: ERROR.
- Baja de electrodoméstico dado de baja. Resultado: ERROR.

RES_ELECTRODOMESTICO

- Recuperar electrodoméstico de forma correcta. Resultado: OK.
- Recuperar electrodoméstico no existente. Resultado: ERROR.
- Recuperar electrodoméstico no dado de baja. Resultado: ERROR.

UPD_ELECTRODOMESTICO

- Modificar electrodoméstico de forma correcta. Resultado: OK.
- Modificar electrodoméstico no existente. Resultado: ERROR.
- Modificar electrodoméstico con una descripción que ya existe. Resultado: ERROR.
- Modificar electrodoméstico con valor nulo. Resultado: ERROR.
- Modificar electrodoméstico dado de baja. Resultado: ERROR.

8.2.1.3 TEST PROVINCIAS

Id: TEST03

Pruebas realizadas:

INS_PROVINCIA

- Insertar provincia de forma correcta. Resultado: OK.
- Insertar provincia con código existente. Resultado: ERROR.
- Insertar provincia con nombre existente. Resultado: ERROR.
- Insertar provincia con valor nulo. Resultado: ERROR.

DEL_PROVINCIA

- Baja de provincia de forma correcta. Resultado: OK
- Baja de provincia no existente. Resultado: ERROR.
- Baja de provincia con poblaciones asignadas. Resultado: ERROR.

UPD_PROVINCIA

- Modificar provincia de forma correcta. Resultado: OK.
- Modificar provincia no existente. Resultado: ERROR.
- Modificar provincia con un nombre que ya existe. Resultado: ERROR.
- Modificar provincia con valor nulo. Resultado: ERROR.

8.2.1.4 TEST POBLACIONES

Id: TEST04

Pruebas realizadas:

INS_POBLACION

- Insertar población de forma correcta. Resultado: OK.
- Insertar población con código existente. Resultado: ERROR.
- Insertar población con nombre existente. Resultado: ERROR.
- Insertar población con provincia no existente. Resultado: ERROR.
- Insertar población con valor nulo. Resultado: ERROR.

DEL_POBLACION

- Baja de población de forma correcta. Resultado: OK
- Baja de población no existente. Resultado: ERROR.
- Baja de población asignada a una zona. Resultado: ERROR.

UPD_POBLACION

- Modificar población de forma correcta. Resultado: OK.
- Modificar población no existente. Resultado: ERROR.
- Modificar población con un nombre que ya existe. Resultado: ERROR.
- Modificar población con una provincia no existente. Resultado: ERROR.
- Modificar provincia con valor nulo. Resultado: ERROR.

8.2.1.5 TEST ZONAS

Id: TEST05

Pruebas realizadas:

INS_ZONA

- Insertar zona de forma correcta. Resultado: OK.
- Insertar zona con nombre existente. Resultado: ERROR.
- Insertar zona con valor nulo. Resultado: ERROR.

DEL_ZONA

- Baja de zona de forma correcta. Resultado: OK
- Baja de zona no existente. Resultado: ERROR.
- Baja de zona con poblaciones asignadas. Resultado: ERROR.

UPD_ZONA

- Modificar zona de forma correcta. Resultado: OK.
- Modificar zona no existente. Resultado: ERROR.
- Modificar zona con un nombre que ya existe. Resultado: ERROR.
- Modificar zona con valor nulo. Resultado: ERROR.

ASG_PROVINCIAZONA

- Asignamos una provincia de forma correcta. Resultado: OK.
- Asignamos una provincia no existente. Resultado: ERROR.
- Asignamos una provincia a una zona no existente. Resultado: ERROR.

ASG_POBLACIONZONA

- Asignamos una población de forma correcta. Resultado: OK.
- Asignamos una población no existente. Resultado: ERROR
- Asignamos una población a una zona no existente. Resultado: ERROR.
- Asignamos una población ya asignada a la zona. Resultado: ERROR.

DSG_PROVINCIAZONA

- Desasignamos una provincia de forma correcta. Resultado: OK.
- Desasignamos una provincia no existente. Resultado: ERROR.
- Desasignamos una provincia a una zona no existente. Resultado: ERROR.

DSG_POBLACIONZONA

- Desasignamos una población de forma correcta. Resultado: OK.
- Desasignamos una población no existente. Resultado: ERROR
- Desasignamos una población a una zona no existente. Resultado: ERROR.
- Desasignamos una población no asignada a la zona. Resultado: ERROR.

8.2.2 Subsistema Personal

8.2.2.1 TEST TELEOPERADORES

Id: TEST06

Pruebas realizadas:

INS_TELEOPERADOR

- Insertar teleoperador de forma correcta. Resultado: OK.
- Insertar teleoperador con un NIF existente. Resultado: ERROR.
- Insertar teleoperador con valores nulos. Resultado: ERROR.

DEL_TELEOPERADOR

- Baja de teleoperador de forma correcta. Resultado: Ok
- Baja de teleoperador no existente. Resultado: ERROR.
- Baja de teleoperador dado de baja. Resultado: ERROR.

RES_TELEOPERADOR

- Recuperar teleoperador de forma correcta. Resultado: OK.
- Recuperar teleoperador no existente. Resultado: ERROR.
- Recuperar teleoperador no dado de baja. Resultado: ERROR.

UPD_TELEOPERADOR

- Modificar teleoperador de forma correcta. Resultado: OK.
- Modificar teleoperador no existente. Resultado: ERROR.
- Modificar teleoperador con un NIF que ya existe. Resultado: ERROR.
- Modificar teleoperador con valor nulo. Resultado: ERROR.
- Modificar teleoperador dado de baja. Resultado: ERROR.

8.2.2.2 TEST TECNICOS

Id: TEST07

Pruebas realizadas:

INS_TECNICO

- Insertar técnico de forma correcta. Resultado: OK.
- Insertar técnico con un NIF existente. Resultado: ERROR.
- Insertar técnico con valores nulos. Resultado: ERROR.

DEL_TECNICO

- Baja de técnico de forma correcta. Resultado: Ok
- Baja de técnico no existente. Resultado: ERROR.
- Baja de técnico dado de baja. Resultado: ERROR.

RES_ TECNICO

- Recuperar técnico de forma correcta. Resultado: OK.
- Recuperar técnico no existente. Resultado: ERROR.
- Recuperar técnico no dado de baja. Resultado: ERROR.

UPD_ TECNICO

- Modificar técnico de forma correcta. Resultado: OK.
- Modificar técnico no existente. Resultado: ERROR.
- Modificar técnico con un NIF que ya existe. Resultado: ERROR.
- Modificar técnico con valor nulo. Resultado: ERROR.
- Modificar técnico dado de baja. Resultado: ERROR.

ASG_ SERVICIOTECNICO

- Asignamos un servicio de forma correcta. Resultado: OK.
- Asignamos un servicio no existente. Resultado: ERROR.
- Asignamos un servicio a un técnico no existente. Resultado: ERROR.
- Asignamos un servicio dado de baja. Resultado: ERROR.
- Asignamos un servicio a un técnico dado de baja. Resultado: ERROR.
- Asignamos un servicio ya asignado al mismo técnico. Resultado: ERROR.

DSG_ SERVICIOTECNICO

- Desasignamos un servicio de forma correcta. Resultado: OK.
- Desasignamos un servicio no existente. Resultado: ERROR.
- Desasignamos un servicio dado de baja. Resultado: ERROR.
- Desasignamos un servicio a un técnico no existente. Resultado: ERROR.
- Desasignamos un servicio a un técnico dado de baja. Resultado: ERROR.

ASG_ ELECTRODOMESTICOTECNICO

- Asignamos un electrodoméstico de forma correcta. Resultado: OK.
- Asignamos un electrodoméstico no existente. Resultado: ERROR.
- Asignamos un electrodoméstico a un técnico no existente. Resultado: ERROR.
- Asignamos un electrodoméstico dado de baja. Resultado: ERROR.
- Asignamos un electrodoméstico a un técnico dado de baja. Resultado: ERROR.
- Asignamos un electrodoméstico ya asignado al mismo técnico. Resultado: ERROR.

DSG_ ELECTRODOMESTICOTECNICO

- Desasignamos un electrodoméstico de forma correcta. Resultado: OK.
- Desasignamos un electrodoméstico no existente. Resultado: ERROR.
- Desasignamos un electrodoméstico dado de baja. Resultado: ERROR.
- Desasignamos un electrodoméstico a un técnico no existente. Resultado: ERROR.
- Desasignamos un electrodoméstico a un técnico dado de baja. Resultado: ERROR.

ASG_ ZONATECNICO

- Asignamos una zona de forma correcta. Resultado: OK.
- Asignamos una zona no existente. Resultado: ERROR.
- Asignamos una zona a un técnico no existente. Resultado: ERROR.
- Asignamos una zona a un técnico dado de baja. Resultado: ERROR.
- Asignamos una zona ya asignada al mismo técnico. Resultado: ERROR.

DSG_ ZONATECNICO

- Desasignamos una zona de forma correcta. Resultado: OK.
- Desasignamos una zona no existente. Resultado: ERROR.
- Desasignamos una zona a un técnico no existente. Resultado: ERROR.
- Desasignamos una zona a un técnico dado de baja. Resultado: ERROR.

8.2.3 Subsistema Petición

8.2.3.1 TEST CLIENTES

Id: TEST08

Pruebas realizadas:

INS_CLIENTE

- Insertar cliente de forma correcta. Resultado: OK.
- Insertar cliente con un NIF existente. Resultado: ERROR.
- Insertar cliente con valores nulos. Resultado: ERROR.
- Insertar cliente con un código postal no existente. Resultado: ERROR.

UPD_CLIENTE

- Modificar cliente de forma correcta. Resultado: OK.
- Modificar cliente no existente. Resultado: ERROR.
- Modificar cliente con un NIF que ya existe. Resultado: ERROR.
- Modificar cliente con valor nulo. Resultado: ERROR.
- Modificar cliente con un código postal no existente. Resultado: ERROR.

8.2.3.2 TEST PRODUCTOS

Id: TEST09

Pruebas realizadas:

INS_PRODUCTO

- Insertar producto de forma correcta. Resultado: OK.
- Insertar producto con un número de serie existente. Resultado: ERROR.
- Insertar producto con valores nulos. Resultado: ERROR.
- Insertar producto con un electrodoméstico no existente. Resultado: ERROR.
- Insertar producto con un electrodoméstico dado de baja. Resultado: ERROR.
- Insertar producto con un código postal no existente. Resultado: ERROR.
- Insertar producto con un cliente no existente. Resultado: ERROR.

UPD_PRODUCTO

- Modificar producto de forma correcta. Resultado: OK.
- Modificar producto no existente. Resultado: ERROR.
- Modificar producto con un número de serie que ya existe. Resultado: ERROR.
- Modificar producto con valor nulo. Resultado: ERROR.
- Modificar producto con un electrodoméstico no existente. Resultado: ERROR.
- Modificar producto con un electrodoméstico dado de baja. Resultado: ERROR.
- Modificar producto con un código postal no existente. Resultado: ERROR.
- Modificar producto con un cliente no existente. Resultado: ERROR.

8.2.3.3 TEST PETICIONES

Id: TEST10

Pruebas realizadas:

INS_PETICION

- Insertar petición de forma correcta. Resultado: OK.
- Insertar petición con un producto no existente. Resultado: ERROR.
- Insertar petición con un servicio no existente. Resultado: ERROR.
- Insertar petición con un servicio dado de baja. Resultado: ERROR.
- Insertar petición con un teleoperador no existente. Resultado: ERROR.
- Insertar petición con un teleoperador dado de baja. Resultado: ERROR.

CANCEL_PETICION

- Anulamos petición de forma correcta. Resultado: OK.
- Anulamos petición no existente. Resultado: ERROR.
- Anulamos petición con estado distinto de PENDIENTE. Resultado: ERROR.

8.2.4 Subsistema Atención

8.2.4.1 TEST ASIGNACIONES

Id: TEST11

Pruebas realizadas:

ASG_TECNICO

- Asignar técnico a una petición de forma correcta. Resultado: OK.
- Asignar técnico a una petición no existente. Resultado: ERROR.
- Asignar técnico a una petición en estado distinto de PENDIENTE. Resultado: ERROR.
- Asignar técnico no existente a una petición. Resultado: ERROR.
- Asignar técnico dado de baja a una petición. Resultado: ERROR.

DSG_TECNICO

- Desasignar técnico de una petición de forma correcta. Resultado: OK.
- Desasignar técnico de una petición no existente. Resultado: ERROR.
- Desasignar técnico de una petición en estado distinto de ASIGNADA. Resultado: ERROR.

8.2.4.2 TEST TRABAJOS

Id: TEST12

Pruebas realizadas:

INS_TRABAJO

- Insertar trabajo de forma correcta. Resultado: OK.
- Insertar trabajo con valores nulos. Resultado: ERROR.
- Insertar trabajo con una petición no existente. Resultado: ERROR.
- Insertar trabajo con una petición en estado distinto de ASIGNADA. Resultado: ERROR.
- Insertar trabajo con una petición asignada a otro técnico. Resultado: ERROR.
- Insertar trabajo con un técnico no existente. Resultado: ERROR.
- Insertar trabajo con un técnico dado de baja. Resultado: ERROR.
- Insertar trabajo con un valor de importe no valido. Resultado: ERROR.
- Insertar trabajo con un valor de garantía no valido. Resultado: ERROR.

CLOSE_PETICION

- Cerramos petición de forma correcta. Resultado: OK.
- Cerramos petición no existente. Resultado: ERROR.
- Cerramos petición con estado distinto de ASIGNADA. Resultado: ERROR.
- Cerramos petición sin trabajos asociados. Resultado: ERROR.

8.2.4.3 TEST ENCUESTAS

Id: TEST13

Pruebas realizadas:

INS_ENCUESTA

- Insertar encuesta de forma correcta. Resultado: OK.
- Insertar encuesta con una petición no existente. Resultado: ERROR.
- Insertar encuesta con una petición en estado distinto de CERRADA. Resultado: ERROR.
- Insertar encuesta con un teleoperador no existente. Resultado: ERROR.
- Insertar trabajo con un teleoperador dado de baja. Resultado: ERROR.
- Insertar trabajo con un valor de valoración no valido. Resultado: ERROR.

8.2.5 Consultas

8.2.5.1 CONSULTAS SUBSISTEMA CONFIGURACIÓN

Id: TEST14

Pruebas realizadas:

- Consultar un id.
- Consultar un id no existente.
- Consultar todos los registros.

GET_SERVICIO

GET_ELECTRODOMESTICO

GET_PROVINCIA

GET_POBLACION

GET_ZONA

GET_POBLACIONESZONA

8.2.5.2 CONSULTAS SUBSISTEMA PERSONAL

Id: TEST15

Pruebas realizadas:

- Consultar un id.
- Consultar un id no existente.
- Consultar todos los registros.

GET_TELEOPERADOR

GET_TECNICO

GET_SERVICIOSTECNICO

GET_ELECTRODOMESTICOSTECNICO

GET_ZONASTECNICO

8.2.5.3 CONSULTAS SUBSISTEMA PETICIÓN

Id: TEST16

Pruebas realizadas:

- Consultar un id.
- Consultar un id no existente.
- Consultar todos los registros.

GET_CLIENTE
GET_PRODUCTO
GET_PETICION

8.2.5.4 CONSULTAS SUBSISTEMA PETICION

Id: TEST17

Pruebas realizadas:

- Consultar un id.
- Consultar un id no existente.
- Consultar todos los registros.

GET_PETICIONESESERVICIOZONA
GET_PETICIONESETECNICO
GET_TRABAJO
GET_ENCUESTA

8.2.5.5 CONSULTAS SUBSISTEMA HISTORIAL

Id: TEST18

Pruebas realizadas:

- Consultar un id.
- Consultar un id no existente.

GET_HCLIENTE
GET_HPRODUCTO
GET_HTECNICO

8.3 Carga de Datos

En el producto podemos encontrar un paquete con un procedimiento que realizará una carga inicial del sistema con valores aleatorios. Los valores iniciales pueden ser modificados en el procedimiento CARGA del paquete CARGA_INICIAL.

Los valores por defecto son:

- Servicios = 5
- Electrodomésticos = 5
- Zonas y Provincias = 5 (se asignará una zona por provincia)
- Poblaciones = 50 por cada provincia (5x50=250)
- Teleoperadores = 5
- Técnicos = 20
- Clientes = 300

- Productos = valor aleatorio de un máximo de 3 por cliente (máximo $3 \times 300 = 900$)
- Peticiones = valor aleatorio de un máximo de 5 por producto (máximo $5 \times 900 = 4.500$)

Las fechas se calcularán de forma aleatoria proporcionando un abanico de unos ocho años para poder realizar las estadísticas del sistema.

8.4 ETL

Dentro del paquete ETL del usuario DW, nos encontramos con el procedimiento CARGAS (*ETL.CARGAS*). Este procedimiento actualizará la información del almacén de datos con los contenidos en la base de datos operacional previa normalización que nos garantizará la calidad final de los mismos.

Este procedimiento se ejecutará una vez al mes. El proceso será lanzado por un Job que se programará los días 1 de cada mes en un horario no laboral e intentando que no se solape con las copias de seguridad.

Los procedimientos de actualización constan de dos fases. Una primera en la que se actualizan los datos existentes y, una vez actualizados, se insertan los nuevos valores.

8.5 Estadísticas

Las estadísticas las realizaremos con el software Cognos de IBM. Este software nos proporciona una serie de aplicaciones que nos permitirán la creación de informes y cubos multidimensionales que darán información del sistema y empresa al personal responsable.

Se han elaborado los informes identificados en el apartado de requerimientos que podremos ver en el documento Anexo2 – Estadísticas, así como una breve descripción de la elaboración de informes y cubos multidimensionales realizados con la suite Cognos.

9 Valoración Económica

La jornada de trabajo la estableceré en una media de 3 horas diarias lo que hace que el tiempo previsto para el desarrollo del proyecto sea de 303 horas (101 días x 3h/día).

Para el desarrollo del proyecto participarán tres perfiles bien diferenciados: Jefe de Proyecto, Analista y Programador.

Jefe de Proyecto: Interviene en todas las fases del proyecto. Su trabajo será de relación con el cliente y supervisión de los trabajos de análisis como de implementación.

Analista: Realizará la toma de requerimientos, diseño de la base de datos, diseño del almacén de datos, efectuará la prueba final del sistema y confeccionará la documentación del proyecto.

Programador: Realizará los scripts de creación de la base de datos y del almacén de datos. Será el encargado de desarrollar los procedimientos y funciones PL/SQL de todo el sistema. Elaborará los informes y cubos multidimensionales para la obtención de las estadísticas.

A continuación presentamos un desglose de las actividades, junto con una valoración económica:

Recurso	Actividad	Horas	Dedicación	Horas Dedicadas	Precio Hora Recurso	Importe Actividad
Jefe Proyecto	Lectura del PFC	9	100%	9,00	60,00	540,00
Jefe Proyecto	Planificación de tareas	12	100%	12,00	60,00	720,00
Analista	Documento Entrega	27	20%	5,40	50,00	270,00
Jefe Proyecto	Revisión y Entrega	3	100%	3,00	60,00	180,00
Programador	Instalación Oracle 10g Express Edition	3	100%	3,00	40,00	120,00
Analista	Análisis de Requerimientos	15	100%	15,00	50,00	750,00
Analista	Diseño BD	24	100%	24,00	50,00	1.200,00
Programador	Implementación BD	6	100%	6,00	40,00	240,00
Analista	Diseño DW	18	100%	18,00	50,00	900,00
Programador	Implementación DW	6	100%	6,00	40,00	240,00
Analista	Pruebas	9	100%	9,00	50,00	450,00
Analista	Documentos entregables	60	20%	12,00	50,00	600,00
Jefe Proyecto	Revisión y Entrega	3	100%	3,00	60,00	180,00
Programador	Scripts Procedimientos Subsistema Control	3	100%	3,00	40,00	120,00
Analista	Pruebas Subsistema Control	3	100%	3,00	50,00	150,00
Programador	Scripts Procedimientos Subsistema Configuración	6	100%	6,00	40,00	240,00
Analista	Pruebas Subsistema Configuración	6	100%	6,00	50,00	300,00
Programador	Scripts Procedimientos Subsistema Personal	6	100%	6,00	40,00	240,00
Analista	Pruebas Subsistema Personal	6	100%	6,00	50,00	300,00
Programador	Scripts Procedimientos Subsistema Petición	6	100%	6,00	40,00	240,00
Analista	Pruebas Subsistema Petición	6	100%	6,00	50,00	300,00
Programador	Scripts Procedimientos Subsistema Atención	6	100%	6,00	40,00	240,00
Analista	Pruebas Subsistema Atención	6	100%	6,00	50,00	300,00
Programador	Scripts Procedimientos Subsistema Historial	6	100%	6,00	40,00	240,00
Analista	Pruebas Subsistema Historial	6	100%	6,00	50,00	300,00
Programador	Scripts ETL	6	100%	6,00	40,00	240,00
Analista	Pruebas Scripts ETL	6	100%	6,00	50,00	300,00
Analista	Documentos entregables	81	20%	16,20	50,00	810,00
Jefe Proyecto	Revisión y Entrega PAC3	3	100%	3,00	60,00	180,00
Analista	Prueba Sistema Completo	21	100%	21,00	50,00	1.050,00
Programador	Instalación Cognos	6	100%	6,00	40,00	240,00
Analista	Generación de Estadísticas	30	100%	30,00	50,00	1.500,00
Analista	Memoria	48	20%	9,60	50,00	480,00
Analista	Presentación Virtual	30	100%	30,00	50,00	1.500,00
Programador	Producto	3	100%	3,00	40,00	120,00
Jefe Proyecto	Entrega Final	3	100%	3,00	60,00	180,00
TOTAL PROYECTO						15.960,00

Total Jefe Proyecto	1.980,00
Total Analista	11.460,00
Total Programador	2.520,00

Figura 27: Valoración Económica.

Observamos que el coste total del proyecto será de 15.960,00€

10 CONCLUSIONES

En el transcurso de todo el proyecto se han puesto en práctica métodos y técnicas estudiadas a lo largo de la ingeniería, reforzándolos con habilidades y conocimientos propios del área de bases de datos relacionales. El proyecto se ha adaptado a la planificación inicial presentada, salvo una modificación en las actividades realizada para adaptar mejor el proyecto a los subsistemas creados.

Desde un punto de vista personal, puedo decir que la experiencia ha sido muy enriquecedora. Ya que me ha permitido ampliar mis conocimientos de Bases de datos operacionales, programación PL/SQL, Almacenes de datos, Software Cognos y poner en práctica las metodologías de la Gestión de Proyectos.

Dando una vista atrás, parece increíble que partiendo de un enunciado tan escueto como la necesidad de la empresa de gestionar la postventa de los productos hemos:

- Realizado una Planificación
- Establecido una valoración
- Determinado los requerimientos tanto funcionales como no funcionales del sistema
- Elaborado un producto
- Construido unas Estadísticas

11 GLOSARIO

Almacén de Datos: Conjunto de los datos históricos de una empresa.

Atributo: Cada una de las propiedades o características que se almacena en una entidad.

Base de Datos: Conjunto de datos pertenecientes al mismo contexto almacenados bajo una estructura para su posterior uso.

Constraint: Restricción que se aplica a los atributos de una tabla, de tal forma que si no se cumple se generará un error.

Data Warehouse (DW): Ver Almacén de Datos.

Diseño Conceptual: Etapa del diseño de la base de datos en la que se obtiene una estructura independiente de la tecnología.

Diseño Lógico: Etapa del diseño de la base de datos en la que partiendo de un diseño conceptual, éste se adapta al modelo del SGBD con el que se va a implementar la base de datos.

Electrodoméstico: Categoría en la que se clasifican los distintos productos que atiende el sistema.

Encuesta: Valoración de la solución ofrecida a la petición de servicio solicitada por un cliente.

Entidad: Elementos de la vida real del cual nos interesa almacenar alguna de sus propiedades.

Interrelación: Asociación que se crea entre una o más entidades.

Log: Mensajes con información del estado o resultado de un proceso.

Oracle: Es un SGBD relacional, fabricado por Oracle Corporation y considerado el más completo del mercado.

Package (Paquete): Es una encapsulación de determinados objetos de la base de datos.

Petición: Solicitud de trabajo sobre un producto propiedad de un cliente.

Procedimiento Almacenado: Programa que se encuentra físicamente almacenado en la base de datos. Es definido por el usuario y soporta tanto parámetros de entrada como de salida.

PL/SQL: Lenguaje de programación embebido en Oracle que soporta todas las consultas y manipulación de datos que se usan en SQL.

Servicio: Indicará las distintas clases de operaciones que atenderá la empresa (Reparación, Instalación, etc).

Script: Conjunto de instrucciones que permiten la automatización de tareas.

SGBD (Sistema de Gestión de Base de Datos): Software que gestiona y controla la base de datos, encargándose tanto de la integridad de datos como de la utilización simultánea por diferentes usuarios.

Sistema de Base de Datos: Software dedicado a servir de interfaz entre la base de datos y el usuario o las aplicaciones que la utilizan.

SQL: Lenguaje de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas.

Técnico: Persona que realizará el servicio solicitado por el cliente.

Teleoperador: Persona que recogerá telefónicamente las peticiones de los clientes y realizará las encuestas de satisfacción.

Trabajo: Conjunto de las acciones realizadas por un técnico para solucionar la petición de servicio solicitada por un cliente.

Trigger: Evento que se ejecuta cuando se cumple una condición establecida al realizar una operación de inserción, actualización o borrado sobre una tabla.

12 BIBLIOGRAFIA

- ❖ **Base de Datos I**
Jaume Sistac Planas
Segunda Edición: Septiembre 2005
Fundació per a la Universitat Oberta de Catalunya
- ❖ **Base de Datos II**
Jaume Sistac Planas
Segunda Edición: Febrero 2004
Fundació per a la Universitat Oberta de Catalunya
- ❖ **Almacén de datos y modelos multidimensionales**
Àngels Rius Gavídia i Montse Serra Vizern
Primera Edición: Septiembre 2004
Fundació per a la Universitat Oberta de Catalunya
- ❖ **Oracle Database 10g. PL/SQL Programming**
Scott Urman
Kindle Edition: 2004
Oracle Press
- ❖ **Oracle 10g. Administración y Análisis de Bases de Datos**
César Pérez
Primera Edición: 2005
Ra-Ma
- ❖ **Sistemas de Gestión de Bases de Datos**
Jaume Sistac Planas
Segunda Edición: Septiembre 2009
Fundació per a la Universitat Oberta de Catalunya
- ❖ **Oracle Database Lite 10g(10.2.0) Documentation**
http://download.oracle.com/docs/cd/B19188_01/index.htm
- ❖ **Oracle PL/SQL Topics**
<http://www.techonthenet.com/oracle/index.php>
- ❖ **Wikipedia. Enciclopedia de contenido libre**
<http://es.wikipedia.org>
- ❖ **ZonaOracle.com**
<http://www.zonaoracle.com>