

Diseño e implementación de la base de datos de un sistema de control energético

Nombre del Estudiante: *Alberto Ramos Sagullo*
ETIS

Nombre del Consultor: Manuel Rella Ruiz

AGRADECIMIENTOS

Agradezco a mi familia la comprensión por las horas y la dedicación,
a veces cuasi plena al estudio de las asignaturas de la carrera.

Muchas horas invertidas que sirven y han servido para adquirir conocimientos que he
empleado directamente en mi trabajo diario.

Por otro lado la forma de trabajar no presencial y la estructura UOC, me ha permitido
situarme privilegiadamente en el tan dinámico mundo de la informática.

RESUMEN

La Comunidad Europea pretende guardar la información necesaria para entender y mejorar el consumo energético. Dentro de la partida presupuestaria destinada a controlar el uso de la energía decide abrir un concurso público para el diseño e implementación de dicha finalidad.

El objetivo de éste proyecto es proporcionar un diseño completo que cumpla todos los requisitos exigidos. Especial énfasis en todo el modelo estadístico de consulta de datos – actuales e históricos actualizados -.

Por otro lado conseguimos tener una base de datos actualizada, con las personas, tanto empleados (personas que hacen lecturas de contador o bien clientes que nos facilitan la lectura), clientes que tienen o han tenido contratos, personas en general que se han dado de alta en la base de datos para futuras propuestas (posibles clientes, estudios de viabilidad, ampliaciones etc.).

CONCLUSIONES

- Se ha seguido la metodología y planificación prevista.
- Se han alcanzado los objetivos planteados en el enunciado.
- He adquirido destreza y conocimientos en el desarrollo del proyecto.
- Se ha logrado eficiencia, robustez y excelencia.
- Se han automatizado procesos:
 - Al dar de alta un contrato preparamos los contadores, y los consumos.
 - Al dar de alta la lectura de contador, actualizamos el histórico de lecturas, los consumos y las tablas de estadísticas.
- Se han logrado así procesos modulares.
- Se utiliza una implementación encadenada, con mejoras añadidas.
- Se ha conseguido una aplicación modular, con estructura reutilizable para otras aplicaciones. Ej. La gestión de las ubicaciones.
- En resumen un duro trabajo que ha servido para probar la capacidad de generar un proyecto de cierta complejidad, haciendo uso de los conocimientos adquiridos a lo largo de los estudios en la UOC.

Índice de contenidos

AGRADECIMIENTOS	2
RESUMEN.....	3
CONCLUSIONES.....	3
ÍNDICE DE CONTENIDOS.....	4
ÍNDICE DE ILUSTRACIONES.....	8
ÍNDICE DE TABLAS.....	8
1. INTRODUCCIÓN.....	11
1.1 JUSTIFICACIÓN	11
1.2 OBJETIVOS.....	11
1.2.1 Objetivos generales del TFC.....	11
1.2.2 Objetivos específicos del TFC.....	12
1.2.3 Objetivos alternativos de valor añadido.....	12
1.3 METODOLOGÍA.....	13
1.4 PLANIFICACIÓN CON HITOS Y TEMPORALIZACIÓN.....	14
1.4.1 Entregas parciales – PACs -	14
Diagrama de Gant	14
1.4.3 Hitos del proyecto.....	15
1.4.4 Tabla de fechas.....	15
1.4.5 Identificación de tareas.....	16
1.5 VALORACIÓN INICIAL DEL COSTE	16
1.5.1 Desglose del coste económico.....	17
1.6 RECURSOS HUMANOS TÉCNICOS Y MATERIALES.....	18
1.7 ANÁLISIS DE RIESGOS	19
2. ANÁLISIS DE REQUISITOS	20
DESCRIPCIÓN INICIAL.....	20
SUPUESTOS INICIALES	20
REQUISITOS FUNCIONALES.....	21
OTROS REQUISITOS	22
REQUISITOS NO FUNCIONALES.....	22
3. DISEÑO DEL SISTEMA.....	23
DISEÑO CONCEPTUAL	23
• Esquema conceptual.....	23
• Esquema de base de datos.....	23
• Esquema Interno.....	23
• Base de Datos física.....	23
DIAGRAMA UML CONCEPTUAL	24
IDENTIFICACIÓN DE LAS ENTIDADES.....	25
CENTRALES	26
CLIENTES	26
CONSUMOS.....	26
CONTADORES.....	26
CONTRATOS	26
EMPRESAS.....	26
ESTADOS.....	26
INSPECCIONES.....	26
LECTURAS.....	26
LINEAS.....	26
LOCALIDAD.....	26
MODELO	26
PAIS	26

PERSONA	26
PROVINCIA	26
UBICACION	26
VIA	26
IDENTIFICACIÓN DE RELACIONES	27
Central Distribución – Central Producción	27
Contador – Línea de comunicación - Central Distribución	28
Cliente – Contador- Contratos.....	29
Ubicación de contadores, Centrales y clientes	29
Lecturas Contador.....	30
Lecturas Consumo.....	30
Inspecciones Central.....	31
Ubicar –Centrales, contadores, clientes, empresas.....	31
ATRIBUTOS DE ENTIDADES	32
CENTRALES	33
CLIENTES	33
CONSUMOS.....	33
CONTADORES.....	33
CONTRATOS	33
EMPRESAS	34
ESTADOS.....	34
INSPECCIONES.....	34
LECTURAS.....	34
LINEAS	34
LOCALIDAD.....	34
MODELOS	35
PAIS	35
Particular.....	35
PERSONA	35
PROVINCIA	35
TIPOS DE CENTRAL.....	36
Tipos de lectura	36
UBICACIONES	36
VIA.....	36
DISEÑO LÓGICO - RELACIONAL	37
DIAGRAMA LÓGICO - RELACIONAL	38
Módulo estadístico	39
E1_centralConsumo.....	39
E2_lineaConsumo.....	39
E3_lineaMaximaCarga.....	39
E4_lineaMas50	39
E5_centralProduccionMenorEnergia.....	39
E6_topContador.....	39
E6_topContador.....	39
Procedimiento LOG.....	40
DISEÑO FÍSICO	41
• Creación de Tablas.....	41
• Creación de Secuencias y Disparadores (triggers)	41
• Creación de Procdimientos (altas, modificaciones, bajas).....	41
• Creación de Funciones.....	41
<u>IMPLEMENTACIÓN</u>	<u>42</u>
• Creación de inserciones	42
<u>MODIFICACIONES</u>	<u>44</u>
DIAGRAMA LÓGICO RELACIONAL MODIFICADO.....	45
Estadísticas.....	46
ENTIDADES DEL MODELO RELACIONAL FINAL.....	47
estados	47
pais	47
provincia	47
localidad.....	47
via	47
ubicaciones.....	47
tCliente.....	47
clientes.....	47
empresas	47
modelos	47
contadores	48

contratos.....	48
eCentral.....	48
cCentral.....	48
tCentral.....	48
tInspeccion.....	48
centrales.....	48
linea.....	48
lineas.....	48
tLectura.....	48
lecturas.....	48
consumos.....	49
Log.....	49
E_1.....	49
E_2.....	49
E_3.....	49
E_4.....	49
E_5.....	49
E_6.....	49
E_7.....	49
CONSULTA 1 a.....	49
CONSULTA 2 b.....	50
CONSULTA 3 c.....	50
CONSULTA 4 d.....	50
CONSULTA 5 e.....	50
CONSULTA 6 f.....	50
CONSULTA 7 g.....	50
MANUAL DE FUNCIONAMIENTO.....	51
PRUEBAS DE FUNCIONAMIENTO.....	55
✓ GENERAL.-.....	56
✓ LECTURAS.-.....	60
✓ ESTADÍSTICAS.-.....	62
A. PROCEDIMIENTOS EMPLEADOS:.....	62
B. FUNCIONES:.....	62
C. RESULTADO ACTUALIZAR ESTADÍSTICAS.....	64
D. RESULTADO TABLAS ESTADÍSTICAS.....	66
A. TABLE E_1.....	66
B. TABLE E_2.....	66
C. TABLE E_3.....	66
E. TABLE E_4.....	66
F. TABLE E_5.....	66
G. TABLE E_6.....	67
H. TABLE E_7.....	67
✓ CONSULTAS.-.....	68
1) C7.- consultarContadoresAntiguedad.....	68
2) C6.- porcentaje lecturas.....	70
3) C5.- energía Central en un periodo.....	71
4) C3.- línea más cargada en relación cap. máxima.....	72
5) C1.- contadoresCiudadConsumo>80%medio.....	74
WEBGRAFÍA Y BIBLIOGRAFÍA.....	76
Fuentes de Internet.....	77
Apuntes.....	77
Enlaces.....	77
Miscelánea.....	77
ANEXO.- PROCEDIMIENTOS ALTAS.....	78
1. GESTOR_ALTAS.....	78
2. altaCentral.....	79
3. altaCentralDistribucion.....	80
4. altaCentralProduccion.....	81
5. altaClaseCentral.....	82
6. altaClientes.....	83
7. altaConsumos.....	84
8. altaContadores.....	85

9.	altaContratos.....	86
10.	altaEmisionCentral	87
11.	altaEmpresa.....	88
12.	altaEstados.....	89
13.	altaLectura	90
14.	altaLecturas.....	91
15.	altaLinea	92
16.	altaLineas	93
17.	altaLocalidad.....	94
18.	altaLog	95
19.	altaModelos	96
20.	altaPais.....	97
21.	altaProvincia	98
22.	altaTipoCentral	99
23.	altaTipoCliente.....	100
24.	altaTipoInspeccion.....	101
25.	altaTipoLectura	102
26.	altaUbicaciones	103
27.	altaVia	104

ANEXO.- PROCEDIMIENTOS MODIFICACIÓN 105

1.	modificaCentral	105
2.	modificaCentralDistribucion	105
3.	modificaCentraProduccion	105
4.	modificaClaseCentral	105
5.	modificaClientes	105
6.	modificaClientesEstado	106
7.	modificaConsumos.....	106
8.	modificaContadores	106
9.	modificaContratos.....	106
10.	modificaEmisionCentral.....	106
11.	modificaEmpresa.....	106
12.	modificaEmpresa.....	107
13.	modificaEstadoCentral	107
14.	modificaEstados	107
15.	modificaLecturas.....	107
16.	modificaLinea.....	107
17.	modificaLineas	107
18.	modificaLocalidad	107
19.	modificaModelos.....	108
20.	modificaPais.....	108
21.	modificaProvincia.....	108
22.	modificaTipoCentral.....	108
23.	modificaTipoCliente	108
24.	modificaTipoInspeccion.....	108
25.	modificaTipoLectura	108
26.	modificaUbicaciones	109
27.	modificaVia	109

ANEXO.- PROCEDIMIENTOS BAJAS..... 110

•	GESTOR_BAJAS	110
1.	bajaCentral.....	110
2.	bajaContador.....	111
3.	bajaLineas.....	112
4.	bajaClientes	113
5.	bajaUbicaciones	114
6.	GESTOR_CONSULTAS	115
7.	Consulta C1 -a-	115
8.	Consulta C2 -b-	116
9.	Consulta C3 -c-.....	117
10.	Consulta C4 -d-	118
11.	Consulta C5 -e-	119
12.	Consulta C6 -f-	120
13.	Consulta C7 -g-.....	121

ANEXO.- GESTOR ESTADÍSTICAS..... 122

1.	Estadística 1 E1	123
2.	Estadística 2 E2	124

3. Estadística 3 E3	125
4. Estadística 4 E4	126
5. Estadística 5 E5	127
6. Estadística 6 E6	128
7. Estadística 7 E7	129

ANEXO.- CÓDIGO PROCESO ESTADÍSTICAS..... 130

ANEXO.- FUNCIONES..... 137

Guia funcion & procedimiento.	137
------------------------------------	-----

ANEXO.- CÓDIGO – PRUEBAS DE SALIDA DE UTILIDADES 140

ANEXO PROCEDURES ANTERIORES A LA MODIFICACIÓN 152

1. modCentral.....	152
2. modContador	153
3. modLineas.....	154
4. modClientes	155
5. modUbicaciones.....	156

Índice de ilustraciones

Ilustración 1 Ciclo de vida BD	13
Ilustración 2 Diagrama de Gant.....	14
Ilustración 3 Diagrama UML –Entidad/relación-.....	24
Ilustración 4 Central Producción / Central Distribución	27
Ilustración 5 Central Producción / Central Distribución - Lineas-.....	28
Ilustración 6 Contador conectado	28
Ilustración 7 Contratos del cliente con contador	29
Ilustración 8 Ubicación Centrales, contadores y clientes	29
Ilustración 9 Lectura – Consumo	30
Ilustración 10 Consumo medio por cliente.....	31
Ilustración 11 Central Inspecciones.....	31
Ilustración 12 Ubicaciones	32
Ilustración 13 Diagrama Lógico-Relacional.....	38
Ilustración 14 LOG.....	40
Ilustración 15 Diagrama Lógico relacional modificado	45
Ilustración 16 Diagrama Lógico Relacional modificado CONSULTAS	45
Ilustración 17 tabla Consulta C7	69
Ilustración 18 consulta C6.....	70
Ilustración 19 tabla consulta C6	70
Ilustración 20 Consulta C1	74

Índice de tablas

Tabla 1 Hitos del proyecto.....	15
Tabla 2 Tabla de fechas	15
Tabla 3 Identificación de tareas.....	16
Tabla 4 Horas facturables	17
Tabla 5 Tarifas.....	17
Tabla 6 Coste económico	17
Tabla 7 Tabla de entidades	25
Tabla 8 Atributos de Entidades	32
Tabla 9 Módulo estadístico.....	39

Tabla 10 Resumen de Procedures.....	55
Tabla 11 Funciones.....	55
Tabla 12 Carga prueba de BD	56
Tabla 13 Pruebas de ejecución	57
Tabla 14 Inicializar procesos gestión	60
Tabla 15 lista Funciones empleadas	62
Tabla 16 función y procedimiento que se emplea	63
Tabla 17 ejecutar proceso estadísticas.....	63
Tabla 18 Resultado ACTUALIZAR ESTADÍSTICAS	64
Tabla 19 Consulta C7	68
Tabla 20 alta tabla C7.....	68
Tabla 21 consulta C5	71
Tabla 22 alta tabla C5.....	71
Tabla 23 tabla C5.....	71
Tabla 24 consultaC3	72
Tabla 25 altaC3.....	72
Tabla 26 C3	73
Tabla 27 altaC3.....	74
Tabla 28 Alta Central	79
Tabla 29altaCentralDistribucion.....	80
Tabla 30 altaCentralProduccion	81
Tabla 31 altaClaseCentral.....	82
Tabla 32 altaClientes	83
Tabla 33 altaConsumos	84
Tabla 34 altaContadores	85
Tabla 35 altaContratos.....	86
Tabla 36 altaEmisionCentral	87
Tabla 37 altaEmpresa	88
Tabla 38 altaEstados.....	89
Tabla 39 altaLectura	90
Tabla 40 altaLecturas.....	91
Tabla 41 altaLinea	92
Tabla 42 altaLineas.....	93
Tabla 43 altaLog.....	95
Tabla 44 altaModelos	96
Tabla 45 altaProvincia.....	98
Tabla 46 altaTipoCliente	100
Tabla 47 altaUbicaciones.....	103
Tabla 48 modificaCentral	105
Tabla 49 Baja Central	110
Tabla 50 Baja Contador.....	111
Tabla 51 Baja Línea.....	112
Tabla 52 Baja Clientes.....	113
Tabla 53 Baja Ubicación	114
Tabla 54 Consulta C1 -a-.....	115
Tabla 55 Consulta C2 -b-	116
Tabla 56 Consulta C3 -c-.....	117
Tabla 57 Consulta C4 -d-	118
Tabla 58 Consulta C5 -e-.....	119
Tabla 59 Consulta C6 -f-.....	120

Tabla 60 Consulta C7 -g-.....	121
Tabla 61 Estadística E1	123
Tabla 62 Estadística E2	124
Tabla 63 Estadística E3	125
Tabla 64 Estadística E4	126
Tabla 65 Estadística E5	127
Tabla 66 Estadística E6	128
Tabla 67 Estadística E7	129
Tabla 68 Código proceso alta estadísticas	130
Tabla 69 Función - procedimiento	137
Tabla 70 Funciones.....	137
Tabla 71 Prueba inicializarLecturas	140
Tabla 72 Modificación Central.....	152
Tabla 73 Modificación Contador.....	153
Tabla 74 Modificación Línea	154
Tabla 75 Modificación Clientes	155
Tabla 76 Modificación Ubicación	156

1. INTRODUCCIÓN

1.1 Justificación

Este trabajo pretende ser un ejemplo práctico de cómo realizar una base de datos relacional, que a partir de unos requisitos iniciales, permitan mediante la introducción periódica de datos –lecturas de contador- obtener de forma permanentemente actualizada los históricos que nos interesan – especificados en el enunciado -.

Se ha basado en las fases típicas que intervienen en el desarrollo de cualquier proyecto informático –ciclo de vida-:

Planificación
Análisis
Diseño
Implementación
Pruebas
Instalación / Despliegue
Uso y mantenimiento

Cada una de las fases estará documentada en esta memoria. Cada fase no es independiente y cerrada en sí misma sino que concatena y sirve de base a la siguiente, siguiendo la evolución del proyecto. Por lo que según avancemos podrán tener las consiguientes adaptaciones.

La memoria irá evolucionando conforme hagamos las entregas, no conservando el histórico de modificaciones, sino como la versión que contiene el conjunto actualizado.

Pretendemos realizar el proyecto, ajustado a la planificación inicial teniendo en cuenta la duración total y que las horas invertidas tengan relación con los créditos asignados, con la finalidad de obtener un producto equilibrado.

Significar que la inexperiencia y la falta de pericia se tratarán de suplir con una mayor implicación horaria y dedicación.

1.2 Objetivos

1.2.1 Objetivos generales del TFC

- El objetivo de éste Trabajo Final de Carrera es realizar un proyecto informático que permita consolidar los conocimientos adquiridos al cursar las diferentes asignaturas que componen la carrera de Ingeniería Técnica en Informática de Sistemas (Bases de Datos, Sistemas de Gestión de Bases de Datos, entre otras).
- La utilización del Sistema de Gestión de Bases de Datos Oracle, permitirá poner en práctica los conocimientos de un sistema de gestión de gran presencia en el mercado, ampliamente utilizado por muchas grandes compañías en los distintos sectores empresariales.

1.2.2 Objetivos específicos del TFC

Conscientes de la necesidad del aprovechamiento de los recursos energéticos y de su importancia estratégica, la Comunidad Europea pretende obtener actualizados datos estadísticos objetivos sobre su consumo. Se presta especial atención este apartado estadístico, modelando las entidades correspondientes para poder analizar con perspectiva su evolución.

Para la concreción de éste objetivo, tenemos que cumplir los requisitos pedidos en el enunciado:

- Diseñar la Base de Datos – diagramas diseño E/R UMLs... -, realizar los scripts correspondientes - creación de las tablas, índices, etc. – que sean necesarios para implementar los procedimientos almacenados, con especial atención al apartado que ha de cumplir con los requisitos estadísticos.
- Creación de un mecanismo de inicialización de la base de datos, tal que simule en un tiempo suficiente el normal funcionamiento del sistema de control energético. Que tenga en cuenta los supuestos considerados y, que se pueda verificar en tiempo real el correcto funcionamiento del mantenimiento actualizado de las estadísticas de los consumos solicitados en los requisitos.

1.2.2 Objetivos específicos del TFC

- Diseñar una base de datos conforme a los requisitos del enunciado.
- Utilizar el lenguaje SQL – BD Oracle -.
- Gestión de control de errores.
- Tratamiento de excepciones.
- Gestión de ficheros.
- Creación de juegos de pruebas.
- Verificación mantenimiento datos históricos estadísticos.
- Uso y mantenimiento del sistema.

1.2.3 Objetivos alternativos de valor añadido

- Tener un control exhaustivo de cada central, con una búsqueda rápida y sencilla. Implementamos los datos en una sola clase de forma recursiva. Consideramos la Clase Central con un OR lógico (Producción =0 OR Distribución =1).
- En los contadores, tendremos el histórico de todas sus ubicaciones, y también el histórico de todas sus lecturas, pudiendo saber de forma simple todas las instalaciones que han tenido.
- Contemplaremos tener un registro de las inspecciones que han tenido las centrales, incluso los contadores, en fechas determinadas.
- En clientes también lo aprovechamos para tener datos de personas, podemos tener conjuntamente futuros clientes y el control de las personas que efectúan la lectura presencialmente.
- Al efectuar altas de contador podremos escoger por la proximidad geográfica de la subcentral y/o saber las más próximas y a la que podemos conectar.
- Por cada tipo de central tendremos asociado un campo (valorTipoCentral) que cruzado con las características propias de la subcentral nos indicará fácilmente los datos de impacto medio ambiental.

- Con la base de datos de personas podemos estudiar nivel de saturación de contadores o centrales para futuras ampliaciones, o bien campañas personalizadas para racionalizar consumos etc.

1.3 Metodología

La metodología seguida durante el ciclo de vida del proyecto, es la que se conoce como desarrollo en cascada. Consiste en dividir el ciclo de vida en diferentes etapas, de forma que cada etapa no puede teóricamente comenzar hasta que no finalice la anterior.

No daremos por finalizado las revisiones/actualizaciones hasta que los sucesivos refinamientos y depuraciones den como resultado final la solución a las necesidades que el Cliente ha solicitado.

En el sistema comercial ORACLE de base de datos relacional, se diseñara el conjunto de procedimientos que permitirán insertar, modificar, actualizar y consultar, junto con el modulo estadístico que mantendrá actualizada la información crítica.

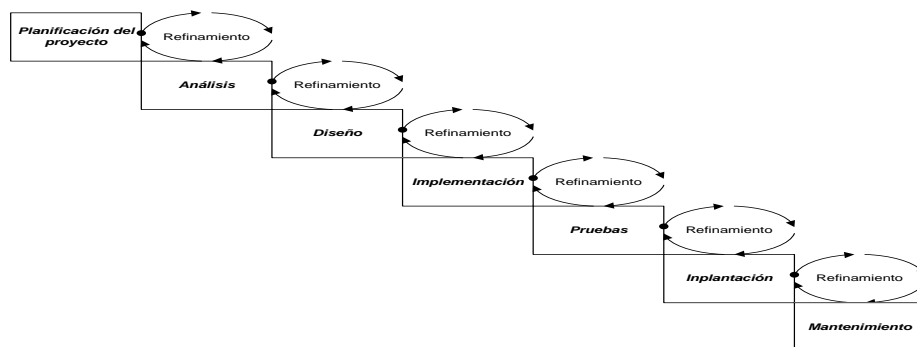


Ilustración 1 Ciclo de vida BD

Ciclo de vida de una base de datos:

- Definición del sistema: Requerimientos de datos.
- Diseño de la base de datos.
- Implementación de la base de datos.
- Carga o conversión de los datos.
- Conversión de aplicaciones.
- Prueba y validación.
- Operación, supervisión y mantenimiento.

1.4 Planificación con hitos y temporalización

1.4.1 Entregas parciales – PACs - .

A lo largo de curso se confeccionan y entregan 3 PACs –pruebas de evaluación continuas- que posibilitan la realización parcial y progresiva de los objetivos propuestos.

- PAC 1.- Referente a la confección del plan de trabajo, describiendo las tareas a realizar para el concreto desarrollo, con la planificación temporal
- PAC 2.- Corresponderá a las fases de Análisis y diseño de la Base de Datos.
- PAC 3.- Implementación y pruebas del Producto BD.
- Presentación final, entrega del producto.

1.4.2 Planificación Temporal

Diagrama de Gant

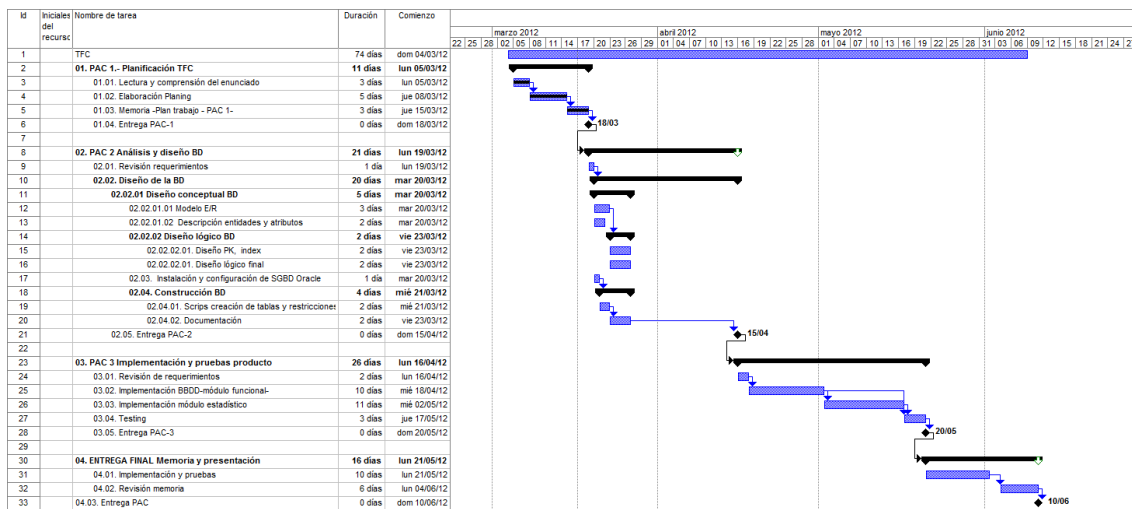


Ilustración 2 Diagrama de Gant

1.4.3 Hitos del proyecto

Tabla 1 Hitos del proyecto

Fecha	Nombre del Hito	Descripción
29/02/2012	Inicio del curso	Fecha de inicio del curso
05/03/2012	Trovada	Encuentro presencial -orientación académica-
18/03/2012	PAC 1	Entrega de la primera parte de la memoria
15/04/2012	PAC 2	Entrega de la segunda parte de la memoria
20/05/2012	PAC 3	Entrega de la tercera parte de la memoria
21/05/2012	Inicio final TFC	Inicio enunciado del final de TFC
10/06/2012	Entrega final TFC	Final entrega producto, pruebas y memoria

1.4.4 Tabla de fechas

Tabla 2 Tabla de fechas

TFC	74 días	04/03/2012 9:00	10/06/2012 19:00
01. PAC 1.- Planificación TFC	11 días	05/03/2012 9:00	18/03/2012 19:00
01.01. Lectura y comprensión del enunciado	3 días	05/03/2012 9:00	07/03/2012 19:00
01.02. Elaboración Planning	5 días	08/03/2012 9:00	14/03/2012 19:00
01.03. Memoria -Plan trabajo - PAC 1-	3 días	15/03/2012 9:00	18/03/2012 19:00
01.04. Entrega PAC-1	0 días	18/03/2012 19:00	18/03/2012 19:00
02. PAC 2 Análisis y diseño BD	21 días	19/03/2012 9:00	15/04/2012 19:00
02.01. Revisión requerimientos	1 día	19/03/2012 9:00	19/03/2012 19:00
02.02. Diseño de la BD	20 días	20/03/2012 9:00	15/04/2012 19:00
02.02.01 Diseño conceptual BD	5 días	20/03/2012 9:00	26/03/2012 19:00
02.02.01.01 Modelo E/R	3 días	20/03/2012 9:00	22/03/2012 19:00
02.02.01.02 Descripción entidades y atributos	2 días	20/03/2012 9:00	21/03/2012 19:00
02.02.02 Diseño lógico BD	2 días	23/03/2012 9:00	26/03/2012 19:00
02.02.02.01. Diseño PK, índice	2 días	23/03/2012 9:00	26/03/2012 19:00
02.02.02.01. Diseño lógico final	2 días	23/03/2012 9:00	26/03/2012 19:00
02.03. Instalación y configuración de SGBD Oracle	1 día	20/03/2012 9:00	20/03/2012 19:00
02.04. Construcción BD	4 días	21/03/2012 9:00	26/03/2012 19:00
02.04.01. Scripts creación de tablas y restricciones	2 días	21/03/2012 9:00	22/03/2012 19:00
02.04.02. Documentación	2 días	23/03/2012 9:00	26/03/2012 19:00
02.05. Entrega PAC-2	0 días	15/04/2012 19:00	15/04/2012 19:00
03. PAC 3 Implementación y pruebas producto	26 días	16/04/2012 9:00	20/05/2012 19:00
03.01. Revisión de requerimientos	2 días	16/04/2012 9:00	17/04/2012 19:00
03.02. Implementación BBDD-módulo funcional-	10 días	18/04/2012 9:00	01/05/2012 19:00
03.03. Implementación módulo estadístico	11 días	02/05/2012 9:00	16/05/2012 19:00
03.04. Testing	3 días	17/05/2012 9:00	20/05/2012 19:00
03.05. Entrega PAC-3	0 días	20/05/2012 19:00	20/05/2012 19:00
04. ENTREGA FINAL Memoria y presentación	16 días	21/05/2012 9:00	10/06/2012 19:00
04.01. Implementación y pruebas	10 días	21/05/2012 9:00	01/06/2012 19:00
04.02. Revisión memoria	6 días	04/06/2012 9:00	10/06/2012 19:00
04.03. Entrega PAC	0 días	10/06/2012 19:00	10/06/2012 19:00

1.4.5 Identificación de tareas

Tabla 3 Identificación de tareas

T1	Descripción del Trabajo Final de Carrera.	Corresponde a la descripción del proyecto que se debe desarrollar.
T2	Objetivos del proyecto.	Se definen los objetivos que se pretenden conseguir.
T3	Identificación de las tareas.	División en partes en las que podemos dividir el proyecto.
T4	Planificación temporal.	Establecimiento de tiempos parciales.
T5	Recursos humanos, técnicos y materiales.	Miembros y medios empleados en el proyecto.
T6	Análisis de riesgo.	Definición posibles riesgos que puedes retrasar la ejecución del TFC.
T7	Valoración Inicial del coste.	Valoración aproximada del tiempo de ejecución TFC.
T8	Análisis de los requisitos.	Definir funciones de la BD.
T9	Diseño.	Diseño estructural BD.
T9_1.1	Modelo UML	Representar la estructura.
T9_1.2	Descripción de entidades y atributos	Enumeración entidades y atributos.
T9_2	Diseño Lógico	Transformación del modelo E/R en un modelo relacional.
T9_2.1	Identificación de Claves primarias y foráneas	Del modelo UML se obtiene las relaciones con sus claves primarias y foráneas.
T9_2.2	Modelo UML	La representación definitiva de la estructura según el modelo E/R.
T9_2.3	Restricciones	Se definirán las reglas que debe cumplir el sistema de base de datos, tanto las reglas de integridad como las de restricciones.
T10	Diseño Físico	Transformación del modelo físico teniendo en cuenta la implementación física de la base de datos.
T11	Implementación	Generar código para crear la BD.

1.5 Valoración inicial del coste

Consideramos las siguientes variables.

7,5 créditos / 25 horas = 187,5 horas
+ 10% de coeficiente por impericia = 208 horas. Aproximadamente 3 horas/día de dedicación.

De un total de 14 semanas de trabajo, consideramos 70 días laborables y un incremento de un día mas por cada entregable, por lo que dedicaremos un periodo de 74 días para el desarrollo del producto.

Para el desarrollo del proyecto, al contar con el soporte UOC. Guía y corrección de los consultores – asignatura y laboratorio -, podemos acotar el cálculo de la facturación a 2 horas /día.

Tabla 4 Horas facturables

Nombre del Hito	Días naturales	Horas invertidas facturables
PAC 1	11	22
PAC 2	21	42
PAC 3	26	52
Entrega final TFC	16	32
Total	74	148

Teniendo en cuenta las siguientes tarifas

Tabla 5 Tarifas

Recurso	Ratio Coste / hora	Coste / hora asumido
Jefe de Proyecto	80 – 100 € / hora	90 € / hora
Analista Funcional	40 – 50 € / hora	45 € / hora
Analista Programador	30 € / hora	30 € / hora

1.5.1 Desglose del coste económico.

Tabla 6 Coste económico

Actividad	Horas	Responsable	Coste (€)
Gestión y Planificación - PAC 1-	22	Jefe de Proyecto	1.980
Análisis -PAC 2-	42	Analista Funcional	1.890
Diseño y programación -PAC 3-	52	Analista Programador	1.560
Pruebas, documentación -Entrega final TFC-	32	Analista Programador	960
Total	148		6.390

No consideramos el coste de formación del usuario, no interaccionará directamente con la BD.

La carga de datos – analista programador- y la puesta en producción –analista de sistemas, aprox. 45€/hora-.

Otros costes a considerar.

Licencia de Oracle 10g multiplataforma y multiusuario 7.000€. (10 puestos de trabajo, considerando la amplia utilización para los departamentos correspondientes de la UE).


Por lo que tendríamos un coste aproximado de 13.400€

1.6 Recursos humanos técnicos y materiales

- Personas integrantes:
 - El consultor D. Manuel Rella Ruiz, en calidad de cliente-interlocutor para acudir al concurso de la CEE. Será la persona encargada de resolver las dudas que vayan surgiendo.
 - El alumno D. Alberto Ramos Saguillo, en calidad de responsable de la empresa de desarrollo.

En cuanto al Hardware utilizado:

Sistema

Fabricante:	TOSHIBA
Modelo:	Personal Computer
Evaluación:	 Evaluación de la experiencia en Windows
Procesador:	Intel(R) Core(TM) i5 CPU M 460 @ 2.53GHz 2.53 GHz
Memoria instalada (RAM):	4,00 GB (3,73 GB utilizable)
Tipo de sistema:	Sistema operativo de 64 bits

Edición de Windows

Windows 7 Home Premium

Programas utilizados:

- Memoria: Microsoft Office 2003.
- Diagramas: Microsoft Project 2003
- SGBD -Sistema de Gestión de Base de Datos Relacionales-: Oracle Express v10.2.0.1
- Diagramas E/R: Magic Draw o Microsoft Visio 2003/2010, ambos utilizados en asignaturas realizadas anteriormente.
- Desarrollo SQL y PL/SQL: SQL Developer v1.0.015.
- Presentación: Microsoft PowerPoint 2003.

1.7 Análisis de riesgos

Como en toda actividad, pueden surgir imprevistos, que pueden alterar seriamente el desarrollo de la misma. Hemos de tomar las medidas oportunas que minimizar estas situaciones ya que no hay forma de evitarlas. Al menos reduciremos su impacto.

- Nivel técnico. Se tomarán todas las medidas posibles para no perder información; copia de seguridad por cada sesión de trabajo en unidades USB y en disco virtual -dropbox-.
 - La elaboración del documento elaborando versiones correlativas, minimizando el riesgo de pérdida en caso de desastre imprevisto.
 - La política de copia seguida es la de una por cada día de la semana que contendrá las sesiones correspondientes.
 - Nivel personal. Salvo enfermedad o accidente, no consideramos especialmente ningún factor añadido, sólo las cargas de trabajo extras que en la actividad diaria, y que puedan tener incidencia.
- Prevención.- Nivel personal- . Como medidas a tomar a nivel preventivo y teniendo en cuenta la posible complejidad del desarrollo del código final y de posibles imprevistos, trato de avanzar el cumplimiento del contenido de la PAC 2.

2. ANÁLISIS DE REQUISITOS

Descripción inicial

La CE pretende tener una base de datos para tener datos objetivos sobre los consumos, tanto de las centrales de distribución (que alimentan a los contadores), como de cada central de producción que alimenta a sus correspondientes centrales de distribución.

Se pretende tener en tiempo real, datos permanentemente actualizados, la energía producida por las centrales, y los consumos de los contadores, parametrizados según superen las medias establecidas o bien ranking de los mayores consumidores.

El sistema debe dar respuesta a la funcionalidad de las peticiones, basadas en insertar, modificar datos críticos o combinaciones de los mismos.

Por lo consideramos como puntos base los Clientes, Contadores, Contratos, Líneas de Conexión, Centrales de Distribución, Centrales de Producción, Tipos Energías, Control de Inspecciones, Lecturas de Consumos, Localización física (geográfica) de cada una de las entidades, Histórico de datos y Estadísticas. Considerado este último módulo estadístico como el producto actualizado y real de las inserciones producidas en las correspondientes entidades / relaciones según los requerimientos iniciales del proyecto.

Supuestos iniciales

A priori no conocemos el dato concreto de la energía producida, por lo que validaremos el valor de producción, como el consumo generado por los contadores. Como tolerancia tendremos el rango máximo que puede soportar como capacidad máxima. Este dato será decisivo para controlar, no sólo las posibles caídas de tensión en red, sino para gestionar la asignación de contadores a las líneas correspondientes que deberán tener acceso a las Centrales de Distribución.

Por simplificar el modelo consideraremos que los contadores de alta disponibilidad, es decir conectados a dos centrales, computan su consumo al 50% en cada una de ellas.

También en Modelo, consideramos que un modelo determinado puede llevar de serie u opcionalmente el módulo telemático o para conexión dual, y en Contador, sí realmente esa unidad lo lleva instalado. Para hacer la lectura telemática comprobaremos que el contador, como restricción lleva instalado el módulo telemático.

Los contadores pueden soportar cualquier potencia contratada, todo que el modelo puede contemplar la gestión de diferentes modelos, con lo que sus especificaciones pueden variar y se podrían gestionar.

Tampoco tendremos en cuenta posibles pérdidas de energía en el transporte ni por la gestión de las Centrales ni demás elementos.

Requisitos funcionales

El sistema diseñado permite las siguientes funcionalidades:

- Gestión de Centrales: Alta, baja y modificación.
- Gestión de tipos de energía: Alta y mantenimiento.
- Gestión de Líneas de comunicación: Alta, baja y modificación.
- Gestión de Contadores: Alta, baja y modificación.
- Gestión de tipos de Centrales: Alta, baja, modificación y gestión de toda la información asociada de impacto medio ambiental.
- Gestión de Clientes: Alta, baja y modificación.
- Gestión de fabricantes: Alta, baja y modificación.
- Gestión de Contratos: Alta, baja y modificación.
- Gestión de Países: Alta, baja y modificación.
- Gestión de Localidad: Alta, baja y modificación.
- Gestión de Ubicaciones: Alta, baja y modificación.
-

Remarcar que las bajas serán lógicas y no físicas, por tanto pasarán del estado alta=1 a baja=0.

- Histórico de Inspecciones, ordenas por fecha con resultado y observaciones de las mismas.

Procedimientos de consulta que permitan:

- Listado de los contadores, cuyo consumo supere el 80% del consumo medio mensual en ese periodo de tiempo considerado.
- Listado de las 10 Centrales de distribución, que suministran más energía.
- Listado de las 10 líneas de comunicación más cargadas en relación a su potencia máxima
- Listado de clientes que tienen contador con servicio de alta disponibilidad (el modelo tiene módulo dual y está conectado)
- Listado del consumo los contadores de una central y la energía producida por esta en un periodo de tiempo.
- Listado de porcentajes de las lecturas de contador, efectuadas de forma presencial y de forma telemática en un periodo de tiempo.
- Listado de contadores que tengan una determinada antigüedad (expresado en años).

Módulo estadístico, deberá dar respuesta en tiempo real:

- E1.- Dada una Central de producción, el consumo de los contadores que dependen de la misma. El acceso se hará directamente al registro correspondiente actualizado mediante el código de la Central.
- E2.- Línea de comunicación con el valor medio de energía consumida en un año concreto, teniendo en cuenta que la conexión dual del contador computa en las dos líneas.
- E3.- Línea más cargada, a nivel de energía consumida.
- E4.- Porcentaje de líneas que superan el 50% de energía consumida en un año concreto.
- E5.- Número de Centrales que generan menos del 30% de la producción en un año concreto.
- E6.- El ranking de los 10 contadores que históricamente han tenido más consumo.
- E7.- El consumo medio de todos los clientes.

Otros requisitos

Para el desarrollo del sistema, además de los requisitos señalados anteriormente, se deberá ajustar a los siguientes requerimientos:

- El sistema de Gestión de Base de Datos escogido será Oracle.
- Portabilidad del sistema sobre hardware convencional, no específico.
- Integridad y consistencia de la Base de Datos, buscando las normalizaciones standards, evitando redundancia de datos y mantenimiento complejo.
- Eficiencia en la gestión del espacio de ocupación de memoria y disco.
- Los procedimientos almacenados contarán con el tratamiento de las excepciones que puedan producirse al computar entradas no esperadas.
- La política de permisos seguida, es común a todos los usuarios, tendrán acceso a todas las funcionalidades; por lo que no se generan perfiles específicos.
- No se imponen restricciones al sistema operativo.
- No se contempla la realización de tareas periódicas de backup de los datos almacenados en la Base de Datos.
- Se contempla la posibilidad de crear una tabla LOG en la Base de Datos para tener constancia de las transacciones correspondientes a las Centrales – Producción y Distribución-.
- En sistema llevará un control depurativo de excepciones que pueden generarse durante la manipulación de los datos o en la introducción de los mismos.

Requisitos no funcionales

- El diseño de este sistema no contempla el de una interficie gráfica para el usuario final del producto.
- El sistema debe tener un comportamiento estable, informando de la correcta ejecución o bien dando el mensaje codificado de error, que podrá consultarse en la tabla correspondiente.
- Los accesos a la base de datos se realizan utilizando el entorno PL/SQL Developer, o bien utilizando los procedimientos creados para el resto de las funcionalidades.

3. DISEÑO DEL SISTEMA

Abordaremos el diseño del sistema en tres etapas:

1. Diseño conceptual: buscamos un diseño independiente de la tecnología que vamos a utilizar y obtendremos un diagrama UML como resultado de esta etapa.
2. Diseño lógico: transformaremos el modelo UML, utilizando como tecnología un SGBD relacional.
3. Diseño físico: buscaremos optimizar la eficiencia y completar el modelo con aspectos físicos dependientes del SGBD concreto utilizado.

Diseño conceptual

En el diseño conceptual contemplamos las entidades, sus atributos, claves primarias (pero no externas), relaciones entre entidades y sus cardinalidades. Las relaciones pueden ser simples, asociativas, de agregación, de composición, de generalización o de especialización. Las entidades pueden ser principales, auxiliares o asociativas. Se ha intentado que las relaciones se nombren de forma explícita.

- **Esquema conceptual.**
 - Desarrollamos el esquema conceptual de la organización de la BD independientemente del Sistema Gestor de Base de Datos (SGBD).
- **Esquema de base de datos.**
 - Representamos los datos en un formato más próximo al que el Sistema tendrá en el hardware instalado.
- **Esquema Interno.**
 - Representa los datos según el modelo concreto de SGBD, en nuestro caso Oracle 10g.
- **Base de Datos física.**
 - Los datos son almacenados en el disco interno del Host o bien en un Sistema de Bases de Datos distribuidas en diferentes Host interconectados en distintos lugares.

Diagrama UML Conceptual

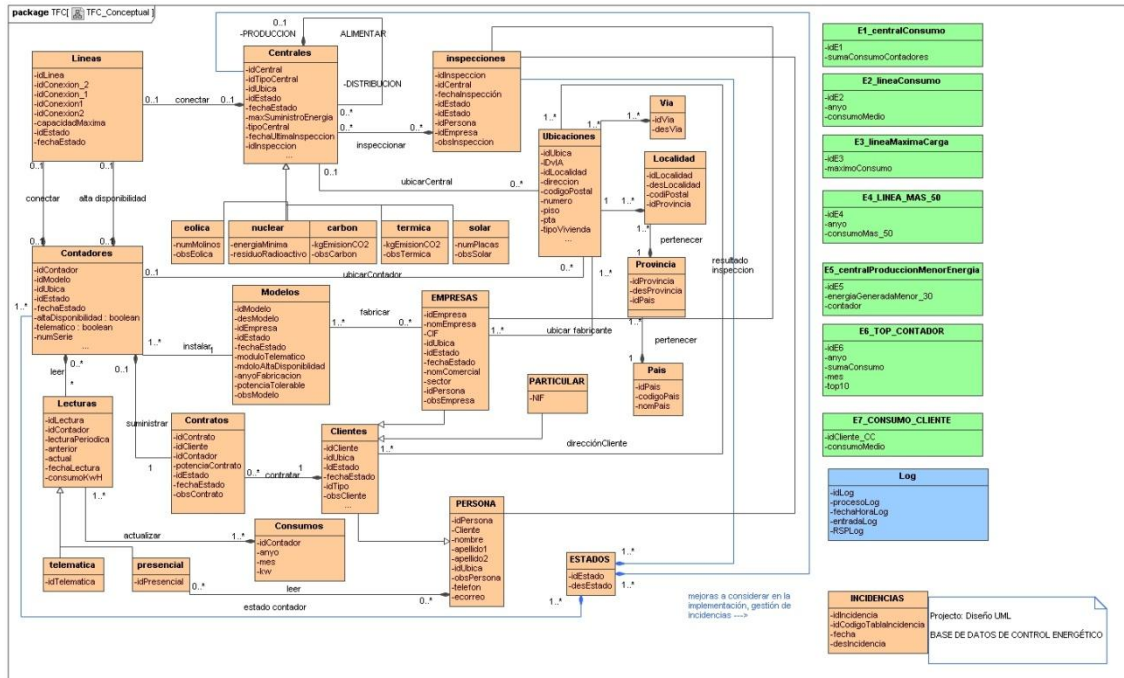


Ilustración 3 Diagrama UML –Entidad/relación-

Identificación de las entidades

Por orden alfabético:

Tabla 7 Tabla de entidades

NOMBRE	resumen descripción contenido
---------------	--------------------------------------

CENTRALES	datos correspondientes a cada Central (Producción =0 OR Distribución=1)
CLIENTES	datos del cliente y su domicilio -código cliente y ubicación-, identificadores de particular o empresa.
CONSUMOS	datos históricos del consumo acumulado por año, mes de cada contador
CONTADORES	datos del contador concreto, indicando código modelo y número de serie. Su ubicación y su estado.
CONTRATOS	datos de la modalidad de contratación, potencia contratada con un contador determinado y observaciones. Por simplificar el modelo no consideramos una entidad para los tipos de contrato.
EMPRESAS	datos relativos a los fabricantes de equipos, componentes o empresas inspectoras.
ESTADOS	datos relativos a los códigos de uso, alta, baja o futuras funcionalidades. En principio consideramos pocos valores, pero dejamos previsión para futuros crecimientos o funcionalidades.
INSPECCIONES	datos históricos de las inspecciones, puede tener datos de central o contadores, persona que la hace y resultado. En la Central la fecha de inicio vendrá marcada por la fecha de alta en la inspección. Tendremos la empresa y persona que realiza el trabajo.
LECTURAS	datos de las lecturas de los contadores, indicando el método como se efectúan - telemático o presencial- si es de instalación o es lectura de consumo periódico. num. Kw. Simplemente con el valor de lectura periódica -instalación=0 OR periódica=1-, podremos tener los consumos desde la primera, última instalación o a partir de una fecha determinada.
LINEAS	datos identificativos de la línea, indicando capacidad máxima, estado de uso y fecha de ultimo estado. Contendrá los datos del código de contador y de central conectados.
LOCALIDAD	datos de descripción de la Localidad, código de la Provincia.
MODELO	datos del modelo concreto con la descripción y características, módulos instalados, año de fabricación y código de empresa
PAIS	datos correspondientes a cada país Europeo, según la norma ISO 3166-1
PERSONA	datos de todas las personas relacionadas con el proyecto, clientes, empleados, colaboradores, proveedores, inspectores etc.
PROVINCIA	Datos relativos a la Provincia y el código del País
UBICACION	Datos, código de ubicación, dirección física y el código de la Localidad, código de la vía y el código postal.
VIA	Contiene el conjunto de abreviaturas que identifican las vías públicas.

Identificación de Relaciones

Central Distribución – Central Producción

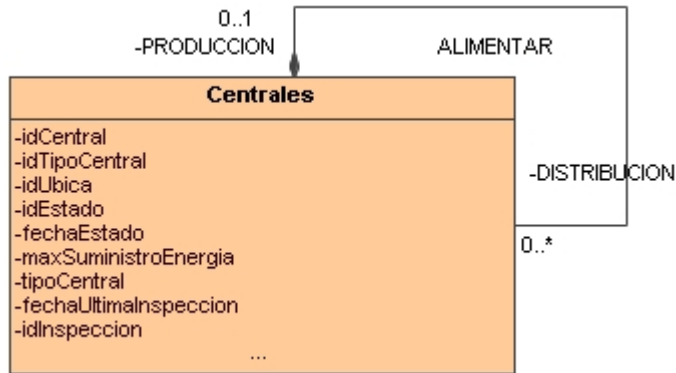


Ilustración 4 Central Producción / Central Distribución

Establecemos una relación reflexiva, dado que las ocurrencias de la entidad se pueden relacionar entre ellas. Tendremos el rol correspondiente a cada Central (Producción: tipoCentral = 0 OR Distribución: tipoCentral=1). Cada instancia de la Central de Distribución se relaciona con una, si esta conectada, de la Central de Producción o con ninguna, y varias Centrales de Distribución se relacionan con una Central de Producción.

Mediante el código de ubicación –idUbica-, asociado a las entidades correspondientes, podremos gestionar las centrales más próximas geográficamente. Por la fecha de inspección sabremos su fecha de alta, baja o incidencias que ha tenido. Por el atributo idCentral sabremos si esta operativa o no (Baja=0 OR Operativa=1).

Tendremos una especialización por tipo de central, para gestionar las particularidades de cada una.

Podemos tener una entidad que relacione la conexión de las Centrales –ALIMENTA- o bien tratarlas en la entidad LINEAS.

ALIMENTAR (idCentral_P, idCentral_D, idLinea)
{idCentral_P} es clave foránea de CENTRAL
{idCentral_D} es clave foránea de CENTRAL
{idLinea} es clave foránea de LINEAS

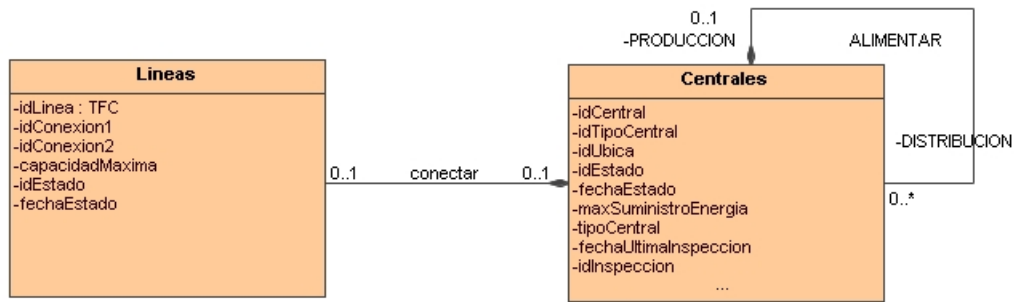


Ilustración 5 Central Producción / Central Distribución - Lineas-

LINEAS (idLinea, idConexion1, idConexion2)
 { idLinea } es foránea de CENTRAL
 { idConexion1 } es clave foránea de CENTRAL
 { idConexion2 } es clave foránea de CENTRAL

Contador – Línea de comunicación - Central Distribución

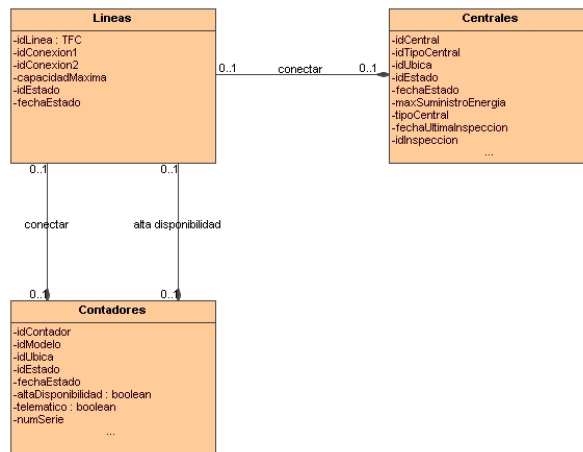


Ilustración 6 Contador conectado

La relación entre Centrales y Contadores es de muchos a muchos, por tanto establecemos una relación de dependencia entre Central y Contador, con una entidad dependiente Líneas, para relacionar las ocurrencias de las entidades.

Una línea de comunicación, identificada por su código y estado (foránea de ESTADOS, podremos tener alta, baja, reparación, estado de conservación etc.), relaciona el contador con la central a la que se conecta, y estableceremos un atributo booleano de alta disponibilidad para reflejar que su conexión es de alta disponibilidad con la Central secundaria.

Tener una clave foránea –idEstado- permite poder tener una gestión de incidencias, que ahora no contemplamos, pero puede implementarse a posteriori. También nos permite gestionar ágilmente cualquier cambio en la codificación de los estados que pueden tener las entidades. (Contadores, Líneas...).

Cliente – Contador- Contratos

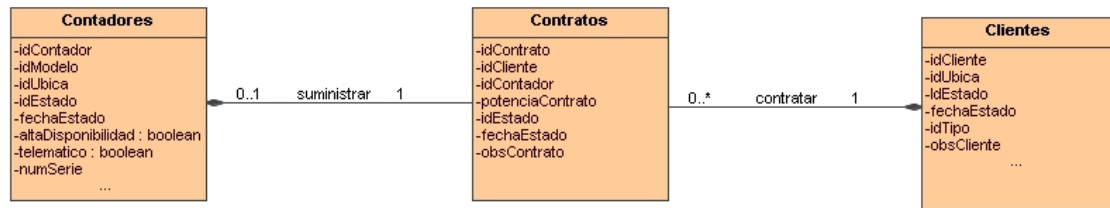


Ilustración 7 Contratos del cliente con contador

Tenemos una relación de muchos a muchos por tanto la relación de dependencia entre Cliente y Contador, con la entidad Contratos. Por cada ocurrencia de Contratos, reflejaremos un contrato del cliente, que tendrá asociado los identificadores únicos del cliente y contador.

Tendremos el rol correspondiente a Cliente (idEstado=0, Baja OR idEstado=1, Alta). También se adoptarán los mismo roles en Contratos (idEstado=0; Baja OR idEstado=1 Alta) y en Contador (idEstado=0; NO disponible OR idEstado=1 Operativo).

Para Cliente contamos con la especialización de Empresa (CIF) o Particular (NIF), como claves alternativas.

Las clases de Cliente y Particular heredan de Persona, así como la lectura Presencial y la persona que hace las inspecciones o contacto con una Empresa determinada. En empresa tendremos un atributo para la persona de contacto. De esta forma las en las búsquedas en la BD encontraremos rápidamente la persona y el rol que desempeña en la BD.

Ubicación de contadores, Centrales y clientes

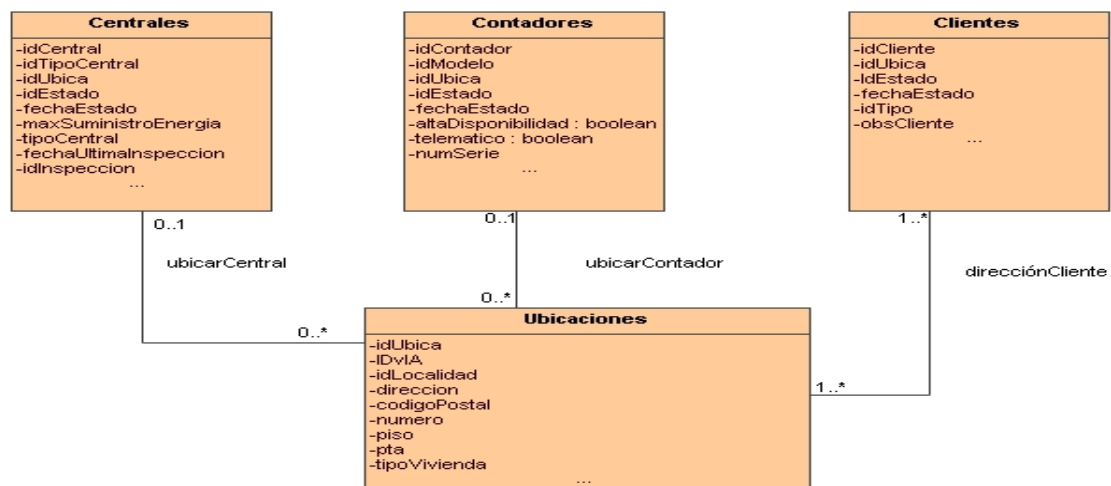
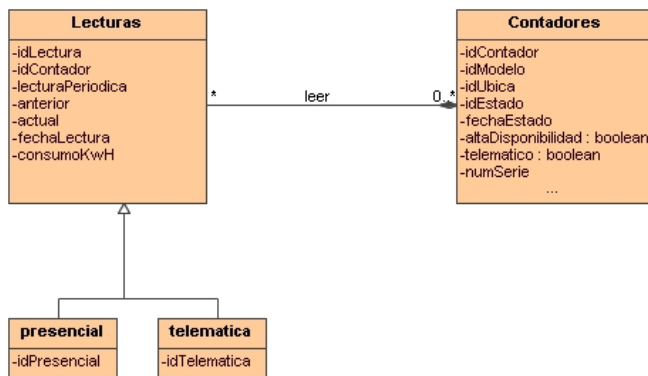


Ilustración 8 Ubicación Centrales, contadores y clientes

Asociado a Clientes, Contadores y Centrales tenemos Ubicación, para poder gestionar y ubicar físicamente tanto los contadores como clientes, debidamente codificadas para optimizar la utilización en la BD.

Lecturas Contador



Tenemos una asociación de muchos a muchos, un contador puede tener muchas lecturas, y podemos tener contadores que no hemos leído.

En Lecturas tenemos la especialización de Telemática o Presencial. En Contador tiene que tener instalado el módulo de lectura telemático, y en el Modelo registramos que puede llevar ese módulo. Misma situación para la conexión dual. Ese modelo puede hacerlo, y en contador lo tenemos activo, o lo hemos instalado.

Lecturas Consumo

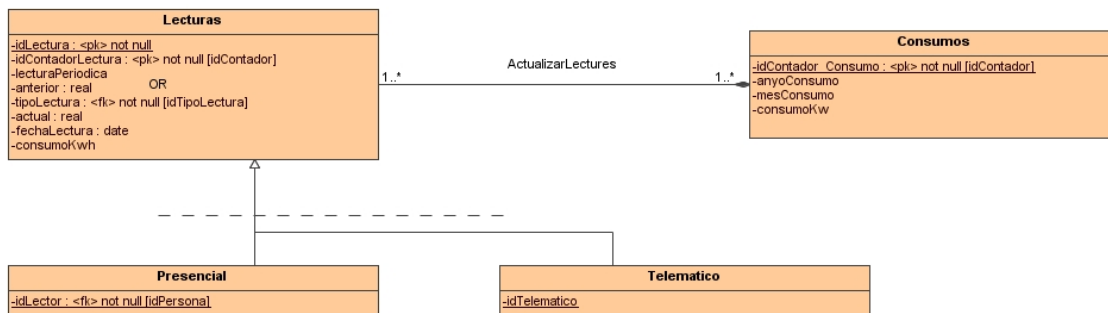


Ilustración 9 Lectura – Consumo

Asociación de uno a muchos, por cada lectura de contador actualizaremos su consumo mensual y actualizaremos el contador de Kwh.

De Consumo tenemos la relación con la E7_consumosClientes para satisfacer un requisito del enunciado y actualizar el consumo medio del cliente.

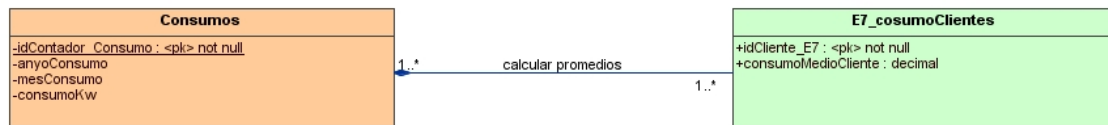


Ilustración 10 Consumo medio por cliente

Inspecciones Central



Ilustración 11 Central Inspecciones

Asociación de uno a muchos, cada instancia de Central se relaciona con muchas inspecciones.

En inspecciones también codificamos el resultado de la inspección. En principio consideramos 0=No válida OR 1=Correcto, pero también podemos considerar tener relación con la entidad Estados y poder codificar cualquier situación. La Empresa inspectora como clave foránea que puede tener valores nulos, también podemos codificar los inspectores intervinientes como clave foránea que puede contener nulos.

De la misma forma podríamos gestionar las inspecciones de los contadores si fuera requisito posterior.

Ubicar –Centrales, contadores, clientes, empresas

Para poder codificar adecuadamente las direcciones de los clientes etc. establecemos la relación con la entidad Ubicaciones.

De esta forma podremos parametrizar las búsquedas en la Base de forma rápida y fiable. Exige entrada normalizada de nombres.

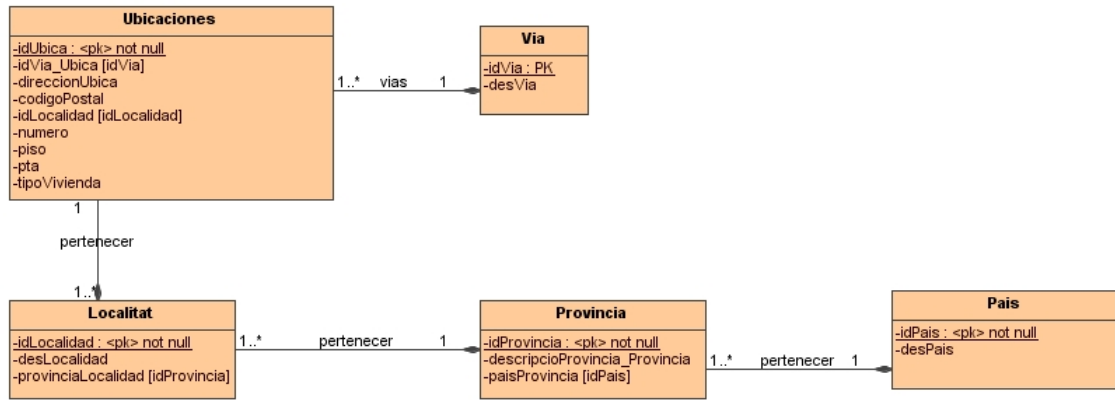


Ilustración 12 Ubicaciones

Atributos de entidades

Los atributos detectados tras el análisis preliminar son los siguientes –claves primarias subrayadas-

Tabla 8 Atributos de Entidades

NOMBRE	resumen descripción contenido
--------	-------------------------------

<p>CENTRALES</p>	<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">Centrales</p> <p>-idCentral : <pk> not null -idTipoCentral : not null -idUbica : <fk> not null -idEstado : <fk> not null -fechaEstado : not null -maxSuministroEnergia : not null -fechaUltimaInspeccion : not null -idInspeccion : <fk> not null ...</p> </div>
<p>CLIENTES</p>	<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">Cientes</p> <p>-idCliente : <pk> not null -idUbica : <fk> not null -idEstado : <fk> not null -fechaEstado : not null -idTipo : not null -obsCliente : null ...</p> </div>
<p>CONSUMOS</p>	<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">Consumos</p> <p>-idContador : <pk> not null -anyo : not null -mes : not null -kw : not null</p> </div>
<p>CONTADORES</p>	<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">Contadores</p> <p>-idContador : <pk> not null -idModelo : <fk> not null -idUbica : <fk> not null -idEstado : <fk> not null -fechaEstado : not null -altaDisponibilidad : boolean -telematico : boolean -numSerie : not null ...</p> </div>
<p>CONTRATOS</p>	<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">Contratos</p> <p>-idContrato : <pk> not null [TFC] -idCliente : <fk> not null -idContador : <fk> not null -potenciaContrato : not null -idEstado : <fk> not null -fechaEstado : not null -obsContrato : null</p> </div>

<p>EMPRESAS</p>	<table border="1"> <tr> <td data-bbox="852 210 1169 510"> <p>EMPRESAS</p> <ul style="list-style-type: none"> -idEmpresa : <pk> not null [TFC] -nomEmpresa : not null -CIF : <ak not null> -idUbica : <fk> not null -idEstado : <fk> not null -fechaEstado : not null -nomComercial : not null -idPersona : <fk> not null -obsEmpresa : null -sector : not null </td> </tr> </table>	<p>EMPRESAS</p> <ul style="list-style-type: none"> -idEmpresa : <pk> not null [TFC] -nomEmpresa : not null -CIF : <ak not null> -idUbica : <fk> not null -idEstado : <fk> not null -fechaEstado : not null -nomComercial : not null -idPersona : <fk> not null -obsEmpresa : null -sector : not null
<p>EMPRESAS</p> <ul style="list-style-type: none"> -idEmpresa : <pk> not null [TFC] -nomEmpresa : not null -CIF : <ak not null> -idUbica : <fk> not null -idEstado : <fk> not null -fechaEstado : not null -nomComercial : not null -idPersona : <fk> not null -obsEmpresa : null -sector : not null 		
<p>ESTADOS</p>	<table border="1"> <tr> <td data-bbox="852 546 1152 658"> <p>ESTADOS</p> <ul style="list-style-type: none"> -idEstado : <pk> not null [TFC] -desEstado : null </td> </tr> </table>	<p>ESTADOS</p> <ul style="list-style-type: none"> -idEstado : <pk> not null [TFC] -desEstado : null
<p>ESTADOS</p> <ul style="list-style-type: none"> -idEstado : <pk> not null [TFC] -desEstado : null 		
<p>INSPECCIONES</p>	<table border="1"> <tr> <td data-bbox="852 703 1134 981"> <p>inspecciones</p> <ul style="list-style-type: none"> -idInspeccion : <pk> not null -idCentral : <fk> not null -fechaInspección : not null -idEstado : <fk> not null -idEstado : <fk> not null -idPersona : <fk> not null -idEmpresa : <fk> not null -obsInspeccion : null </td> </tr> </table>	<p>inspecciones</p> <ul style="list-style-type: none"> -idInspeccion : <pk> not null -idCentral : <fk> not null -fechaInspección : not null -idEstado : <fk> not null -idEstado : <fk> not null -idPersona : <fk> not null -idEmpresa : <fk> not null -obsInspeccion : null
<p>inspecciones</p> <ul style="list-style-type: none"> -idInspeccion : <pk> not null -idCentral : <fk> not null -fechaInspección : not null -idEstado : <fk> not null -idEstado : <fk> not null -idPersona : <fk> not null -idEmpresa : <fk> not null -obsInspeccion : null 		
<p>LECTURAS</p>	<table border="1"> <tr> <td data-bbox="852 1016 1160 1249"> <p>Lecturas</p> <ul style="list-style-type: none"> -idLectura : <pk> not null -idContador : <pk> not null -lecturaPeriodica : not null -anterior : not null -actual : not null -fechaLectura : not null -consumoKwH : not null </td> </tr> </table>	<p>Lecturas</p> <ul style="list-style-type: none"> -idLectura : <pk> not null -idContador : <pk> not null -lecturaPeriodica : not null -anterior : not null -actual : not null -fechaLectura : not null -consumoKwH : not null
<p>Lecturas</p> <ul style="list-style-type: none"> -idLectura : <pk> not null -idContador : <pk> not null -lecturaPeriodica : not null -anterior : not null -actual : not null -fechaLectura : not null -consumoKwH : not null 		
<p>LINEAS</p>	<table border="1"> <tr> <td data-bbox="852 1285 1262 1496"> <p>Lineas</p> <ul style="list-style-type: none"> -idLinea : <pk> not null -idConexion1 : <pk> not null -idConexion2 : <fk> not null -capacidadMaxima : not null -idEstado : <pk> not null -fechaEstado : not null </td> </tr> </table>	<p>Lineas</p> <ul style="list-style-type: none"> -idLinea : <pk> not null -idConexion1 : <pk> not null -idConexion2 : <fk> not null -capacidadMaxima : not null -idEstado : <pk> not null -fechaEstado : not null
<p>Lineas</p> <ul style="list-style-type: none"> -idLinea : <pk> not null -idConexion1 : <pk> not null -idConexion2 : <fk> not null -capacidadMaxima : not null -idEstado : <pk> not null -fechaEstado : not null 		
<p>LOCALIDAD</p>	<table border="1"> <tr> <td data-bbox="852 1532 1123 1720"> <p>Localidad</p> <ul style="list-style-type: none"> -idLocalidad : <pk> not null -desLocalidad : not null -codiPostal : not null -idProvincia : <fk> not null </td> </tr> </table>	<p>Localidad</p> <ul style="list-style-type: none"> -idLocalidad : <pk> not null -desLocalidad : not null -codiPostal : not null -idProvincia : <fk> not null
<p>Localidad</p> <ul style="list-style-type: none"> -idLocalidad : <pk> not null -desLocalidad : not null -codiPostal : not null -idProvincia : <fk> not null 		

<p>MODELOS</p>	<div style="border: 1px solid black; padding: 5px; background-color: #f9cb9c;"> <p style="text-align: center;">Modelos</p> <p>-idModelo : <pk> not null -desModelo : not null -idEmpresa : <fk> not null -idEstado : <fk> not null -fechaEstado : not null -moduloTelematico : not null -mdoloAltaDisponibilidad : not null -anyoFabricacion : not null -potenciaTolerable : not null -obsModelo : null</p> </div>
<p>PAIS</p>	<div style="border: 1px solid black; padding: 5px; background-color: #f9cb9c;"> <p style="text-align: center;">Pais</p> <p>-idPais : <pk> not null -codigoPais : <ak not null> -nomPais : not null</p> </div>
<p>Particular</p>	<div style="border: 1px solid black; padding: 5px; background-color: #f9cb9c;"> <p style="text-align: center;">particular</p> <p>-NIF : <ak not null></p> </div>
<p>PERSONA</p>	<div style="border: 1px solid black; padding: 5px; background-color: #f9cb9c;"> <p style="text-align: center;">PERSONA</p> <p>-idPersona : <pk> not null [TFC] -Cliente : not null -nombre : not null -apellido1 : not null -apellido2 : not null -idUbica : <pk> not null -obsPersona : null -telefon : not null -ecorreo : not null</p> </div>
<p>PROVINCIA</p>	<div style="border: 1px solid black; padding: 5px; background-color: #f9cb9c;"> <p style="text-align: center;">Provincia</p> <p>-idProvincia : <pk> not null -desProvincia : not null -idPais : <fk> not null</p> </div>

<p>TIPOS DE CENTRAL</p>	<table border="1"> <tr> <td>eolica</td> </tr> <tr> <td>-numMolinos : not null -obsEolica : not null</td> </tr> <tr> <td>nuclear</td> </tr> <tr> <td>-energiaMinima : not null -residuoRadioactivo : not null</td> </tr> <tr> <td>carbon</td> </tr> <tr> <td>-kgEmisionCO2 : not null -obsCarbon : not null</td> </tr> <tr> <td>termica</td> </tr> <tr> <td>-kgEmisionCO2 : not null -obsTermica : not null</td> </tr> <tr> <td>solar</td> </tr> <tr> <td>-numPlacas : not null -obsSolar : not null</td> </tr> </table>	eolica	-numMolinos : not null -obsEolica : not null	nuclear	-energiaMinima : not null -residuoRadioactivo : not null	carbon	-kgEmisionCO2 : not null -obsCarbon : not null	termica	-kgEmisionCO2 : not null -obsTermica : not null	solar	-numPlacas : not null -obsSolar : not null
eolica											
-numMolinos : not null -obsEolica : not null											
nuclear											
-energiaMinima : not null -residuoRadioactivo : not null											
carbon											
-kgEmisionCO2 : not null -obsCarbon : not null											
termica											
-kgEmisionCO2 : not null -obsTermica : not null											
solar											
-numPlacas : not null -obsSolar : not null											
<p>Tipos de lectura</p>	<table border="1"> <tr> <td>telematica</td> </tr> <tr> <td>-idTelematica : not null</td> </tr> <tr> <td>presencial</td> </tr> <tr> <td>-idPresencial : not null</td> </tr> </table>	telematica	-idTelematica : not null	presencial	-idPresencial : not null						
telematica											
-idTelematica : not null											
presencial											
-idPresencial : not null											
<p>UBICACIONES</p>	<table border="1"> <tr> <td>Ubicaciones</td> </tr> <tr> <td>-idUbica : <pk> not null -IDvIA : <fk> not null -idLocalidad : <fk> not null -direccion : not null -codigoPostal : not null -numero : not null -piso : not null -pta : not null -tipo/Vivienda : not null ...</td> </tr> </table>	Ubicaciones	-idUbica : <pk> not null -IDvIA : <fk> not null -idLocalidad : <fk> not null -direccion : not null -codigoPostal : not null -numero : not null -piso : not null -pta : not null -tipo/Vivienda : not null ...								
Ubicaciones											
-idUbica : <pk> not null -IDvIA : <fk> not null -idLocalidad : <fk> not null -direccion : not null -codigoPostal : not null -numero : not null -piso : not null -pta : not null -tipo/Vivienda : not null ...											
<p>VIA</p>	<table border="1"> <tr> <td>Via</td> </tr> <tr> <td>-id/Via : <pk> not null -des/Via : not null</td> </tr> </table>	Via	-id/Via : <pk> not null -des/Via : not null								
Via											
-id/Via : <pk> not null -des/Via : not null											

Diseño Lógico - Relacional

Intenta refinar el diseño anterior, presentándolo de una forma algo más afín, contemplamos:

Claves primarias y externas

Restricciones (PK, FK, NOT NULL)

Las relaciones y su cardinalidad se mantienen igual.

Las relaciones expresan la relación clave primaria externa entre cada lado de la relación.

Para clarificar el dibujo en algunos casos el nombre de la clave externa es el mismo que la clave primaria.

La representación gráfica queda de la siguiente forma:

Diagrama Lógico - Relacional

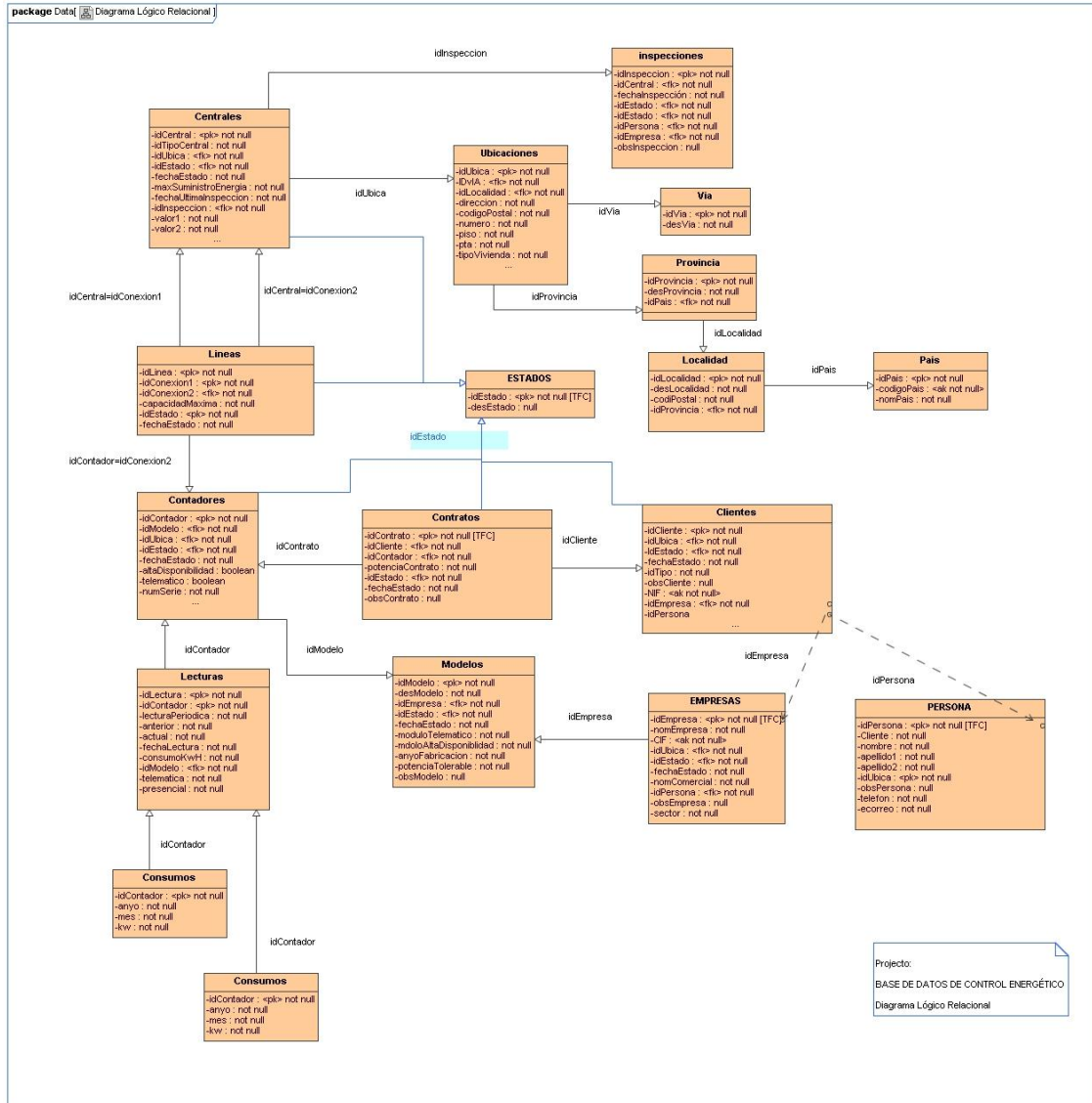


Ilustración 13 Diagrama Lógico-Relacional

NOTA.- Por error Localidad y Provincia están al revés, y en Empresas la punta de flecha también, se corregirá en la siguiente versión

Módulo estadístico

Para satisfacer los requerimientos del módulo estadístico se han creado las siguientes entidades

Tabla 9 Módulo estadístico

<p>E1_centralConsumo (<u>idE1</u>, sumaConsumoContadores)</p> <p>Dada una Central, la suma del consumo de sus contadores.</p>	<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">E1_centralConsumo</p> <p>-idE1 : <pk> not null -sumaConsumoContadores : not null</p> </div>
<p>E2_lineaConsumo (<u>idE2</u>, anyo, consumoMedio)</p> <p>Dada una línea y un año, el valor medio del consumo de los contadores de esta línea.</p>	<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">E2_lineaConsumo</p> <p>-idE2 : <pk> not null -anyo : not null -consumoMedio : not null</p> </div>
<p>E3_lineaMaximaCarga (<u>idE3</u>, maximoConsumo)</p> <p>Línea más cargada a nivel de consumo.</p>	<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">E3_lineaMaximaCarga</p> <p>-idE3 : <pk> not null -maximoConsumo : not null</p> </div>
<p>E4_lineaMas50 (<u>idE4</u>, anyo, consumoMas_50)</p> <p>Dado un año, porcentaje de líneas que superan el 50% de energía consumida</p>	<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">E4_LINEA_MAS_50</p> <p>-idE4 : <pk> not null -anyo : not null -consumoMas_50 : not null</p> </div>
<p>E5_centralProduccionMenorEnergia (<u>idE5</u>, anyo, energiaGeneradaMenor_30, contador30)</p> <p>Dado un año concreto: numero de centrales de producción que generan menos del 30% de producción.</p>	<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">E5_centralProduccionMenorEnergia</p> <p>-idE5 : <pk> not null -energiaGeneradaMenor_30 : not null -contador : not null</p> </div>
<p>E6_topContador (<u>idE6</u>, anyo, mes, sumaConsumo, Top10)</p> <p>Top-10 de contadores que históricamente han tenido más consumo.</p>	<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">E6_TOP_CONTADOR</p> <p>-idE6 : <pk> not null -anyo : not null -sumaConsumo : not null -mes : not null -top10 : not null</p> </div>
<p>E6_topContador (<u>idE7</u>, idCliente, consumoMedio)</p> <p>Consumo medio por cliente</p>	<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">E7_CONSUMO_CLIENTE</p> <p>-idCliente_CC : <pk> not null -consumoMedio : not null</p> </div>

Procedimiento LOG

Log
-idLog : <pk> not null -procesoLog : not null -fechaHoraLog : not null -entradaLog : not null -RSPLog : not null

Ilustración 14 LOG

Registro de todas las llamadas a procedimiento.

Al acabar la ejecución del proceso tendremos los siguientes datos:

- Nombre del procedimiento
- Parámetro de entrada
- Parámetro de salida
- Valor RSP
- Fecha y hora

El valor de RSP, según enunciado lleva un indicador de la realización correcto o incorrecto, constará de un parámetro de salida de tipo String que mostrará el valor "OK" en caso que la ejecución haya finalizado con éxito o "ERROR" + "tipo de error" en caso de error.

Diseño Físico

Corresponde a la creación de la base de datos y de su estructura de tablas, campos tipos de datos, claves primarias externas y relaciones mediante scripts. Siguiendo el enunciado se ha evitado el uso de cualquier herramienta de generación automática.

En esta fase buscaremos los siguientes objetivos:

- Disminuir el tiempo de respuesta de las consultas.
 - Minimizar el espacio que ocupan los datos en disco.
 - Garantizar la máxima seguridad de los datos.
 - Optimizar los recursos del sistema.
 - Para ello partiremos del diseño lógico y aplicaremos los conocimientos concretos que tenemos sobre el SGBD
-
- **Creación de Tablas**
El código correspondiente a este script aparecerá el archivo 2_GESTOR_TABLAS.sql
 - **Creación de Secuencias y Disparadores (triggers)**
El código correspondiente a este script aparecerá el archivo 3_GESTOR_SEQ_TRIG.sql
 - **Creación de Procedimientos (altas, modificaciones, bajas)**
El código correspondiente a este script en los gestores relacionados.
4_GESTOR_ALTAS
5_GESTOR_BAJAS
6_GESTOR_MODIFICACIONES
 - **Creación de Funciones**
Se crean funciones, para clarificar los cálculos en los procedimientos de consultas y estadísticas. El código correspondiente se encuentra en:
7_GESTOR_FUNCIONES

IMPLEMENTACIÓN

Antes de entrar en detalle de cada apartado de la implementación, destacar que todos los procedimientos serán implementados de tal forma, que permitan entrar los datos independientemente de mayúsculas o minúsculas, sin que altere el resultado. Todas las transacciones quedarán registrada en un registro de la tabla LOG, y que se mostrará un mensaje por pantalla del resultado almacenado en dicha tabla.

Para agrupar los procedimientos se utiliza Gestores – agrupar procedimientos y obtener un mejor mantenimiento-.

Se realiza un control exhaustivo de errores. Si se realiza correctamente la operación, devuelve un mensaje de éxito, sino genera un código detallado del error.

Se incorporan utilidades, para reiniciar las consultas y estadísticas.

Los diferentes procesos almacenados que se implementarán, y sus parámetros de entrada se han agrupado según los gestores.

1. GESTOR_FUNCIONES
2. GESTOR_ALTAS
3. GESTOR_MODIFICACION
4. GESTOR_BAJAS
5. GESTOR_CONSULTAS
6. GESTOR_ESTADISTICAS

Para propiciar la reutilización y evitar errores de manipulación se codifican las tablas auxiliares, muy aprovechables para otras aplicaciones. Ej. Ubicaciones (país, provincia, localidad).

Se realizan pruebas exhaustivas del funcionamiento y de todos los procesos, forzando los errores para comprobar las excepciones.

- **Creación de inserciones**

- ✓ Se crean dos juegos de pruebas, el primero verifica el funcionamiento de todos los procedimientos, altas, modificaciones y bajas, provoca situaciones de error para ver los mensajes, y detalla las excepciones.
- ✓ El segundo realiza una carga completa, para comprobar el funcionamiento de la aplicación. Genera las estadísticas, muestra un mensaje por pantalla y llena las tablas correspondientes.

1. Borra_Crea_BD \ 3_Testing.sql
2. Implementacion \ 11_LOAD_INSERTS.sql

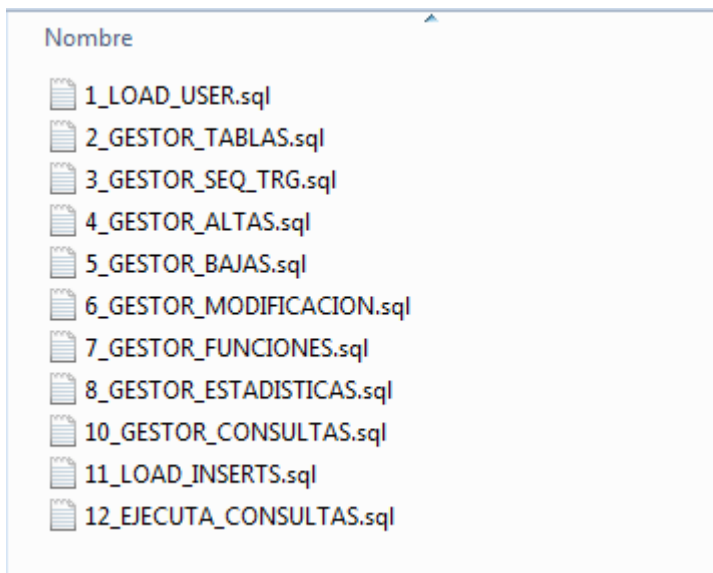
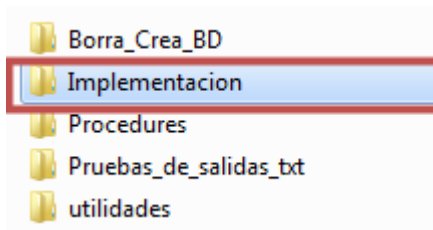
Se crea un apartado específico **Borra_Crea_BD** para poder hacer una carga y descarga rápida de la BD y poder funcionar como una demo.

Ejemplo de secuencias de uso:

A. 2_LOAD_TOTAL
B. Testing.sql
A. 1_DROP_TOTAL
B. 2_LOAD_TOTAL
C. Inserts.sql

La instalación de la aplicación se especifica en el apartado de manual de funcionamiento.

Hay que ejecutar secuencialmente y por orden los scripts detallados en :



Modificaciones

En la evolución tenemos que

Hemos añadido para clarificar el funcionamiento las tablas de

tCentral : tipos de centrales

cCentral: clases de centrales

eCentral: emisiones de las centrales

tLectura: tipos de lectura

tCliente: tipos de cliente

tInspeccion :Inspecciones se ha cambiado por tipo de inspección, y tendremos el tipo de inspección y en Centrales tenemos el campo de la última inspección realizada. Dejamos para posteriores implementaciones el histórico de inspecciones.

No implementamos la entidad de persona, por simplificar el diseño.

Implementamos la tabla tipo de cliente, así aprovechar futuros crecimientos, sin necesidad de implantar código en la BD.

La entidad empresas, implementamos los campos imprescindibles para su identificación.

En Modelos entendemos que todos los modelos pueden soportar diferentes tipos de potencia, no entramos en las características especiales de cada uno.

Al ejecutar el alta de un contrato, previamente tenemos que tener datos de alta los contadores y clientes, inicializamos los campos de lecturas y consumos para no tener valores extraños en los procedimientos posteriores.

La alimentación natural de la base de datos, se produce con el alta de una nueva lectura –altaLectura(contador,fechaLectura,tipoLectura,anterior,actual,consumo,rst);, en este momento, actualizamos los datos de la tabla de LECTURAS, y CONSUMOS, que servirán de base para calcular consultas y estadísticas,

Por lo que el diagrama Lógico relacional es el siguiente:

Diagrama Lógico relacional modificado

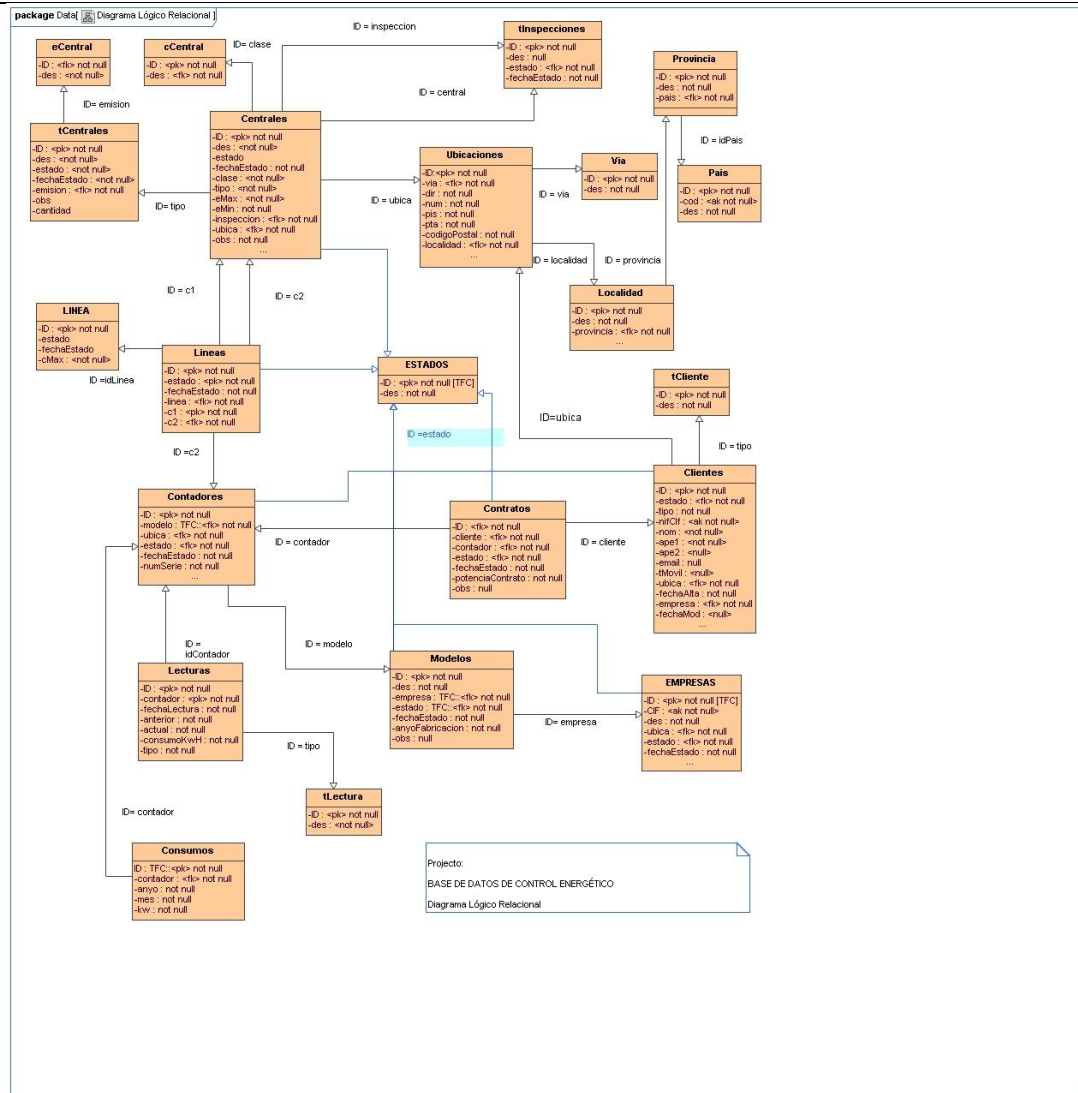
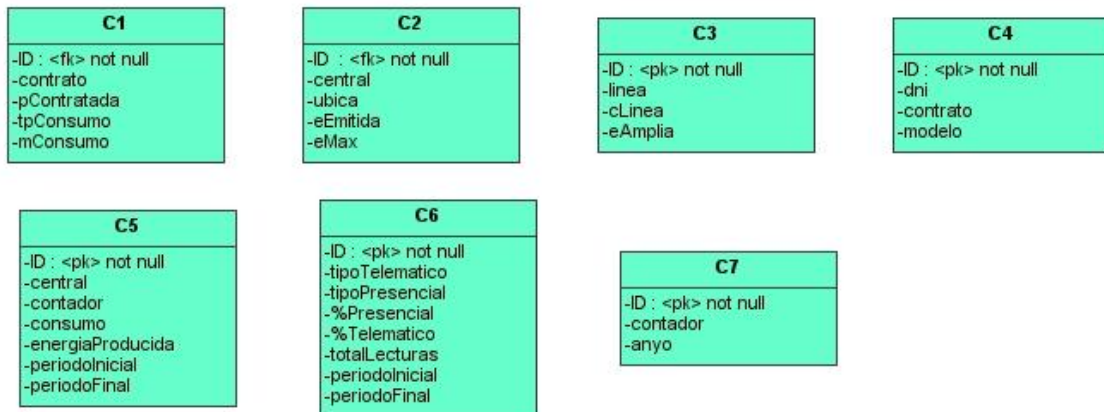


Ilustración 15 Diagrama Lógico relacional modificado

Ilustración 16 Diagrama Lógico Relacional modificado CONSULTAS



Estadísticas

E1
-ID : <pk> not null -central -consumoContadores

E2
-ID : <pk> not null -anyo : not null -consumoMedio : not null

E3
-idE3 : <pk> not null -maximoConsumo : not null

E4
-idE4 : <pk> not null -anyo : not null -consumoMas_50 : not null

E5
-idE5 : <pk> not null -energiaGeneradaMenor_30 : not null -contador : not null

E6
-idE6 : <pk> not null -anyo : not null -sumaConsumo : not null -mes : not null -top10 : not null

E7
-idCliente_CC : <pk> not null -consumoMedio : not null

Log
-idLog : <pk> not null -procesoLog : not null -fechaHoraLog : not null -entradaLog : not null -RSPLog : not null

Entidades del modelo relacional final:

estados	Contiene los datos de los posibles estados en que pueden estar central, contadores, línea, líneas,cliente, contratos,modelos, tipos de central -tCentral-, empresas.
<u>ID,des</u>	
pais	Contiene los datos de los países.
<u>ID,des,cod</u>	
provincia	Contiene los datos de las provincias.
<u>ID,des,pais</u>	
localidad	Contiene los datos de las localidades.
<u>ID,des,provincia</u>	
via	Contiene los datos de los tipos de vía pública
<u>ID,des</u>	
ubicaciones	Contiene los datos específicos de la ubicación concreta: dirección, número, puerta, código postal y el código foráneo de la localidad
<u>ID, via, dir, pis, pta, codigoPostal, localidad</u>	
tCliente	Contiene los datos de los tipos de clientes posibles.
<u>ID,des</u>	
clientes	Contiene los datos de los clientes, con las claves externas de: estado, tipo. Clave alternativa el nifCif, que dependiendo del tipo de cliente pondremos el nif o cif, conservamos también la fecha de alta y la fecha de la última modificación.
<u>ID, estado, tipo, nifCif, nom, ape1, ape2, ecorreu, tMovil, Ubica, fechaAlta, fechaMod</u>	
empresas	Contiene los datos de la empresa. Clave externa estado.
<u>ID, cif, des, estado, fechaEstado</u>	
modelos	Contiene los datos del modelo, claves externas: empresa y estado.
<u>ID, des, anyoFabrica, empresa, estado, fechaEstado, obs</u>	

contadores	Contiene los datos de los contadores, claves externas: modelo, ubica .
<u>ID, modelo, ubica, estado, fechaEstado, numSerie,</u>	
contratos	Contiene los datos de los contratos. Claves externas Cliente, contador, estado.
<u>ID, cliente, contador, estado, fechaEstado, pContrato, obs</u>	
eCentral	Contiene los datos de las emisiones / residuos que produce una central. (CO2, Placas solares, molinos etc.)
<u>ID,des</u>	
cCentral	Contiene las posibles clases de centrales, en principio : Producción y Distribución
<u>ID,des</u>	
tCentral	Contiene los tipos posibles de Central: Eólica...
<u>ID, des, estado, fechaEstad, cantidad, obs</u>	
tInspeccion	Contiene los posibles tipos de inspección : anual...
<u>ID,des estado, fechaEstado</u>	
centrales	Contiene los datos de las diferentes centrales. Claves externas: estado, clase, tipo, inspección, ubica. Conservamos la fecha de la última inspección
<u>ID, des, estado, fechaEstado, clase, tipo eMax, eMin, inspeccion, flns, ubica, obs</u>	
linea	Contiene los datos de la línea. Clave externa: estado
<u>ID, estado, fechaEstado, cMax, obs</u>	
lineas	Contiene los datos de las diferentes conexiones posibles. En C1 se pueden conectar las Centrales y en C2 los contadores y las centrales. Clave externa línea.
<u>ID, estado, fechaEstado, linea, c1,c2,obs</u>	
tLectura	Contiene los datos de los diferentes tipos de lectura: presencial, telemática
<u>ID, des</u>	
lecturas	Contiene los datos de las lecturas efectuadas, conservamos la anterior, la actual y calculamos el consumo, que servirá para actualizar los datos de la tabla de consumos. Clave externa: contador
<u>ID,contador, fechaLectura, tipo, anterior, actual, consumo</u>	

consumos	Contiene los datos de los consumos de los contadores, agrupados por el mes y el año.
<u>ID, contador, anyo,mes</u>	
Log	
<u>idLog,procesoLog,fechaLog,entradaLog,salidaLog,RSPLog</u>	
E_1	Dada una central de producción, el consumo de los contadores que dependen de la central.
<u>ID, central, consumoContadores , fecha</u>	
E_2	Dada una línea de comunicación y un año concreto, el valor medio de la energía consumida, teniendo en cuenta que este consumo depende de los contadores que se alimentan mediante esta línea. Si un contador puede utilizar dos líneas, suponga que las dos líneas computan el consumo por garantizar que las líneas están bien dimensionadas en caso de caída de alguna de ellas.
<u>ID, linea, anyo, consumoEnergiaMedio</u>	
E_3	Línea que más cargada a nivel de energía consumida.
<u>ID, linea , consumoContadores, fecha</u>	
E_4	Dado un año concreto: porcentaje de líneas que superan el 50% de energía consumida.
<u>ID , anyo , perLineaMas50</u>	
E_5	Dado un año concreto: el número de centrales de producción que generan menos del 30% de producción.
<u>ID , anyo, cenProduccionMenos30</u>	
E_6	Top-10 de contadores que históricamente han tenido más consumo.
<u>ID, contador , consumo</u>	
E_7	Consumo medio de todos los clientes.
<u>ID, cliente , consumoMedio</u>	
CONSULTAS	
CONSULTA 1 a	
C1	Dada una ciudad y una fecha como parámetros, el listado de todos los contadores donde el consumo mensual de la fecha indicada ha superado el 80% del consumo medio de todos los contadores de la ciudad en ese mismo período de tiempo.
<u>ID, contrato, pContratada, tpConsumo, mConsumo</u>	

CONSULTA 2 b	
C2	Listado de las 10 centralitas de distribución que distribuyen más energía --consultarCentralesDistribucionMasEnergia
ID , central, ubica, eEmitida , eMax	
CONSULTA 3 c	
C3	Listado de las 10 líneas de comunicación más cargadas en relación a su propia capacidad máxima --consultarLineasMasCargadasCapacidadMaxima
ID , linea , cLinea , eAmplia	
CONSULTA 4 d	
C4	El listado de los clientes que disponen de contadores con servicio en alta isponibilidad tanto de centrales de distribución, como de líneas de comunicación y de centrales de producción. --consultarClientesContadoresAltaDisponibiliad
ID , dni , contrato , modelo	
CONSULTA 5 e	
C5	Dada una central de producción y un intervalo de tiempo, queremos conocer el consumo producido por los contadores que dependen de esta central y la energía producida por la central en este mismo periodo. --consultarConsumoContadoresCentralIntervaloTiempo
ID , central , contador , consumo , eProd, pInicial , pFinal	
CONSULTA 6 f	
C6	Porcentaje de lecturas de contadores efectuadas de forma presencial y de forma telemática en un periodo de tiempo --consultarPercentLecturasPresencialTelematica
ID , telematica , presencial, pTelematica , pPresencial , total , pInicial , pInicial	
CONSULTA 7 g	
C7	Listado de contadores que tengan un determinado número de años de antigüedad. --consultarContadoresAntiguedad
ID , contador , anyo	







Manual de Funcionamiento

El primer paso consiste en instalar el Sistema de Gestión de Bases con el que se trabajará, en nuestro caso, Oracle Database Express Edition 10g Release 2 (10.2)

Posteriormente y para facilitar la gestión de los scripts, se recomienda la instalación de un editor gráfico SQL con el que el usuario se encuentre cómodo. En nuestro caso, se ha utilizado Oracle SQL Developer versión 1.0.0.15.27

“En SQL Developer la carga de cualquier objeto de base de datos (tabla, vista, secuencia, disparador, procedimiento almacenado, función...) no se observa inmediatamente y se debe refrescar. Se puede refrescar por objeto o refrescando completamente la base de datos. Otra forma es salir de SQL Developer y volver a entrar.”


La estructura creada de ficheros se organiza en :

	Borra_Crea_BD	06/06/2012 19:45	Carpeta de archivos
	dades	30/05/2012 0:33	Carpeta de archivos
	Implementacion	06/06/2012 21:06	Carpeta de archivos
	Procedures	06/06/2012 20:22	Carpeta de archivos
	Pruebas_de_salidas_txt	06/06/2012 20:03	Carpeta de archivos
	utilidades	06/06/2012 20:05	Carpeta de archivos

Desde el usuario SYSTEM, ejecutamos el código correspondiente al archivo

- ✓ 1_LOAD_USER.sql

En:




	Implementacion	06/06/2012 21:06	Carpeta de archivos
---	----------------	------------------	---------------------

Una vez ejecutado ya podremos iniciar la sesión con el nuevo usuario: TFC

La carpeta contiene la carga agrupada por procesos, que han de cargarse y ejecutarse de forma ordenada.

Se adjuntan también carpetas con procedimientos complementarios para gestionar la BD.

- Borra_Crea_BD: contiene los ficheros .sql que permiten crear la estructura de la BD de forma rápida.

	1_DROP_TOTAL.sql	06/06/2012 12:03	Archivo SQL	4 KB
	2_LOAD_TOTAL.sql	06/06/2012 18:18	Archivo SQL	283 KB
	3_Testing.sql	06/06/2012 18:23	Archivo SQL	26 KB

Se puede crear, borrar y probar de forma sencilla y rápida, para mejor eficiencia se realiza desde el editor:



Borra_Crea_BD	06/06/2012 19:45	Carpeta de archivos
dades	06/06/2012 21:20	Carpeta de archivos
Implementacion	06/06/2012 21:06	Carpeta de archivos
Procedures	06/06/2012 20:22	Carpeta de archivos
Pruebas_de_salidas_txt	06/06/2012 20:03	Carpeta de archivos
utilidades	06/06/2012 20:05	Carpeta de archivos

- Procedures: contiene todos los procedimientos agrupados por la funcionalidad correspondiente:

funciones	06/06/2012 20:04	Carpeta de archivos
proc_altas	06/06/2012 20:01	Carpeta de archivos
proc_bajas	06/06/2012 20:01	Carpeta de archivos
proc_borrado_Registros	08/05/2012 12:20	Carpeta de archivos
proc_consultas	05/06/2012 20:50	Carpeta de archivos
proc_estadisticas	06/06/2012 20:04	Carpeta de archivos
proc_modifica	06/06/2012 20:02	Carpeta de archivos

- Pruebas_de_salida_txt: contiene los ficheros .txt con el resultado de las pruebas correspondientes.

1_prueba_DropTotal.txt	06/06/2012 18:21	Documento de texto	3 KB
2_prueba_CargaTotal.txt	06/06/2012 18:22	Documento de texto	6 KB
3_prueba_Testing.txt	06/06/2012 18:23	Documento de texto	14 KB
4_prueba_InicializarLecturas.txt	05/06/2012 12:37	Documento de texto	12 KB
5_prueba_AltaEstadísticas.txt	02/06/2012 23:42	Documento de texto	4 KB

- Utilidades: contiene una serie de procesos para probar que los procesos encadenados, funcionan correctamente, poder hacer pruebas y ver los resultados inmediatamente.

inicializarConsultas.sql	04/06/2012 12:01	Archivo SQL	8 KB
inicializarEstadisticas.sql	03/06/2012 17:29	Archivo SQL	7 KB
inicializarLecturas.sql	05/06/2012 12:25	Archivo SQL	10 KB
inicializarLog.sql	28/05/2012 17:51	Archivo SQL	1 KB

Como norma general los procesos se cargan, ejecutan y muestran resultados por pantalla y los registros correspondientes en el fichero de log.

Una posible combinación de creación de tablas: el código DDL de:

Creación de tablas : IMPLEMENTACION \ 2_GESTOR_TABLAS.sql

El siguiente paso es ejecutar el código de creación de tablas anterior

✓ 2_GESTOR_TABLES.sql

Crearé las tablas necesarias para el correcto funcionamiento de la aplicación:

A continuación ejecutaremos el código del archivo

✓ 3_GESTOR_SEQ_TRG.sql

Se encuentran todos los triggers y las secuencias necesarias.

El siguiente paso es ejecutar los archivos donde se encuentran los procedimientos que se hayan almacenados en los gestores:

(Detalle teórico de los procedimientos en anexo Gestores de Procedimiento)

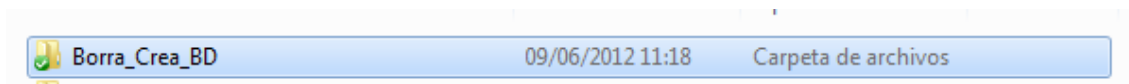
- ✓ 4_GESTOR_ALTAS.sql
- ✓ 5_GESTOR_BAJAS.sql
- ✓ 6_GESTOR_MODIFICACION.sql
- ✓ 7_GESTOR_FUNCIONES.sql
- ✓ 8_GESTOR_ESTADISTICA.sql
- ✓ 9_GESTOR_CONSULTAS.sql

Por último corresponde a la inserción de datos que se encuentra disponible en el archivo

✓ 10_LOAD_INSERTS.sql





A continuación se muestra una vista de los registros almacenados en la tabla LOG, en ella podemos ver toda la información que concierne a la actividad de la aplicación:

Se puede verificar con una sola carga completa en:



Se han creado los procedimientos, para crear y borrar la BD, se puede hacer la prueba de carga parcial, regenerar la BD y hacer carga completa.

Ejecutando secuencialmente:

 1_DROP_TOTAL.sql	06/06/2012 12:03	Archivo SQL	4 KB
 2_LOAD_TOTAL.sql	09/06/2012 10:38	Archivo SQL	342 KB
 3_Testing.sql	06/06/2012 18:23	Archivo SQL	26 KB
 4_Inserts.sql	08/06/2012 21:27	Archivo SQL	29 KB

Ejemplo de ejecución:

- a) 2...,3...
- b) 1...,2...,4...

En el caso a) obtenemos las prueba de un test a los procedimientos.
En el caso b) obtendremos una carga con gran cantidad de datos.

Pruebas de Funcionamiento

Todos los procedimientos están documentados y con el código documentado, tanto en la memoria como en cada fichero individual.

Se crea una carpeta de utilidades, para hacer pruebas en fase de desarrollo, poder variar datos de forma rápida y apreciar resultados inmediatos.





 inicializarConsultas.sql	04/06/2012 12:01	Archivo SQL	8 KB
 inicializarEstadisticas.sql	03/06/2012 17:29	Archivo SQL	7 KB
 inicializarLecturas.sql	05/06/2012 12:25	Archivo SQL	10 KB
 inicializarLog.sql	28/05/2012 17:51	Archivo SQL	1 KB

Tabla 10 Resumen de Procedures

Procedimientos			estado	Procedimientos Genéricos	Proc. Encadenado
altaCentralDistribucion	modificaCentralDistribucion	bajaCentral	estado	modificaCentral	
altaCentralProduccion	modificaCentraProduccion	bajaCentral	estado	modificaCentral	
altaClaseCentral	modificaClaseCentral		estado		
altaClientes	modificaClientes	bajaClientes	estado	modificaClientesEstado	
altaConsumos	modificaConsumos				
altaContadores	modificaContadores	bajaContadores	estado		
altaContratos	modificaContratos	bajaContratos	estado		altaLecturas
altaEmisionCentral	modificaEmisionCentral		estado		
altaEmpresa	modificaEmpresa	bajaEmpresa	estado	modificaEstadoCentral	
altaEstados	modificaEstados		estado		
altaLectura			noEstado		altaLecturas
altaLecturas	modificaLecturas		noEstado		altaConsumo
altaLinea	modificaLinea	bajaLinea	estado		
altaLineas	modificaLineas	bajaLineas	estado		
altaLocalidad	modificaLocalidad		noEstado		
altaLog			noEstado		
altaModelos	modificaModelos	bajaModelos	estado		
altaPais	modificaPais		noEstado		
altaProvincia	modificaProvincia		noEstado		
altaTipoCentral	modificaTipoCentral	bajaTipoCentral	estado		
altaTipoCliente	modificaTipoCliente				
altaTipoInspeccion	modificaTipoInspeccion	bajaTipoInspeccion	estado		
altaTipoLectura	modificaTipoLectura				
altaUbicaciones	modificaUbicaciones				
altaVia	modificaVia				

Tabla 11 Funciones


getCentralEmax
getConsumoCentral
getConsumoCentralAnyo
getConsumoCliente
etConsumoContador

getConsumoContadorFecha
getConsumoLinea
getConsumoLocalidadFecha
getConsumoMedioCliente
getConsumoMedioLocalidad
Fecha

getContadorContrato
getContadorPotContrato
getLineaCmax
getLineaConsumo
getNomCliente
getUbica

✓ GENERAL.-

En la carpeta

 Pruebas_de_salidas_txt	09/06/2012 11:17	Carpeta de archivos
--	------------------	---------------------







Están los ficheros .txt con los resultados de ejecutar las pruebas realizadas.




Creamos procedimientos para efectuar de forma rápida y exhaustiva pruebas de funcionamiento en la Base de datos:

Tabla 12 Carga prueba de BD

1. DROP TOTAL
2. LOAD TOTAL
3. Testing







Se adjuntan al final del documento y en el fichero del **Producto \ Borra_Crea_BD**






 Borra_Crea_BD	06/06/2012 19:45	Carpeta de archivos
 dades	30/05/2012 0:33	Carpeta de archivos
 Implementacion	06/06/2012 19:41	Carpeta de archivos
 Procedures	05/06/2012 11:45	Carpeta de archivos
 Pruebas_de_salidas_txt	06/06/2012 20:03	Carpeta de archivos
 utilidades	06/06/2012 20:05	Carpeta de archivos

Nombre	Fecha de modifica...	Tipo	Tamaño
 1._DROP_TOTAL.sql	06/06/2012 12:03	Archivo SQL	4 KB
 2._LOAD_TOTAL.sql	06/06/2012 18:18	Archivo SQL	283 KB
 3._Testing.sql	06/06/2012 18:23	Archivo SQL	26 KB

Se adjunta el resultado de la ejecución en los ficheros correspondientes en:

Producto \ Pruebas_de_salidas

 Borra_Crea_BD	06/06/2012 19:45	Carpeta de archivos
 dades	30/05/2012 0:33	Carpeta de archivos
 Implementacion	06/06/2012 19:41	Carpeta de archivos
 Procedures	05/06/2012 11:45	Carpeta de archivos
 Pruebas_de_salidas_txt	06/06/2012 20:03	Carpeta de archivos
 utilidades	06/06/2012 20:05	Carpeta de archivos

Nombre	Fecha de modifica...	Tipo	Tamaño
 1._prueba_DropTotal.txt	06/06/2012 18:21	Documento de tex...	3 KB
 2._prueba_CargaTotal.txt	06/06/2012 18:22	Documento de tex...	6 KB
 3._prueba_Testing.txt	06/06/2012 18:23	Documento de tex...	14 KB
 4._prueba_InicializarLecturas.txt	05/06/2012 12:37	Documento de tex...	12 KB
 5._prueba_AltaEstadísticas.txt	02/06/2012 23:42	Documento de tex...	4 KB

En el fichero de prueba de testing, tenemos todos los procedimientos, con las excepciones correspondientes, a modo de ejemplo:

1_DROP_TOTAL.sql	06/06/2012 12:03	Archivo SQL	4 KB
2_LOAD_TOTAL.sql	06/06/2012 18:18	Archivo SQL	283 KB
3_Testing.sql	06/06/2012 18:23	Archivo SQL	26 KB

Tabla 13 Pruebas de ejecución

--ESTADOS

--altaEstados
declare
rst varchar2(250);
begin
--INPUT: n_des
altaEstados('ALTA_INICIAL', rst);
altaEstados('BAJA', rst);
altaEstados('MODIFICACION', rst);
end;
/
/* exception
rst := 'ERROR: Ya existe este tipo de ESTADOS';
rst := 'ERROR: el tipo de ESTADOS no debe ser nulo';
*/
---modificaEstados
declare
rst varchar2(250);
begin
--INPUT: n_id,n_des
modificaEstados(1,'ALTA', rst);
end;
/
/* exceptions
rst := 'ERROR: No existe el id del Estado';
rst := 'ERROR: Ya existe este tipo de ESTADOS';
rst := 'ERROR: el tipo de ESTADOS no debe ser nulo';
*/

-- UBICACIONES

-- altaVia
declare
rst varchar2(250);
begin
--INPUT: n_des
altaVia('Avd.',rst);
altaVia('C/',rst);
altaVia('Pje.',rst);
altaVia('Plaza',rst);
altaVia('Pol.Ind.',rst);
altaVia('Ctra.',rst);
altaVia('Paseo',rst);
end;
/
/* exceptions
rst := 'ERROR: Ya existe este tipo de Via';
rst := 'ERROR: el tipo de via no debe ser nulo';
*/
-- modificaVia
declare
rst varchar2(250);
begin
--INPUT: n_id, n_des
modificaVia(1,'Avd.', rst);
end;

/
/* exceptions
rst := 'ERROR: No existe esta Via';
rst := 'ERROR: Ya existe este tipo de Via';
rst := 'ERROR: el tipo de via no debe ser nulo';
*/
--altaPais
declare
rst varchar2(250);
begin
--INPUT: n_id, n_des, n_cod
altaPais(1,'ESPAÑA','ES',rst);
altaPais(2,'FRANCIA','FR',rst);
altaPais(3,'MARRUECOS','MA',rst);
altaPais(4,'ALEMANIA','DE',rst);
end;
/
/* exceptions
rst := 'ERROR: Ya existe este id de PAIS';
rst := 'ERROR: Ya existe este PAIS';
rst := 'ERROR: Ya existe el codigo de Pais';
rst := 'ERROR: hay campos NO nulos, sin contenido';
*/
--modificaPais
declare
rst varchar2(250);
begin
--INPUT: n_id, n_des, n_cod
modificaPais(1,'ESPANYA','SP',rst);
end;
/
/* exceptions
rst := 'ERROR: NO existe este id de PAIS';
rst := 'ERROR: Ya existe este PAIS';
rst := 'ERROR: hay campos NO nulos, sin contenido';
*/
.../...
--altaCentralDistribucion
declare
rst varchar2(250);
begin
--
n_des,n_estado,n_fechaEstado,n_clase,n_tipo,n_eMax,n_eMin,n_inspeccion,n_flns,n_ubica,n_obs
altaCentralDistribucion('BCN_D1',1,to_char(sysdate,'DD/MM/YYYY'),1,1,17000,100,1,to_char(sysdate,'DD/MM/YYYY'),4,null,rst);
altaCentralDistribucion('BCN_D2',1,to_char(sysdate,'DD/MM/YYYY'),1,2,18000,100,1,to_char(sysdate,'DD/MM/YYYY'),5,null,rst);
altaCentralDistribucion('BCN_D3',1,to_char(sysdate,'DD/MM/YYYY'),1,3,17000,100,1,to_char(sysdate,'DD/MM/YYYY'),6,null,rst);
altaCentralDistribucion('BCN_D4',1,to_char(sysdate,'DD/MM/YYYY'),1,3,17000,100,1,to_char(sysdate,'DD/MM/YYYY'),7,null,rst);
altaCentralDistribucion('BCN_D5',1,to_char(sysdate,'DD/MM/YYYY'),1,4,17000,100,1,to_char(sysdate,'DD/MM/YYYY'),8,null,rst);
altaCentralDistribucion('BCN_D6',1,to_char(sysdate,'DD/MM/YYYY'),1,4,17000,100,1,to_char(sysdate,'DD/MM/YYYY'),9,null,rst);
altaCentralDistribucion('BCN_D7',1,to_char(sysdate,'DD/MM/YYYY'),1,4,17000,100,1,to_char(sysdate,'DD/MM/YYYY'),9,null,rst);
altaCentralDistribucion('BCN_D8',1,to_char(sysdate,'DD/MM/YYYY'),1,5,17000,100,1,to_char(sysdate,'DD/MM/YYYY'),9,null,rst);
altaCentralDistribucion('BCN_D9',1,to_char(sysdate,'DD/MM/YYYY'),1,5,17000,100,1,to_char(sysdate,'DD/MM/YYYY'),9,null,rst);
altaCentralDistribucion('BCN_D10',1,to_char(sysdate,'DD/MM/YYYY'),1,5,17000,100,1,to_char(sysdate,'DD/MM/YYYY'),9,null,rst);
end;
/
/*exceptions
rst := 'ERROR: Ya existe esta Central';
rst := 'ERROR: No existe la clase de central';
rst := 'ERROR: No existe el tipo de central';
rst := 'ERROR: No existe la ubicacion de la central';
rst := 'ERROR: No es CENTRAL DE DISTRIBUCION';

rst := 'ERROR: No existe el tipo de inspeccion';
rst := 'ERROR: Hay campos nulos sin contenido';
*/
--modificaCentralDistribucion
declare
rst varchar2(250);
begin
--
n_des,n_estado,n_fechaEstado,n_clase,n_tipo,n_eMax,n_eMin,n_inspeccion,n_flns,n_ubica,n_obs
modificaCentralDistribucion(1,'BCN_D1_modificado',1,to_char(sysdate,'DD/MM/YYYY'),1,1,20,10,1,to_char(sysdate,'DD/MM/YYYY'),4,null,rst);
end;
/
/*exceptions
rst := 'ERROR: No existe el código de la Central de Distribución';
rst := 'ERROR: Ya existe esta descripción de Central';
rst := 'ERROR: No existe la clase de central';
rst := 'ERROR: No existe la ubicacion de la central';
rst := 'ERROR: No existe el tipo de inspeccion';
rst := 'ERROR: Hay campos nulos sin contenido';
*/

✓ **LECTURAS.-**

Para probar exhaustivamente los procesos de gestión, podemos inicializar las tablas principales: Contratos, Lecturas y Consumos.

Lanza el proceso utilidades\ inicializarLecturas.

Al dar de un contrato, encademos el proceso inicializando los contadores de lecturas y consumo, y posteriormente actualizamos las lecturas correspondientes a los contadores.

Producto \ utilidades

Borra_Crea_BD	06/06/2012 19:45	Carpeta de archivos
dades	30/05/2012 0:33	Carpeta de archivos
Implementacion	06/06/2012 19:41	Carpeta de archivos
Procedures	06/06/2012 20:22	Carpeta de archivos
Pruebas_de_salidas_txt	06/06/2012 20:03	Carpeta de archivos
utilidades	06/06/2012 20:05	Carpeta de archivos

inicializarConsultas.sql	04/06/2012 12:01	Archivo SQL	8 KB
inicializarEstadisticas.sql	03/06/2012 17:29	Archivo SQL	7 KB
inicializarLecturas.sql	05/06/2012 12:25	Archivo SQL	10 KB
inicializarLog.sql	28/05/2012 17:51	Archivo SQL	1 KB

Sirve para poder probar el proceso que de forma natural alimenta el sistema, una vez cargados los datos en la BD, las lecturas que recibimos de los contadores. No actuamos sobre los datos que consideramos más estáticos, como por ejemplo los datos de empresas, clientes, ubicaciones etc.

Tabla 14 Inicializar procesos gestión

```

/*
drop table contratos cascade constraint purge;
drop table lecturas cascade constraint purge;
drop table consumos cascade constraint purge;
*/
-----
-----  CONTRATOS
drop table contratos cascade constraint purge;
CREATE TABLE contratos
(
ID          NUMBER          CONSTRAINT PK_contratos PRIMARY KEY,
cliente     CONSTRAINT FK1_contratos_cliente REFERENCES clientes(ID) NOT NULL,
contador    CONSTRAINT FK2_contratos_contador REFERENCES contadores(ID) NOT NULL,
estado      CONSTRAINT FK3_contratos_estado REFERENCES estados(ID) NOT NULL,
fechaEstado DATE           CONSTRAINT NN_contratos_fechaEstado NOT NULL,
pContrato   NUMBER (9,2)    CONSTRAINT NN_contratos_pContrato NOT NULL,
obs         VARCHAR(50 CHAR) NULL
);
drop SEQUENCE          contratos_SEQ_GRAL;
CREATE SEQUENCE        contratos_SEQ_GRAL START WITH 1 INCREMENT BY 1 NOMAXVALUE
NOCACHE;
CREATE OR REPLACE TRIGGER contratos_TR_KEY
BEFORE INSERT ON contratos
FOR EACH ROW
BEGIN SELECT contratos_SEQ_GRAL.NEXTVAL INTO :NEW.id FROM dual;

```

```

END;
/

-- LECTURAS
drop table lecturas cascade constraint purge;
CREATE TABLE lecturas
(
ID NUMBER CONSTRAINT PK_lecturas PRIMARY KEY,
contador CONSTRAINT FK1_lecturas_contador REFERENCES contadores(ID) NOT NULL,
fechaLectura DATE CONSTRAINT NN_lecturas_fechaLectura NOT NULL,
tipo CONSTRAINT FK2_lecturas_tipo REFERENCES tLectura(ID) NOT NULL,
anterior NUMBER(9,2) NULL,
actual NUMBER(9,2) NULL,
consumo NUMBER(9,2) CONSTRAINT NN_lecturas_consumo NOT NULL
);
drop sequence lecturas_SEQ_GRAL;
CREATE SEQUENCE lecturas_SEQ_GRAL START WITH 1 INCREMENT BY 1 NOMAXVALUE
NOCACHE;
CREATE OR REPLACE TRIGGER lecturas_TR_KEY
BEFORE INSERT ON lecturas
FOR EACH ROW
BEGIN
SELECT lecturas_SEQ_GRAL.NEXTVAL INTO :NEW.id FROM dual;
END;
/

-- CONSUMOS
drop table consumos cascade constraint purge;
CREATE TABLE consumos
(
ID NUMBER CONSTRAINT PK_consumos PRIMARY KEY,
contador CONSTRAINT FK_consumos_contador REFERENCES contadores(ID) NOT NULL,
anyo NUMBER CONSTRAINT NN_consumos_anyo NOT NULL,
mes NUMBER CONSTRAINT NN_consumos_mes NOT NULL,
consumo NUMBER(9,2) --CONSTRAINT NN_consumos_consumo
);
DROP SEQUENCE consumos_SEQ_GRAL;
CREATE SEQUENCE consumos_SEQ_GRAL START WITH 1 INCREMENT BY 1 NOMAXVALUE
NOCACHE;
CREATE OR REPLACE TRIGGER consumos_TR_KEY
BEFORE INSERT ON consumos
FOR EACH ROW BEGIN SELECT consumos_SEQ_GRAL.NEXTVAL INTO :NEW.id FROM dual;
END;
/

-----
declare
rst varchar2(250);
begin
-- n_cliente, n_contador, n_estado, n_fechaEstado, n_pContrato, n_obs
altaContratos(1,1,1,'01/01/2011',4000,'obs_cliente_1',rst);
altaContratos(2,2,1,'01/01/2011',4000,'obs_cliente_2',rst);
altaContratos(3,3,1,'01/01/2011',4000,'obs_cliente_3',rst);
altaContratos(4,4,1,'01/01/2011',4000,'obs_cliente_4',rst);
altaContratos(5,5,1,'01/01/2011',4000,'obs_cliente_5',rst);
altaContratos(6,6,1,'01/01/2011',4000,'obs_cliente_6',rst);
altaContratos(7,7,1,'01/01/2011',4000,'obs_cliente_7',rst);
altaContratos(8,8,1,'01/01/2011',4000,'obs_cliente_8',rst);
altaContratos(9,9,1,'01/01/2011',4000,'obs_cliente_9',rst);
altaContratos(10,10,1,'01/01/2011',4000,'obs_cliente_10',rst);
altaContratos(11,11,1,'01/02/2012',4000,'obs_cliente_11',rst);
altaContratos(12,12,1,'01/03/2012',4000,'obs_cliente_12',rst);
altaContratos(13,13,1,'01/04/2012',4000,'obs_cliente_13',rst);
altaContratos(14,14,1,'01/05/2012',4000,'obs_cliente_14',rst);
altaContratos(15,15,1,'01/06/2012',4000,'obs_cliente_15',rst);
altaContratos(16,16,1,'01/07/2012',4000,'obs_cliente_16',rst);
altaContratos(17,17,1,'01/08/2012',4000,'obs_cliente_17',rst);
altaContratos(18,18,1,'01/09/2012',4000,'obs_cliente_18',rst);
altaContratos(19,19,1,'01/10/2012',4000,'obs_cliente_19',rst);
altaContratos(20,20,1,'01/11/2012',4000,'obs_cliente_20',rst);
end;
    
```

✓ **ESTADÍSTICAS.-**

Con la finalidad de actualizar en tiempo real las tablas que contienen los requisitos de las estadísticas, se ha creado un procedimiento único –altaEstadísticas, con un parámetro de año. Sólo con esta entrada actualiza los requisitos solicitados.

NOMBRE: altaEstadísticas

PROPÓSITO: gestiona la actualización de las estadísticas

PARAMETERS: n_año ,rst

INPUT: n_año

OUTPUT: rst

RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario

EJEMPLO DE USO: altaEstadísticas(2011,rst);

En pantalla se muestran resultados de lo que se va ir actualizando en las tablas correspondientes, mediante los procedimientos específicos.

Se valida en cada procedimiento los parámetros de entrada y se implementan las excepciones.

Procedimiento complementario de desarrollo: UTILIDAD\ inicializarESTADISTICAS (inicializa todas las tablas de estadísticas, secuencias y procedimientos).

A. Procedimientos empleados:

	altaEstadísticas	
altaE1	bajaE1	modificaE1
altaE2	bajaE2	modificaE2
altaE3	bajaE3	modificaE3
altaE4	bajaE4	modificaE4
altaE5	bajaE5	modificaE5
altaE6	bajaE6	modificaE6
altaE7	bajaE7	modificaE7

B. Funciones:

Tabla 15 lista Funciones empleadas

FUNCTION getCentralEmax(n_id Compiled.
FUNCTION getConsumoCentral(n_id Compiled.
FUNCTION getConsumoCentralAnyo(n_id Compiled.
FUNCTION getConsumoCliente(n_id Compiled.
FUNCTION getConsumoContador(n_id Compiled.
FUNCTION getConsumoContadorFecha(n_id Compiled.
FUNCTION getConsumoLinea(n_id Compiled.
FUNCTION getConsumoLocalidadFecha(n_id Compiled.
FUNCTION getConsumoMedioCliente(n_id Compiled.
FUNCTION getLineaCmax(n_id Compiled.
FUNCTION getLineaConsumo(n_id Compiled.
FUNCTION getNomCliente(n_id Compiled.
FUNCTION getUbica(n_id Compiled.
FUNCTION getCentralEmax(n_id Compiled.
FUNCTION getConsumoCentral(n_id Compiled.
FUNCTION getConsumoCentralAnyo(n_id Compiled.
FUNCTION getConsumoCliente(n_id Compiled.
FUNCTION getConsumoContador(n_id Compiled.
FUNCTION getConsumoContadorFecha(n_id Compiled.
FUNCTION getConsumoLinea(n_id Compiled.

FUNCTION getConsumoLocalidadFecha(n_id Compiled.
FUNCTION getConsumoMedioCliente(n_id Compiled.
FUNCTION getConsumoMedioLocalidadFecha(n_id Compiled.
FUNCTION getLineaCmax(n_id Compiled.
FUNCTION getLineaConsumo(n_id Compiled.
FUNCTION getNomCliente(n_id Compiled.
FUNCTION getUbica(n_id Compiled.
FUNCTION getCentralEmax(n_id Compiled.
FUNCTION getConsumoCentral(n_id Compiled.
FUNCTION getConsumoCentralAnyo(n_id Compiled.
FUNCTION getConsumoCliente(n_id Compiled.
FUNCTION getConsumoContador(n_id Compiled.
FUNCTION getConsumoContadorFecha(n_id Compiled.
FUNCTION getConsumoLinea(n_id Compiled.
FUNCTION getConsumoLocalidadFecha(n_id Compiled.
FUNCTION getConsumoMedioCliente(n_id Compiled.
FUNCTION getConsumoMedioLocalidadFecha(n_id Compiled.
FUNCTION getLineaCmax(n_id Compiled.
FUNCTION getLineaConsumo(n_id Compiled.
FUNCTION getNomCliente(n_id Compiled.
FUNCTION getUbica(n_id Compiled.
FUNCTION getIdCliente(n_id Compiled.
FUNCTION getConsumoLineaAnyo(Compiled.

Tabla 16 función y procedimiento que se emplea

función	procedimiento
getCentralEmax	E5
getConsumoCentral	E1
getConsumoCentralAnyo	E5
getConsumoCliente	
getConsumoContador	
getConsumoContadorFecha	C1
getConsumoLinea	E3, E4
getConsumoLocalidadFecha	C1
getConsumoMedioCliente	E7
getConsumoMedioLocalidadFecha	C1
getLineaCmax	E4
getLineaConsumo	C3
getNomCliente	
getUbica	
getIdCliente	E7
getConsumoLineaAnyo	E4

Tabla 17 ejecutar proceso estadísticas

--- lanzar proceso
declare
rst varchar2(250);
begin
altaEstadísticas(2011,rst);
END;

Se integra en el procedimiento de altaLectura, de esta forma, al ir actualizando las lecturas, de forma natural se actualizan las tablas de estadísticas, pudiendo ser consultas de forma inmediata.

n_contador: 9 n_consumo: 3546
Alta de E6: n_contador: 8
Alta de E6: n_consumo: 2226
OK, Alta de E6 realizada con exito

n_contador: 8 n_consumo: 2226
Alta de E6: n_contador: 7
Alta de E6: n_consumo: 2106
OK, Alta de E6 realizada con exito

OK: Actualizo E_6

Alta de E_7: n_cliente: 1
Alta de E_7: n_consumoMedio : 171,666666666666666666666666666667
OK, Alta de E_7 realizada con exito
Alta de E_7: n_cliente: 2
Alta de E_7: n_consumoMedio : 65
OK, Alta de E_7 realizada con exito
Alta de E_7: n_cliente: 3
Alta de E_7: n_consumoMedio : 35,5
OK, Alta de E_7 realizada con exito
Alta de E_7: n_cliente: 4
Alta de E_7: n_consumoMedio : 45,5
OK, Alta de E_7 realizada con exito
Alta de E_7: n_cliente: 5
Alta de E_7: n_consumoMedio : 55,5
OK, Alta de E_7 realizada con exito
Alta de E_7: n_cliente: 6
Alta de E_7: n_consumoMedio : 75,5
OK, Alta de E_7 realizada con exito
Alta de E_7: n_cliente: 7
Alta de E_7: n_consumoMedio : 175,5
OK, Alta de E_7 realizada con exito
Alta de E_7: n_cliente: 8
Alta de E_7: n_consumoMedio : 185,5
OK, Alta de E_7 realizada con exito
Alta de E_7: n_cliente: 9
Alta de E_7: n_consumoMedio : 295,5
OK, Alta de E_7 realizada con exito

OK: Actualizo E_7

D. RESULTADO TABLAS ESTADÍSTICAS

- A. **TABLE E_1** --Dada una Central de Producción, el consumo de los contadores que dependen de la Central.

Results:

ID	CENTRAL	CONSUMOCONTADORES	FECHA
1	1	7254	02/06/2012
2	2	13812	02/06/2012

- B. **TABLE E_2** --consumo de energía media de una línea en un año

Results: Script Output Explain DBMS

Results:

ID	LINEA	ANYO	CONSUMOENERGIAMEDIO
1	1	2011	3,4
2	2	2011	5,67
3	3	2011	7,75
4	4	2011	7,75

- C. **TABLE E_3** -- E3_Línea que ha estado más cargada a nivel de energía consumida.

Results

Results: Script Output Explain DBMS Out

Results:

ID	LINEA	CONSUMOCONTADORES	FECHA
1	1	9984	02/06/20...

- E. **TABLE E_4** -- E4 Dado un año concreto porcentaje de líneas que superan el 50% de energía consumida.

Results: Script Output Explain

Results:

ID	ANYO	PERLINEAMAS50
1	2011	2

- F. **TABLE E_5** -- E5 Dado un año en concreto: el número de centrales de producción que generan menos del 30% de producción.

Results: Script Output Explain DBMS

Results:

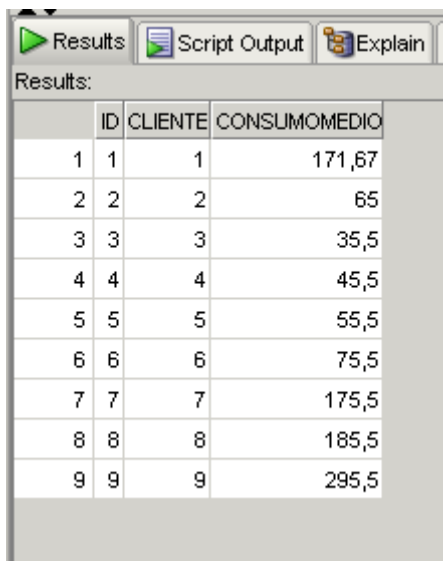
ID	ANYO	CENPRODUCCIONMENOS30
1	2011	0

G. **TABLE E_6** -- E6 Top-10 de contadores que históricamente han tenido más consumo



	ID	CONTADOR	CONSUMO
1	1	9	3546
2	2	8	2226
3	3	7	2106
4	4	1	2060
5	5	6	906
6	6	2	780
7	7	5	666
8	8	4	546
9	9	3	426
10	10	3	426

H. **TABLE E_7** -- E7 Consumo medio de todos los clientes.



	ID	CLIENTE	CONSUMOMEDIO
1	1	1	171,67
2	2	2	65
3	3	3	35,5
4	4	4	45,5
5	5	5	55,5
6	6	6	75,5
7	7	7	175,5
8	8	8	185,5
9	9	9	295,5

✓ **CONSULTAS.-**

Procedimiento complementario de desarrollo: UTILIDAD \ inicializarCONSULTAS (inicializa todas las tablas de Consultas, secuencias y procedimientos).

1) C7.- consultarContadoresAntigüedad

Listado de contadores que tengan un determinado número de años de antigüedad.

Se añade internamente en el procedimiento la fecha del día que en que se efectua la consulta.

Tabla 19 Consulta C7

NOMBRE: consultaC7
PROPÓSITO: Consulta para obtener el listado todos los contadores de un numero de años de antigüedad la antigüedad considerada por la fecha de fabricacion del modelo
PARAMETERS: n_anyos,rst
INPUT: n_anyos
OUTPUT: n_id, n_contador, n_anyos,rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: consultaC7(5, rst);

Procedimiento que lanza:

Tabla 20 alta tabla C7

NOMBRE: altaC7
PROPÓSITO: alta en la tabla de consultas C7: contadores, años de antigüedad
PARAMETERS: n_contador, n_anyo , n_fabrica,rst
INPUT: n_contador, n_anyo, n_fabrica
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: altaC7(1,7,'2010', rst);

```

--PRUEBA DE USO
declare rst varchar2(250);
begin
consultaC7(5, rst);
end;
    
```

MENSAJE POR PANTALLA:

CONSULTA DE CONTADORES CON ANTIGÜEDAD
 DE 5 años

Contador: 4 Num. años: 5 Fabricado en: 2005

Ok, Alta consultarContadoresAntigüedad

Contador: 5 Num. años: 5 Fabricado en: 2005

Ok, Alta consultarContadoresAntigüedad

Contador: 6 Num. años: 5 Fabricado en: 2005

Ok, Alta consultarContadoresAntigüedad

Contador: 7 Num. años: 5 Fabricado en: 2006

Ok, Alta consultarContadoresAntigüedad

Contador: 8 Num. años: 5 Fabricado en: 2006

Ok, Alta consultarContadoresAntigüedad

Contador: 9 Num. años: 5 Fabricado en: 2006

Ok, Alta consultarContadoresAntigüedad

Contador: 10 Num. años: 5 Fabricado en: 2006

Ok, Alta consultarContadoresAntigüedad

Contador: 16 Num. años: 5 Fabricado en: 2005

Ok, Alta consultarContadoresAntigüedad

Contador: 17 Num. años: 5 Fabricado en: 2005

Ok, Alta consultarContadoresAntigüedad

Contador: 18 Num. años: 5 Fabricado en: 2005

Ok, Alta consultarContadoresAntigüedad

Contador: 19 Num. años: 5 Fabricado en: 2005

Ok, Alta consultarContadoresAntigüedad

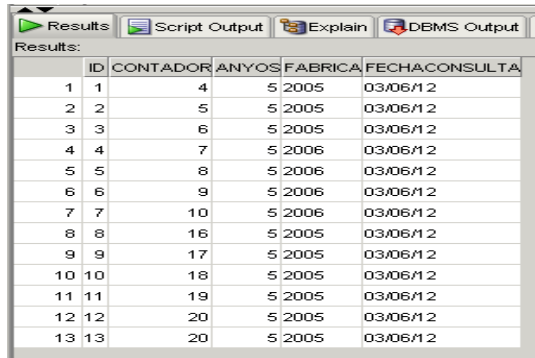
Contador: 20 Num. años: 5 Fabricado en: 2005

Ok, Alta consultarContadoresAntigüedad

Ok, Alta consultarContadoresAntigüedad

Contador: 20 Num. años: 5 Fabricado en: 2005

Tabla actualizada



ID	CONTADOR	ANYOS	FABRICA	FECHACONSULTA
1	1	4	5 2005	03/06/12
2	2	5	5 2005	03/06/12
3	3	6	5 2005	03/06/12
4	4	7	5 2006	03/06/12
5	5	8	5 2006	03/06/12
6	6	9	5 2006	03/06/12
7	7	10	5 2006	03/06/12
8	8	16	5 2005	03/06/12
9	9	17	5 2005	03/06/12
10	10	18	5 2005	03/06/12
11	11	19	5 2005	03/06/12
12	12	20	5 2005	03/06/12
13	13	20	5 2005	03/06/12

Ilustración 17 tabla Consulta C7

2) C6.- porcentaje lecturas...

Porcentaje de lecturas de contadores efectuadas de forma presencial y de forma telemática en un periodo de tiempo
 Porcentaje de lecturas de contadores efectuadas de forma presencial y de forma telemática en un periodo de tiempo

NOMBRE: consultaC6
PROPÓSITO: Consulta para obtener porcentaje de lecturas de contadores efectuadas de forma presencial y de forma telemática en un período de tiempo
PARAMETERS: n_pInicial, n_pFinal ,rst
INPUT: n_pInicial, n_pFinal
OUTPUT: tTelema,tPresent,n_pTelematica,n_pPresencial,n_total,n_pInicial_ins,n_pFinal_ins,rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: consultaC6('01/01/2012','01/06/2012', rst);

Ilustración 18 consulta C6

```
--PRUEBA DE USO
declare rst varchar2(250);
begin
consultaC6('01/01/2012','01/07/2012', rst);
end;
```

MENSAJE POR PANTALLA:

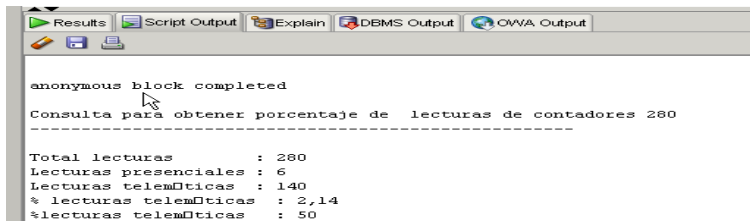


Tabla actualizada

ID	TELEMÁTICA	PRESENCIAL	PTelemática	PPresencial	TOTAL	PINICIAL	PFINAL
1	1	2	1	50	2,14	280 01/01/20...	01/07/20...

Ilustración 19 tabla consulta C6

3) C5.- energía Central en un periodo...

Dada una central de producción y un intervalo de tiempo, queremos conocer el consumo producido por los contadores que dependen de esta central y la energía producida por la central en este mismo periodo.

Tabla 21 consulta C5

NOMBRE: consultaC5
PROPÓSITO: Dada una central de producción y un intervalo de tiempo, consumo producido por los contadores que dependen de la central y la energía producida por la central en el periodo de tiempo.
PARAMETERS: n_central, n_pInicial, n_pFinal ,rst
INPUT: n_central, n_pInicial, n_pFinal
OUTPUT: rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: consultaC5(1,'01/01/2010','01/07/2012', rst);

Procedimiento que lanza:

Tabla 22 alta tabla C5

NOMBRE: altaC5
PROPÓSITO: Alta en la tabla C5
PARAMETERS: n_id, n_contador, n_consumo, n_eProd, n_ini, n_fin ,rst
INPUT: n_id, n_contador, n_consumo, n_eProd, n_ini, n_fin
OUTPUT: rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: altaC5(1,2,30,40,'01/01/2012','01/06/2012', rst);

```
--PRUEBA DE USO
declare rst varchar2(250);
begin
consultaC5(1,'01/01/2010','01/07/2012', rst);
end;
```

MENSAJE POR PANTALLA:

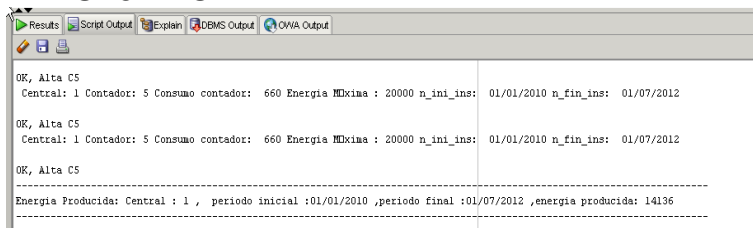


Tabla actualizada

Tabla 23 tabla C5

RESULTS	ID	CENTRAL	CONTADOR	CONSUMO	ENERGIA	INICIAL	FINAL
10	10	1	3	584	4248	01/01/2010	01/07/2012
11	11	1	3	432	4680	01/01/2010	01/07/2012
12	12	1	3	372	5040	01/01/2010	01/07/2012
13	13	1	3	260	5412	01/01/2010	01/07/2012
14	14	1	4	618	5828	01/01/2010	01/07/2012
15	15	1	4	640	6488	01/01/2010	01/07/2012
16	16	1	4	488	6848	01/01/2010	01/07/2012
17	17	1	4	528	7476	01/01/2010	01/07/2012
18	18	1	4	504	7880	01/01/2010	01/07/2012
19	19	1	4	652	852	01/01/2010	01/07/2012
20	20	1	5	492	8924	01/01/2010	01/07/2012
21	21	1	5	624	934	01/01/2010	01/07/2012
22	22	1	5	612	10260	01/01/2010	01/07/2012
23	23	1	5	672	10832	01/01/2010	01/07/2012
24	24	1	5	636	11568	01/01/2010	01/07/2012
25	25	1	5	600	12168	01/01/2010	01/07/2012
26	26	1	5	648	12816	01/01/2010	01/07/2012
27	27	1	5	660	13476	01/01/2010	01/07/2012
28	28	1	5	660	14136	01/01/2010	01/07/2012

4) C3.- línea más cargada en relación cap. máxima...

Se parametriza la entrada para poder listar el número de líneas que interese en cada momento.

Listado de las 10 Líneas de comunicación más cargadas en relación a su propia capacidad máxima...

Tabla 24 consultaC3

NOMBRE: consultaC3
PROPÓSITO: Consulta de las 10 lineas más cargadas en relación a su capacidad máxima
PARAMETERS: n_linea,n_cLinea,n_eAmplia,rst
INPUT: n_linea,n_cLinea,n_eAmplia
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: consultaC3(10, rst);

Procedimiento que lanza:

Tabla 25 altaC3

NOMBRE: altaC3
PROPÓSITO: alta en la tabla C3
PARAMETERS: n_linea,n_cLinea,n_eAmplia,rst
INPUT: n_linea,n_cLinea,n_eAmplia
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: altaC3(1,20,30, rst);

```

--PRUEBA DE USO
declare rst varchar2(250);
begin
consultaC3(10, rst);
end;
    
```

MENSAJE POR PANTALLA:

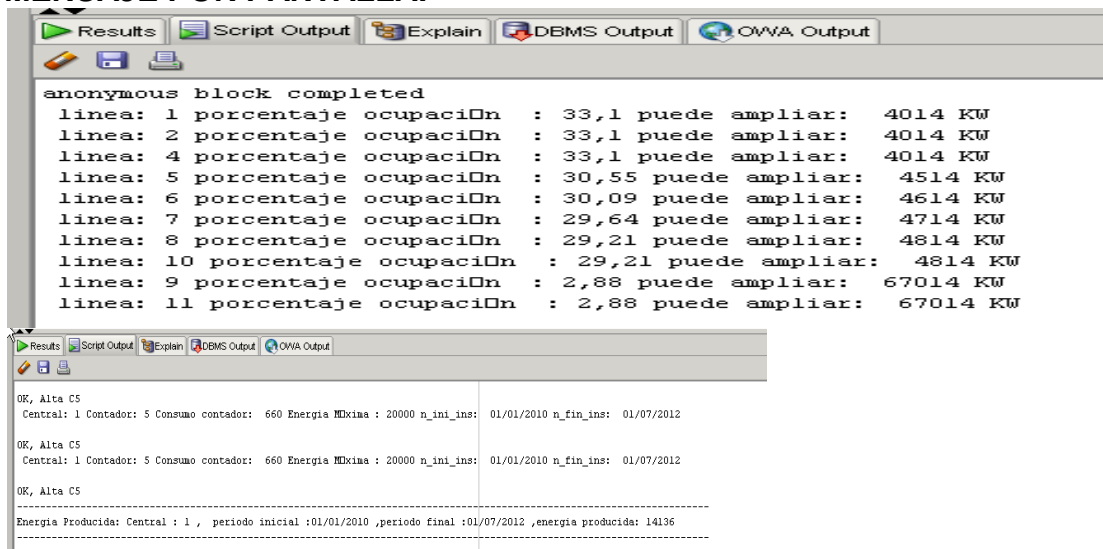
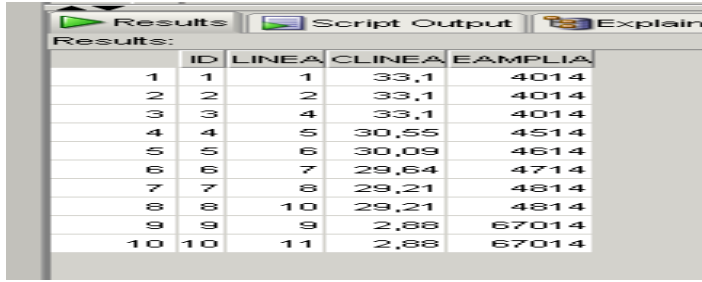


Tabla actualizada

Tabla 26 C3



	ID	LINEA	CLINEA	EAMPLIA
1	1	1	33,1	4014
2	2	2	33,1	4014
3	3	4	33,1	4014
4	4	5	30,55	4514
5	5	6	30,09	4614
6	6	7	29,64	4714
7	7	8	29,21	4814
8	8	10	29,21	4814
9	9	9	2,88	67014
10	10	11	2,88	67014

5) C1.- contadoresCiudadConsumo>80%medio....

Dada una ciudad y una fecha como parámetros, el listado de todos los contadores donde el consumo mensual de la fecha indicada ha superado el 80% del consumo medio de todos los contadores de la ciudad en ese mismo período de tiempo. En este listado se deberá devolver la información básica siguiente:

Código de contrato. La potencia máxima contratada. El tanto por ciento de consumo eléctrico consumido en relación al consumo medio. Todo ello ordenado de forma ascendente por el tanto por ciento de consumo eléctrico consumido.

NOMBRE: consultaC1
PROPÓSITO: Dada una ciudad y una fecha , los contadores donde el consumo mensual en la fecha supere el 80% del consumo medio de todos los contadores de la ciudad en ese periodo
Donada una ciutat i una data com a paràmetres, el llistat de
Codi de contracte.
La potència màxima contractada.
El tant per cent de consum elèctric consumit en relació al consum mitjà.
Tot això ordenat de forma ascendent pel tant per cent de consum elèctric consumit.
PARAMETERS: n_anyos,rst
INPUT: n_localidad_des, anyo, n_mes
OUTPUT: n_localidad_des, anyo, n_mes ,rst
RETORNA: 'OK' si la operació ha tenido exito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: consultaC1('TERRASSA',2011,7, rst);
Procedimiento que lanza: altaC1(1,12,12,12,12,'12/01/2012', rst);

Ilustración 20 Consulta C1

Tabla 27 altaC3

NOMBRE: altaC1
PROPÓSITO: Alta registro en la tabla C1
PARAMETERS: n_contrato, n_pContratada , n_tpConsumo, n_mConsumo, n_anyo, n_mes,rst
INPUT: n_contrato, n_pContratada , n_tpConsumo, n_mConsumo, n_anyo, n_mes
OUTPUT: ,rst
RETORNA: 'OK' si la operació ha tenido exito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: altaC1(1,12,12,12,12,'12/01/2012', rst);

--PRUEBA DE USO
declare rst varchar2(250);
begin
altaC1(1,12,12.30,12,2011,7, rst);
end;

MENSAJE POR PANTALLA:

```

Results | Script Output | Explain | DBMS Output | OWA Output
Ok, Cl con exito
n_contrato: 4 n_pContratada: 4000 n_tpConsumo: 37,43 n_mConsumo: 46 n_anyo: 2011
-----
Codigo contrato: 4 Potencia contratada: 4000 % consumo respecto consumo medio : 37,43 Consumo contador: 46 Anyo: 2011 Mes: 7

Ok, Cl con exito
n_contrato: 5 n_pContratada: 4000 n_tpConsumo: 45,56 n_mConsumo: 56 n_anyo: 2011
-----
Codigo contrato: 5 Potencia contratada: 4000 % consumo respecto consumo medio : 45,56 Consumo contador: 56 Anyo: 2011 Mes: 7

Ok, Cl con exito
n_contrato: 6 n_pContratada: 4000 n_tpConsumo: 61,83 n_mConsumo: 76 n_anyo: 2011
-----
Codigo contrato: 6 Potencia contratada: 4000 % consumo respecto consumo medio : 61,83 Consumo contador: 76 Anyo: 2011 Mes: 7

Ok, Cl con exito
n_contrato: 2 n_pContratada: 4000 n_tpConsumo: 56,95 n_mConsumo: 70 n_anyo: 2011
-----
Codigo contrato: 2 Potencia contratada: 4000 % consumo respecto consumo medio : 56,95 Consumo contador: 70 Anyo: 2011 Mes: 7
    
```

Tabla actualizada

	ID	CONTRATO	PCONTRATADA	TPCONSUMO	MCONSUMO	ANYO	MES
1	1	4	4000	37,43	46	2011	7
2	2	5	4000	45,56	56	2011	7
3	3	6	4000	61,83	76	2011	7
4	4	2	4000	56,95	70	2011	7
5	5	7	4000	143,19	176	2011	7
6	6	8	4000	151,33	186	2011	7
7	7	9	4000	240,83	296	2011	7
8	8	1	4000	138,31	170	2011	7
9	9	1	4000	138,31	170	2011	7

Webgrafía y bibliografía

GUIA PRÁCTICA SQL Editorial Anaya. Autor Francisco Charre Ojeda.

ORACLE 11g Editorial Ra-Ma. Autor Teaching Soft Group.

APRENDIENDO UML Editorial Prentice Hill. Autor Joseph Achmuller.

Base de datos Modelo Entidad Relación Autores Guillermo Storti, Gladys Ríos y Gabriel Campodónico.

Bases de Datos / O.E.I./ U.P.M. Transparencias de modelos Conceptuales de BD.

BASES DE DATOS Editorial UOC formación de Postgrado. Autores Rafael Camps Paré, Oscar Pérez Mora, Carme Martín Escofet, Marc Gibert Ginestà, Dolors Costal Costa, Luis Alberto Casillas Santillán. Primera edición: mayo 2005.

Components lògics d'una base de dades. Universitat Oberta de Catalunya Autores M. Elena Rodríguez González, Toni Urpí Tubella.

Components d'emmagatzematge d'una base de dades. Universitat Oberta de Catalunya. Autor Ramon Segret i Sala.

El llenguatge SQL Universitat Oberta de Catalunya. Autor Carme Martín Escofet , Carme Martín Escofet.

Programació amb SQL Universitat Oberta de Catalunya. Autores M. Elena Rodríguez González Josep Maria Marco Simó.

Fuentes de Internet

- **ORACLE PL/SQL Topics:** Contiene información útil para la realización de SELECT, UPDATE y DELETE.
<http://www.techonthenet.com/oracle/index.php>
- **Triggers i PL/SQL:** enlace a una presentación con información básica del funcionamiento de los disparadores, útil para refrescar el uso de Triggers
<http://www.infor.uva.es/~cheman/Bases/Teoria/TySQL.pdf>
- **Control de capturas y Excepciones:** Enlace con información útil que sirvió para la captura de las excepciones durante la programación de los procedimientos:
<http://www.devjoker.com/contenidos/Tutorial-PLSQL/48/Excepciones-en-PLSQL.aspx>
- **Errores en Oracle:** Página muy útil que describe los errores que van apareciendo durante la creación del código
<http://www.ora-error.com/>
- **Tutorial de Oracle:** Tutorial muy utilizado durante todo el proceso de implementación
<http://www.oracleya.com.ar/index.php?inicio=0>
- **Documentación Oracle:** Pagina con toda la información necesaria para trabajar con Oracle, desde instrucciones a códigos de ejemplo
http://download.oracle.com/docs/cd/B10501_01/index.htm

Apuntes

Benet Campderrich Falgueras, Recerca informàtica SL. Enginyeria del programari. Edició UOC. Febrer 2004.

Jaume Sistac Planas. Bases de dades I. Edició UOC.

Jaume Sistac Planas. Bases de dades II. Edició UOC..

Jaume Sistac Planas. Sistemes de gestió de bases de dades. Edició UOC

TFC de la Biblioteca de la UOC.

Módulo "Gestió i desenvolupament de projectes". © Fundació para la UOC.¹

http://materials.cv.uoc.edu/continguts/XW08_19018_00443/index.html?ajax=true

Enlaces

<http://www.oracle.com/>

<http://www.techonthenet.com/>

<http://www.psoug.org/>

<http://www.ajpdsoft.com/>

<http://www.google.com/>

Miscelánea

SIGADE

Especificaciones requeridas para el hardware, software y la capacitación.

<http://r0.unctad.org/dmfas/docs/hardsoft+trainsp.pdf>

¹ [Treball final de carrera](#)

Anexo.- Procedimientos Altas

1. GESTOR_ALTAS

Este gestor tiene asociado todos los procedimientos referentes al mantenimiento de altas de las tablas:

- 1) altaCentralDistribucion
- 2) altaCentralProduccion
- 3) altaClaseCentral
- 4) altaClientes
- 5) altaConsumos
- 6) altaConsumos_N
- 7) altaContadores
- 8) altaContratos
- 9) altaEmisionCentral
- 10) altaEmpresa
- 11) altaEstados
- 12) altaLectura
- 13) altaLecturas
- 14) altaLinea
- 15) altaLineas
- 16) altaLocalidad
- 17) altaLog
- 18) altaModelos
- 19) altaPais
- 20) altaProvincia
- 21) altaTipoCentral
- 22) altaTipoCliente
- 23) altaTipoInspeccion
- 24) altaTipoLectura
- 25) altaUbicaciones
- 26) altaVia

2. altaCentral

Tabla 28 Alta Central

Funcionalidad	El procedimiento permite crear una alta de una Central de Distribución o Producción
Explicación	<p>Tenemos creadas y cumplimentadas las tablas de Ubicaciones, vías, localidad, provincia, país, tipos de Centrales – tCentral, clase de Central –cCentral, emisión de la Central –eCentral , tipo de inspección –tInspeccion, y los estados correspondientes.</p> <p>Secuencia de acciones:</p> <ol style="list-style-type: none"> 1. comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • El campo de ID, es auto numérico. • Verificar que no hay nulos en los parámetros de entrada. • Comprobar que no existe descripción igual • Verificar que la clase de Central existe. • Verificar que el tipo de Central existe. • Verificar que el tipo de inspección existe. • Estado: clave externa indica el estado 1, ALTA, 2 BAJA; estados comunes para todas las entidades. • Verificar que el código de la ubicación existe. • El campo de estado, para comprobar que la Central está operativa y puede dar servicio, sino excepción. • Cumplimentar el máximo suministro de energía que puede generar. • El campo de inspección con el código de alta, para operar estar a 1 –operativa-, sino lanzar una excepción, el campo de fecha también relleno, conservaremos también la fecha de la última inspección por requisitos del enunciado. • No se ha de sobrepasar la capacidad máxima, en el caso de Centrales de Distribución la suma total no ha de sobrepasar a la capacidad máxima de la Central de Producción a la que estén conectados. 2. Se realiza la inserción en la tabla de Centrales, con los parámetros de entrada. 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de OK • se inserta en la tabla LOG 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>ID IN NUMBER, idConexion1 IN NUMBER, idConexion2 IN NUMBER, capacidadMaxima IN NUMBER, idEstado IN NUMBER, fechaEstado IN DATE, rst OUT VARCHAR2</p> <p>donde: idLinea corresponde al código identificador de la línea. idConexion1 corresponde al código de la central a la que se conecta. idConexion2 corresponde al código de la central o bien el código del contador. capacidadMaxima corresponde a la máxima capacidad que podemos tener conectado en esa línea. idEstado corresponde al estado de uso de esta línea, si está operativa o n, valores de la tabla de Estados. fechaEstado corresponde a la fecha del estado que tiene actualmente.</p> <p>Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p>

3. altaCentralDistribucion

Tabla 29altaCentralDistribucion

<p>Funcionalidad</p> <p>Explicación</p>	<p>El procedimiento permite crear una alta de una Central de Distribución</p> <p>Tenemos creadas y complementadas las tablas de Ubicaciones, vías, localidad, provincia, país, tipos de Centrales – tCentral, clase de Central –cCentral, emisión de la Central –eCentral , tipo de inspección –tInspeccion, y los estados correspondientes.</p> <p>Secuencia de acciones:</p> <ol style="list-style-type: none"> 1. comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • El campo de ID, es auto numérico. • Verificar que no hay nulos en los parámetros de entrada. • Comprobar que no hay descripción igual • Verificar que la clase de Central es de Distribución. • Verificar que el tipo de Central existe. • Verificar que el tipo de inspección existe. • Estado: clave externa indica el estado 1, ALTA, 2 BAJA; estados comunes para todas las entidades. • Verificar que el código de la ubicación existe. • El campo de estado, para comprobar que la Central está operativa y puede dar servicio, sino excepción. • Cumplimentar el máximo suministro de energía que puede generar. • El campo de inspección con el código de alta, para operar estar a 1 –operativa-, sino lanzar una excepción, el campo de fecha también relleno, conservaremos también la fecha de la última inspección por requisitos del enunciado. • No se ha de sobrepasar la capacidad máxima, en el caso de Centrales de Distribución la suma total no ha de sobrepasar a la capacidad máxima de la Central de Producción a la que estén conectados. 2. Se realiza el insert en la tabla de Centrales, con los parámetros de entrada. 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de OK • se inserta en la tabla LOG 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
<p>Parámetros de entrada y salida</p>	<p>n_des IN VARCHAR2, n_estado IN NUMBER, n_fechaEstado IN VARCHAR2, n_clase IN NUMBER, n_tipo IN NUMBER, n_eMax IN NUMBER, n_eMin IN NUMBER, n_inspeccion IN NUMBER, n_fIns IN VARCHAR2, n_ubica IN NUMBER, n_obs IN VARCHAR2, rst OUT VARCHAR2</p> <p>donde:</p> <p>n_des corresponde a la descripción de la Central n_estado estado de la central n_fechaEstado fecha de estado n_clase clase de Central –clave externa:Producción o Distribución n_tipo tipo de Central –externa: Eólica... n_eMax energía máxima que puede producir n_eMin energía mínima n_inspeccion tipo de inspección –externa: anual... n_fIns fecha de la última inspección n_ubica código de la ubicación –externa: ubicaciones n_obs observaciones, campo libre puede ser nulo rst OUT VARCHAR2</p> <p>Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p> <p>❖ EJEMPLO DE USO: altaCentralDistribucion('BCN_D4',1,to_char(sysdate,'DD/MM/YYYY'),2,3,20,10,1 ,to_char(sysdate,'DD/MM/YYYY'),500,null,rst);</p>

4. altaCentralProduccion

Tabla 30 altaCentralProduccion

Funcionalidad	El procedimiento permite crear una alta de una Central de Producción
Explicación	<p>Tenemos creadas y cumplimentadas las tablas de Ubicaciones, vías, localidad, provincia, país, tipos de Centrales – tCentral, clase de Central –cCentral, emisión de la Central –eCentral , tipo de inspección –tInspeccion, y los estados correspondientes.</p> <p>Secuencia de acciones:</p> <ol style="list-style-type: none"> 1. comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • El campo de ID, es auto numérico. • Verificar que no hay nulos en los parámetros de entrada. • Comprobar que no hay descripción igual • Verificar que la clase de Central es de Producción. • Verificar que el tipo de Central existe. • Verificar que el tipo de inspección existe. • Estado: clave externa indica el estado 1, ALTA, 2 BAJA; estados comunes para todas las entidades. • Verificar que el código de la ubicación existe. • El campo de estado, para comprobar que la Central está operativa y puede dar servicio, sino excepción. • Cumplimentar el máximo suministro de energía que puede generar. • El campo de inspección con el código de alta, para operar estar a 1 –operativa-, sino lanzar una excepción, el campo de fecha también relleno, conservaremos también la fecha de la última inspección por requisitos del enunciado. • No se ha de sobrepasar la capacidad máxima, en el caso de Centrales de Distribución la suma total no ha de sobrepasar a la capacidad máxima de la Central de Producción a la que estén conectados. 2. Se realiza el insert en la tabla de Centrales, con los parámetros de entrada. 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de OK • se inserta en la tabla LOG 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>n_des IN VARCHAR2, n_estado IN NUMBER, n_fechaEstado IN VARCHAR2, n_clase IN NUMBER, n_tipo IN NUMBER, n_eMax IN NUMBER, n_eMin IN NUMBER, n_inspeccion IN NUMBER, n_flns IN VARCHAR2, n_ubica IN NUMBER, n_obs IN VARCHAR2, rst OUT VARCHAR2</p> <p>donde:</p> <p>n_des corresponde a la descripción de la Central n_estado estado de la central n_fechaEstado fecha de estado n_clase clase de Central –clave externa:Producción o Distribución n_tipo tipo de Central –externa: Eólica... n_eMax energía máxima que puede producir n_eMin energía mínima n_inspeccion tipo de inspección –externa: anual... n_flns fecha de la última inspección n_ubica código de la ubicación –externa: ubicaciones n_obs observaciones, campo libre puede ser nulo rst OUT VARCHAR2</p> <p>Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p> <p>❖ EJEMPLO DE USO: <code>altaCentralProduccion('BARCELONA_2',1,to_char(sysdate,'DD/MM/YYYY'),1,3,20,10,1,'15/05/2012',1,null,rst);</code></p>

5. altaClaseCentral

Tabla 31 altaClaseCentral

Funcionalidad	El procedimiento permite crear una alta de una clase de Central.
Explicación	<p>Crea el registro correspondiente para dar de alta una clase de Central, actualmente DISTRIBUCIÓN Y PRODUCCIÓN. Podemos ampliar clases de centrales en esta tabla.</p> <p>Secuencia de acciones:</p> <ol style="list-style-type: none"> 1. comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • El campo de ID, es auto numérico. • Verificar que no hay nulos en los parámetros de entrada. • Comprobar que no existe la descripción. • Verificar que la clase de Central es de Producción. 2. Se realiza la inserción en la tabla de cCentral, con los parámetros de entrada. 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de OK • se inserta en la tabla LOG 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>n_des IN VARCHAR2, rst OUT VARCHAR2</p> <p>donde: n_des corresponde a la descripción de la Central de Central rst OUT VARCHAR2</p> <p>Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p> <p style="text-align: center;">❖ EJEMPLO DE USO: altaClaseCentral('DISTRIBUCIÓN', rst);</p>

6. altaClientes

Tabla 32 altaClientes

Funcionalidad	El procedimiento permite crear una alta de Clientes en la Base de Datos-
Explicación	<p>La secuencia de acciones que sigue el procedimiento es:</p> <ol style="list-style-type: none"> 1. comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • El campo de id, es auto numérico. • El campo de tipo, clave externa de la tabla tipo de clientes, sino lanzar una excepción. • El campo de estado, es clave externa de la tabla de Estados. • El campo de nifCIF, cumplimentado siempre, contendrá el nif si el cliente es particular o cif si es empresa. • Fecha de estado por defecto la fecha del día. • Los campos de nombre, apellidos (nom,ape1,ape2), el segundo apellido admite valores nulos • tMovil para los datos del teléfono. • Ubica, clave externa de ubicaciones, identifica la ubicación concreta. • fechaAlta, contiene la fecha de alta en el sistema. • fechaMod, contiene la fecha de la última modificación. 2. Se realiza el insert en la tabla de Clientes, con los parámetros de entrada. 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de OK • se inserta en la tabla LOG 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>n_estado IN NUMBER, n_tipo IN NUMBER, n_nifCif IN VARCHAR2, n_nom IN VARCHAR2, n_ape1 IN VARCHAR2, n_ape2 IN VARCHAR2, n_email IN VARCHAR2, n_tMovil IN VARCHAR2, n_ubica IN NUMBER, n_fechaAlta IN VARCHAR2, n_fechaMod IN VARCHAR2, rst OUT VARCHAR2</p> <p>n_estado , corresponde al código del estado del cliente n_tipo, corresponde al código del tipo del cliente n_nifCif , corresponde al nif o cif, cliente particular o empresa n_nom , nombre del cliente n_ape1 , primer apellido n_ape2 , segundo apellido n_email , dirección de correo electrónico n_tMovil , número de teléfono movil n_ubica , código de la ubicación n_fechaAlta, fecha de alta en el systema n_fechaMod , fecha de la última modificación rst OUT VARCHAR2</p> <p>Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p> <p>❖ EJEMPLO DE USO: altaClientes(1,4,'43073208X', 'Alberto', 'Ramos', null, 'aramossa@uoc.edu','34671535222',1,'17/04/2012',null, rst);</p>

7. altaConsumos

Tabla 33 altaConsumos

Funcionalidad	El procedimiento permite crear una alta de los consumos de contadores en la Base de Datos-
Explicación	<p>La secuencia de acciones que sigue el procedimiento es:</p> <ol style="list-style-type: none"> 1. comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • El campo de id, es auto numérico. • El campo de n_contador, clave externa de la tabla contadores, sino existe lanzar una excepción. • El campo n_anyo, contiene el año del consumo. • El campo n_mes, contiene el mes del consumo. • El campo n_consumo, contiene el consumo del contador en ese periodo. 2. Se realiza la inserción en la tabla de Consumos, con los parámetros de entrada. 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de OK • se inserta en la tabla LOG 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>n_contador IN NUMBER, n_anyo IN VARCHAR2, n_mes IN NUMBER, n_consumo IN NUMBER, rst OUT VARCHAR2</p> <p>n_contador , corresponde al código del contador n_anyo , corresponde año correspondiente -YYYY-. n_mes , corresponde al mes del consumo -MM-. n_consumo , corresponde al consumo en ese periodo rst OUT VARCHAR2 , contendrá los mensajes de salida.</p> <p>Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p> <p>❖ EJEMPLO DE USO: altaConsumos(1,to_char(sysdate,'YYYY'),to_char(sysdate,'MM'),12 rst)</p>

8. altaContadores

Tabla 34 altaContadores

Funcionalidad	El procedimiento permite crear una alta de Contador en la Base de Datos-
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. Comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • El campo de id, es auto numérico. • Ubica, clave externa de la tabla de ubicaciones, si no existe lanzar excepción. • Modelo, clave externa de la tabla modelos, sino existe lanzar excepción. • Estado, clave externa de la tabla de estados, si no existe lanza excepción. • Campo de numSerie, contendrá el número de serie para posterior uso. • A desarrollar: • Para llenar el campo de ubica se abre la consulta correspondiente mostrando todos los datos para completar la ubicación del Contador. • Para llenar el campo de modelo se abre la consulta correspondiente mostrando todos los campos correspondientes al modelo de Contador. 2. Se realiza la inserción en la tabla de Contadores, con los parámetros de entrada. 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de OK • se inserta en la tabla LOG 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>n_modelo IN NUMBER, n_ubica IN NUMBER, n_estado IN NUMBER, n_fechaEstado IN VARCHAR2, n_numSerie rst OUT VARCHAR2</p> <p>donde: n_modelo corresponde al modelo del contador, clave externa. n_ubica corresponde al identificador de la ubicación, clave externa. n_estado corresponde al identificador del estado del contador, clave externa. n_fechaEstado corresponde a la fecha actual del estado. Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p> <p>❖ EJEMPLO DE USO: altaContadores(1,1,1,to_char(sysdate,'DD/MM/YYYY'), 'n/s_1', rst);</p>

9. altaContratos

Tabla 35 altaContratos

Funcionalidad	El procedimiento permite crear una alta de un Contrato en la Base de Datos-
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. Comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • El campo de id, es auto numérico. • cliente, clave externa de la tabla de clientes, si no existe lanzar excepción. • contador, clave externa de la tabla de contadores, sino existe lanzar excepción. • estado, clave externa de la tabla de estados, si no existe lanza excepción. • fechaEstado, contiene la fecha del estado actual. • pContrato, contiene la potencia contratada. • obs, campo libre y no obligatorio para contener las observaciones. • mostrando todos los campos correspondientes al modelo de Contador. 2. Se realiza el insert en la tabla de Contratos, con los parámetros de entrada. 3. Lanza el procedimiento de actualizaLecturas, para inicializar los campos de la tabla lecturas. 4. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de OK • se inserta en la tabla LOG 5. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>n_cliente, n_contador, n_estado, n_fechaEstado_ins, n_pContrato, n_obs_ins IN VARCHAR2, rst OUT VARCHAR2</p> <p>donde: n_cliente corresponde al modelo del contador, clave externa. n_ubica corresponde al identificador de la ubicación, clave externa. n_estado corresponde al identificador del estado del contador, clave externa. n_fechaEstado corresponde a la fecha actual del estado. Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p> <p>❖ EJEMPLO DE USO: altaContratos(1,1,1,to_char(sysdate,'DD/MM/YYYY'),30,'obs_contrato',rst);</p>

10. altaEmissionCentral

Tabla 36 altaEmissionCentral

Funcionalidad	El procedimiento permite crear una alta de las emisiones de una Central en la Base de Datos-
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. Comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • El campo de id, es auto numérico. • n_des, descripción del tipo de emisión. 2. Se realiza la inserción en la tabla de eCentral, con los parámetros de entrada. 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de OK • se inserta en la tabla LOG 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>n_des IN VARCHAR2, rst OUT VARCHAR2 donde: n_des, corresponde al tipo de emisión de la Central : CO2, Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p> <p>❖ EJEMPLO DE USO: altaEmissionCentral('CO2', rst);</p>

11. altaEmpresa

Tabla 37 altaEmpresa

Funcionalidad	El procedimiento permite crear una alta de una Empresa en la Base de Datos-
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. Comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • El campo de id, es auto numérico. • n_cif, cif de la empresa, si ya existe lanza una excepción. • n_des, nombre descriptivo, si ya existe lanza una excepción • estado, clave externa, si no existe lanza una excepción • fechaEstado, la fecha del código del estado del campo anterior. 2. Se realiza la inserción en la tabla de empresas, con los parámetros de entrada. 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de OK • se inserta en la tabla LOG 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>n_cif IN VARCHAR2, n_des IN VARCHAR2, n_estado IN NUMBER, n_fechaEstado IN VARCHAR2, donde: n_cif, corresponde al CIF de la empresa. n_des, corresponde al nombre descriptivo de la empresa. n_estado, código del estado de la empresa, clave externa de estados. n_fechaEstado, fecha del estado. Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p> <p style="text-align: center;">❖ EJEMPLO DE USO: altaEmpresa('s121214','empresa3',5,'01/05/2012',rst);</p>

12. altaEstados

Tabla 38 altaEstados

Funcionalidad	El procedimiento permite crear una alta de un estado en Estados en la Base de Datos-
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. Comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • El campo de id, es auto numérico. • n_des, nombre descriptivo, si ya existe lanza una excepción 2. Se realiza la inserción en la tabla de estados, con los parámetros de entrada. 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de OK • se inserta en la tabla LOG 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>n_des IN VARCHAR2, rst OUT VARCHAR2</p> <p>donde: n_des, corresponde al nombre descriptivo del estado. Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p> <p style="text-align: center;">❖ EJEMPLO DE USO: altaEstado('ALTA', rst);</p>

13. altaLectura

Tabla 39 altaLectura

Funcionalidad	El procedimiento permite crear una alta de una lectura del contador actualiza la tabla de consumos.
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. Comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • El campo de id, es auto numérico. • n_contador, código de contador, clave externa, si no existe lanza una excepción. • n_fechaLectura, fecha de la lectura de contador. • n_tipo, código del tipo de lectura, clave externa. • n_lectura, valor numérico de la lectura del contador. 2. Se hacen los cálculos del consumo, en función de la lectura actual y la anterior, se actualizan los valores, anterior, actual y consumo. 3. Se lanza el procedimiento : actualizaLecturas, que ejecuta el procedimiento de altaConsumos, para actualizar la tabla de consumos. 4. Se realiza la inserción en la tabla de consumos, con los parámetros de entrada. 5. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de OK • se inserta en la tabla LOG 6. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>n_contador IN NUMBER, n_fechaLectura IN VARCHAR2, n_tipo IN NUMBER, n_lectura IN NUMBER, rst OUT VARCHAR2</p> <p>donde: n_contador, corresponde al código del contador. n_fechaLectura , corresponde a la fecha de lectura. n_tipo , código del tipo de lectura, presencial, telemática etc. n_lectura , valor numeric de la lectura del contador.</p> <p>Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p> <p style="text-align: center;">❖ EJEMPLO DE USO: altaLectura(1, to_char(sysdate,'DD/MM/YYYY'),1,1212, rst);</p>

14. altaLecturas

Tabla 40 altaLecturas

Funcionalidad	El procedimiento permite crear una alta en la tabla de lecturas y actualiza la tabla de consumos.
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. Comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • El campo de id, es auto numérico. • n_contador, código de contador, clave externa, si no existe lanza una excepción. • n_fechaLectura, fecha de la lectura de contador. • n_tipo, código del tipo de lectura, clave externa. • n_lectura, valor numérico de la lectura del contador. 2. Se hacen los cálculos del consumo, en función de la lectura actual y la anterior, se actualizan los valores, anterior, actual y consumo. 3. Se realiza la inserción en la tabla de lecturas, con los parámetros de entrada y los cálculos realizados. 4. Se lanza el procedimiento de altaConsumos, para actualizar la tabla de consumos, histórico de los consumos del contador. 5. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de OK • se inserta en la tabla LOG 6. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>n_contador IN NUMBER, n_fechaLectura IN VARCHAR2, n_tipo IN NUMBER, n_anterior IN NUMBER, n_actual IN NUMBER, n_consumo IN NUMBER, rst OUT VARCHAR2 donde: n_contador, corresponde al código del contador. n_fechaLectura , corresponde a la fecha de lectura. n_tipo , código del tipo de lectura, presencial, telemática etc. n_lectura , valor numeric de la lectura del contador. n_consumo, consumo calculado del contador</p> <p>Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p> <p style="text-align: center;">❖ EJEMPLO DE USO: altaLecturas(1,to_char(sysdate,'DD/MM/YYYY'),1,1,2,1, rst);</p>

15. altaLinea

Tabla 41 altaLinea

Funcionalidad	El procedimiento permite crear una alta de una línea en la tabla Linea en la Base de Datos-
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. Comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • El campo de id, es auto numérico. • n_estado, código del estado, clave externa. • n_fechaEstado, fecha del estado. • n_cMax, capacidad máxima de la línea • n_obs, observaciones de la línea 2. Se realiza la inserción en la tabla de Linea, con los parámetros de entrada. 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de OK • se inserta en la tabla LOG 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>n_estado IN NUMBER, n_fechaEstado IN VARCHAR2, n_cMax IN NUMBER, n_obs IN VARCHAR2, rst OUT VARCHAR2</p> <p>donde: n_estado, corresponde al código del estado. n_fechaEstado, fecha de estado anterior. n_cMax , contendrá la capacidad máxima que puede soportar la línea. n_obs, campo descriptivo de observaciones, no obligatorio. Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p> <p style="text-align: center;">❖ EJEMPLO DE USO: altaLinea(1,'10/05/2012',100,null,rst);</p>

16. altaLineas

Tabla 42 altaLineas

Funcionalidad	El procedimiento permite crear una alta de líneas –líneas de conexión entre Centrales y entre Centrales y Contadores-
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • El campo de id, es auto numérico. • n_estado, código del estado de la central, ha de estar alta para estar operativa. • n_fechaEstado, fecha del estado. • n_linea, código de la línea que utiliza. • n_c1, extremo de conexión, sólo tendremos conectadas Centrales, por tanto clave externa de Centrales. • n_c2, extremo de conexión, pueden estar conectadas Centrales de Distribución o Contadores. • n_obs, campo libre para observaciones. • Los campos de idConexion1, idConexion2, han de estar llenos. Se visualizan mediante la consulta correspondiente a Centrales o Contadores cuando corresponda. • No se ha de sobrepasar la capacidad máxima, suma de consumos de contadores conectados, o de Centrales de Distribución conectadas inferior a capacidadMaxima. 2. Se realiza el insert en la tabla de Líneas, con los parámetros de entrada. 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de OK • se inserta en la tabla LOG 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>n_estado IN NUMBER, n_fechaEstado IN VARCHAR2, n_linea IN NUMBER, n_c1 IN NUMBER, n_c2 IN NUMBER, n_obs IN VARCHAR2, rst OUT VARCHAR2</p> <p>donde: n_estado clave externa, código de estado. n_fechaEstado, fecha del estado formato DD/MM/YYYY n_linea, código de la línea, clave externa n_c1, extremo de conexión, sólo código de Centrales n_c2, extremo de conexión, código de Centrales de Distribución y contadores.</p> <p>Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p>

17. altaLocalidad

Funcionalidad	El procedimiento permite crear una alta de una localidad en la Base de Datos-
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none">1. Comprobación de los parámetros de entrada.<ul style="list-style-type: none">• El campo de id, es auto numérico.• n_des, nombre descriptivo, si ya existe lanza una excepción.• n_provincia, clave externa con el código de la Provincia.2. Se realiza la inserción en la tabla de Localidad, con los parámetros de entrada.3. Si la operación es correcta:<ul style="list-style-type: none">• mensaje de OK• se inserta en la tabla LOG4. Si se produce una excepción:<ul style="list-style-type: none">• El mensaje de salida tendrá un texto con la explicación correspondiente de excepción.• Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>n_id IN NUMBER, n_des IN VARCHAR2, n_provincia IN NUMBER, rst OUT VARCHAR2 donde: n_des, corresponde al nombre descriptivo de la localidad. n_provincia, corresponde al código de la provincia, clave externa. Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p> <p style="text-align: center;">❖ EJEMPLO DE USO: altaLocalidad(50,'Mataro',10, rst);</p>

18. altaLog

Tabla 43 altaLog

Funcionalidad	El procedimiento almacena todos los movimientos realizados en la Base de Datos.
Explicación	La secuencia de acciones que sigue en el procedimiento es: <ol style="list-style-type: none">1. comprobación de los parámetros de entrada.<ul style="list-style-type: none">• Si los campos de procesoLog, entradaLog, salidaLog, RSPLog no están rellenos se lanza una excepción.• Si los parámetros no son correctos lanza una excepción.2. Se realiza el insert en la tabla Log, con los parámetros de entrada.3. Si la operación es correcta:<ul style="list-style-type: none">• mensaje de OK4. Se inserta en la tabla.5. Si se produce una excepción:<ul style="list-style-type: none">• El mensaje de salida tendrá un texto con la explicación correspondiente de excepción.• Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>idLog IN NUMBER, procesoLog IN NUMBER, fechaHoraLog IN VARCHAR2, entradaLog IN VARCHAR2, salidaLog IN VARCHAR2, RSPLog IN VARCHAR2</p> <p>idLog corresponde al identificador del registro del Log. procesoLog corresponde al nombre del proceso que se ejecuta. fechaHoraLog corresponde a la fecha del sistema. entradaLog corresponde al valor de entrada. salidaLog corresponde al valor de salida. RSPLog corresponde al comentario.</p> <p>Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p>

19. altaModelos

Tabla 44 altaModelos

Funcionalidad	El procedimiento permite crear una alta de un modelo en la tabla de Modelos en la Base de Datos-
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. Comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • El campo de id, es auto numérico. • n_des, nombre descriptivo, si ya existe lanza una excepción. • n_anyoFabrica, año de fabricación. • n_empresa: código de la empresa, clave externa. • n_estado, código del estado, clave externa • n_fechaEstado: fecha del estado. • n_obs, campo libre, no obligatorio 2. Se realiza la inserción en la tabla de modelos, con los parámetros de entrada. 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de OK • se inserta en la tabla LOG 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepci • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>n_des IN VARCHAR2, n_anyoFabrica IN VARCHAR2, n_empresa IN NUMBER, n_estado IN NUMBER, n_fechaEstado IN VARCHAR2, n_obs IN VARCHAR2, donde: n_des, corresponde al nombre descriptivo del estado. n_anyoFabrica , corresponde al año de fabricación. n_empresa , corresponde al código de la empresa, clave externa n_estado , código del estado, clave externa. n_fechaEstado , fecha del estado n_obs , campo libre par alas observaciones.</p> <p>Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p> <p style="text-align: center;">❖ EJEMPLO DE USO: altaModelos('Modelo_1','2012',1,1,to_char(sysdate,'DD/MM/YYYY'),'ob o_1', rst);</p>

20. altaPais

Funcionalidad	El procedimiento permite crear una alta de un pais en la Base de Datos-
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. Comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • n_id, código numérico del Pais. • n_des, nombre descriptivo, si ya existe lanza una excepción • n_cod, código de caracteres del Pais. 2. Se realiza la inserción en la tabla de Pais, con los parámetros de entrada. 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de OK • se inserta en la tabla LOG 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>n_id IN NUMBER, n_des IN VARCHAR2, n_cod IN VARCHAR2, rst OUT VARCHAR2</p> <p>donde: n_id , corresponde al código numérico del pais n_des, corresponde al nombre descriptivo del Pais. n_cod , corresponde al código de caracteres del Pais Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p> <p style="text-align: center;">❖ EJEMPLO DE USO: altaPais(1,'España','ES', rst) ;</p>

21. altaProvincia

Tabla 45 altaProvincia

Funcionalidad	El procedimiento permite crear una alta de una Provincia en la Base de Datos-
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. Comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • id, código numérico de la provincia, si existe lanza excepción. • n_des, nombre descriptivo, si ya existe lanza una excepción. • n_pais, código del Pais, clave externa, si no existe lanza excepción. 2. Se realiza la inserción en la tabla de Provincia, con los parámetros de entrada. 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de OK • se inserta en la tabla LOG 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>n_id IN NUMBER, n_des IN VARCHAR2, n_pais IN NUMBER, rst OUT VARCHAR2</p> <p>donde: n_id , código numérico de la Provincia. n_des, corresponde al nombre descriptivo del estado. n_pais , código del país, clave externa Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p> <p>❖ EJEMPLO DE USO: altaProvincia(2,'BARCELONA',1,rst);</p>

22. altaTipoCentral

Funcionalidad	El procedimiento permite crear una alta de un tipo de Central en la Base de Datos-
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. Comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • El campo de id, es auto numérico. • n_des, nombre descriptivo, si ya existe lanza una excepción. • n_estado, código de estado, clave externa, sino existe lanza una excepción. • n_emisión, código del tipo de emisión de la central, clave externa, sino existe lanza excepción. • n_cantidad, cantidades de emisiones del campo anterior. • n_obs, campo libre, no obligatorio. 2. Se realiza la inserción en la tabla de tipo de central tCentral, con los parámetros de entrada. 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de OK • se inserta en la tabla LOG 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>n_des IN VARCHAR2, n_estado IN NUMBER, n_fechaEstado IN VARCHAR2, n_emision IN NUMBER, n_cantidad IN NUMBER, n_obs IN VARCHAR2, rst OUT VARCHAR2</p> <p>donde: n_des, corresponde al nombre descriptivo del estado. Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p> <p>❖ EJEMPLO DE USO: altaTipoCentral('NUCLEAR',1,'01/05/2012',1,20,NULL,rst);</p>

23. altaTipoCliente

Tabla 46 altaTipoCliente

Funcionalidad	El procedimiento permite crear una alta de un tipo de cliente en la Base de Datos-
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. Comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • El campo de id, es auto numérico. • n_des, nombre descriptivo, si ya existe lanza una excepción 2. Se realiza la inserción en la tabla de tCliente, con los parámetros de entrada. 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de OK • se inserta en la tabla LOG 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>n_des IN VARCHAR2, rst OUT VARCHAR2</p> <p>donde: n_des, corresponde al nombre descriptivo del estado. Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p> <p>❖ EJEMPLO DE USO: altaTipoCliente('PARTICULAR', rst);</p>

24. altaTipInspeccion

Funcionalidad	El procedimiento permite crear una alta de un tipo de inspección en la Base de Datos-
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none">1. Comprobación de los parámetros de entrada.<ul style="list-style-type: none">• El campo de id, es auto numérico.• n_des, nombre descriptivo, si ya existe lanza una excepción2. Se realiza la inserción en la tabla de tipo inspección tInspeccion, con los parámetros de entrada.3. Si la operación es correcta:<ul style="list-style-type: none">• mensaje de OK• se inserta en la tabla LOG4. Si se produce una excepción:<ul style="list-style-type: none">• El mensaje de salida tendrá un texto con la explicación correspondiente de excepción.• Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>n_des IN VARCHAR2, rst OUT VARCHAR2</p> <p>donde: n_des, corresponde al nombre descriptivo del nombre de la inspección. Los parámetros de salida tomarán los siguientes valores:</p> <p>OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p> <p>❖ EJEMPLO DE USO: altaTipInspeccion('ANUAL',to_char(sysdate,'DD/MM/YYYY'),rst);</p>

25. altaTipoLectura

Funcionalidad	El procedimiento permite crear una alta de un estado en Estados en la Base de Datos-
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. Comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • El campo de id, código numérico. Si existe lanza una excepción. • n_des, nombre descriptivo, si ya existe lanza una excepción 2. Se realiza la inserción en la tabla de estados, con los parámetros de entrada. 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de OK • se inserta en la tabla LOG 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>n_id IN NUMERIC, n_des IN VARCHAR2, rst OUT VARCHAR2</p> <p>donde: n_id, corresponde al código numérico. n_des, corresponde al nombre descriptivo del estado. Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p> <p>❖ EJEMPLO DE USO: altaTipoLectura(1,'PRESENCIAL', rst);</p>

26. altaUbicaciones

Tabla 47 altaUbicaciones

Funcionalidad	El procedimiento permite crear una alta en la tabla de Ubicaciones de la Base de Datos.
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • Id código auto numérico. • n_via, código identificativo de la vía, clave externa. • n_dir, descripción de la dirección. • n_num, número de la dirección. • n_pis, descriptivo del piso. • n_pta, descriptivo de la puerta. • n_codigopostal, código postal de la localidad. • n_localidad, código de la localidad, clave externa, sino existe lanza excepción. • Para los campos, vía y código postal se abrirán consultas correspondientes con los datos asociados. 2. Se realiza la inserción en la tabla Ubicaciones, con los parámetros de entrada. 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de OK 4. Se inserta en la tabla Log. 5. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>n_via IN NUMBER, n_dir IN VARCHAR2, n_num IN VARCHAR2, n_pis IN VARCHAR2, n_pta IN VARCHAR2, n_codigoPostal IN VARCHAR, n_localidad IN NUMBER, rst OUT VARCHAR2</p> <p>n_via , corresponde al identificador de Vía. n_dir , corresponde a la descripción de la dirección n_num, corresponde al número de la vivienda. n_pis, corresponde al número del piso por defecto "-". n_pta, corresponde al número de la puerta por defecto "-". n_codigopostal, correspondiente al código postal de la localidad. n_localidad corresponde al identificador de la Localidad, clave externa.</p> <p>Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p> <p>❖ EJEMPLO DE USO: altaUbicacion(1,'del pez','2','1r','2a','08330',1, rst);</p>

27. altaVia

Funcionalidad	El procedimiento permite crear una alta de un tipo de vía en la Base de Datos-
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. Comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • El campo de id, es auto numérico. • n_des, nombre descriptivo, si ya existe lanza una excepción 2. Se realiza la inserción en la tabla de Vias, con los parámetros de entrada. 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de OK • se inserta en la tabla LOG 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>n_des IN VARCHAR2, rst OUT VARCHAR2</p> <p>donde:</p> <p>n_des, corresponde al nombre descriptivo de la vía. Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p> <p>❖ EJEMPLO DE USO: altaVia('Rambla', rst);</p>

Anexo.- Procedimientos Modificación

1. modificaCentral

Tabla 48 modificaCentral

NOMBRE: modificaCentral
PROPÓSITO: gestiona la modificación de una Central
PARAMETERS: n_id,n_des,n_estado,n_fechaEstado,n_clase,n_tipo,n_eMax,n_eMin,n_inspeccion,n_flns,n_ubica,n_obs,rst
INPUT: n_id, n_des, n_estado, n_fechaEstado, n_clase, n_tipo, n_eMax, n_eMin, n_inspeccion, n_flns, n_ubica, n_obs
OUTPUT: rst
RETORNA: 'OK' si la operación ha sido exitosa, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: modificaCentral (1,'GARONA',1,'01/05/2012',1,1,20,10,1,'15/05/2012',1,'OBS_CENTRAL',rst)

2. modificaCentralDistribucion

NOMBRE: modificaCentralDistribucion
PROPÓSITO: gestiona la modificación de una Central de DISTRIBUCION
PARAMETERS: n_id,n_des,n_estado,n_fechaEstado,n_clase,n_tipo,n_eMax,n_eMin,n_inspeccion,n_flns,n_ubica,n_obs,rst
INPUT: n_id, n_des, n_estado, n_fechaEstado, n_clase, n_tipo, n_eMax, n_eMin, n_inspeccion, n_flns, n_ubica, n_obs
OUTPUT: rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: modificaCentralDistribucion(1,'BCN_D4',1,to_char(sysdate,'DD/MM/YYYY'),2,3,20,10,1,to_char(sysdate,'DD/MM/YYYY'),500,null,rst);

3. modificaCentraProduccion

NOMBRE: modificaCentralProduccion
PROPÓSITO: gestiona la modificación de una Central de Producción
PARAMETERS: n_id,n_des,n_estado,n_fechaEstado,n_clase,n_tipo,n_eMax,n_eMin,n_inspeccion,n_flns,n_ubica,n_obs,rst
INPUT: n_id, n_des, n_estado, n_fechaEstado, n_clase, n_tipo, n_eMax, n_eMin, n_inspeccion, n_flns, n_ubica, n_obs
OUTPUT: rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: modificaCentralProduccion(1,'BCN_P4',1,to_char(sysdate,'DD/MM/YYYY'),2,3,20,10,1,to_char(sysdate,'DD/MM/YYYY'),500,null,rst);

4. modificaClaseCentral

NOMBRE: modificaClaseCentral
PROPÓSITO: gestiona la modificación de una Clase de Central
PARAMETERS: n_id,n_des,rst
INPUT: n_id,n_des
OUTPUT: rst
RETORNA: 'OK' si la operación ha sido exitosa, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: modificaClaseCentral(1,'DISTRIBUCION',rst);

5. modificaClientes

NOMBRE: modificaClientes
PROPÓSITO: gestiona la modificación de un cliente en la tabla clientes
PARAMETERS: n_id,n_estado, n_tipo, n_nifCif, n_nombre, n_apellido1, n_apellido2, n_email, n_tMovil,n_ubica,n_fechaAlta,n_fechaMod, rst
INPUT: n_id, n_estado, n_estado, n_tipo, n_nifCif, n_nombre, n_apellido1, n_apellido2, n_email,

n_tMovil,n_ubica,n_fechaAlta,n_fechaMod
OUTPUT: rst
RETORNA: 'OK' si la operación ha sido exitosa, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: modificaClientes(1,4,'43073208X', 'Alberto', 'Ramos', null, aramosa@uoc.edu', '34671535222', 1, '17/04/2012', null, rst);

6. modificaClientesEstado

NOMBRE: modificaClientesEstado
PROPÓSITO: gestiona la modificación del estado de un cliente en la tabla clientes
PARAMETERS: n_id,n_estado,n_fechaMod, rst
INPUT: n_id,n_estado,n_fechaMod
OUTPUT: rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: modificaClientesEstado(3,1,to_char(sysdate,'DD/MM/YYYY'), rst);

7. modificaConsumos

NOMBRE: modificaConsumos
PROPÓSITO: gestiona el alta en la tabla de consumos
PARAMETERS: n_id,n_contador,n_anyo, n_mes,n_consumo, rst
INPUT: n_id,n_contador, n_anyo,n_mes,n_consumo
OUTPUT: rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: modificaConsumos(1,1,to_char(sysdate,'YYYY'),to_char(sysdate,'MM'),12 rst)

8. modificaContadores

NOMBRE: modificaContadores
PROPÓSITO: gestiona la modificación de un contador en la tabla contadores
PARAMETERS: n_id,n_modelo, n_ubica, n_estado, n_fechaEstado, n_numSerie,rst
INPUT: n_id,n_modelo, n_ubica, n_estado, n_fechaEstado, n_numSerie
OUTPUT: rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario

9. modificaContratos

NOMBRE: modificaContratos
PROPÓSITO: gestiona la modificación de un contrato en la tabla contratos
PARAMETERS: n_id,n_cliente, n_contador, n_estado, n_fechaEstado, n_pContrato, n_obs, rst
INPUT: n_id,n_cliente, n_contador, n_estado, n_fechaEstado, n_pContrato, n_obs
OUTPUT: rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: modificaContratos(1,1,1,1,to_char(sysdate,'DD/MM/YYYY'),30,'obs_contrato', rst);

10. modificaEmisionCentral

NOMBRE: modificaEmisionCentral
PROPÓSITO: gestiona la modificación de un tipo de emisión o especificidad de la Central
PARAMETERS: n_id,n_des, rst
INPUT: n_id,n_des
OUTPUT: rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: modificaEmisionCentral(1,'CO2', rst)

11. modificaEmpresa

NOMBRE: modificaEmpresa
PROPÓSITO: gestiona la modificación de una EMPRESA
PARAMETERS: n_id,n_cif,n_des,n_estado,n_fechaEstado, rst
INPUT: n_cif,n_des,n_estado,n_fechaEstado
OUTPUT: rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: modificaEmpresa(1's121214','empresa3',5,'01/05/2012',rst);

12. modificaEmpresa

NOMBRE: modificaEmpresa
PROPÓSITO: gestiona la modificación de una EMPRESA
PARAMETERS: n_id,n_cif,n_des,n_estado,n_fechaEstado, rst
INPUT: n_cif,n_des,n_estado,n_fechaEstado
OUTPUT: rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: modificaEmpresa(1's121214','empresa3',5,'01/05/2012',rst);

13. modificaEstadoCentral

NOMBRE: modificaEstadoCentral
PROPÓSITO: gestiona la modificación del estado de Central
PARAMETERS: n_id,n_estado, rst
INPUT: n_id,n_estado
OUTPUT: rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: modificaEstadoCentral(1,2,rst);

14. modificaEstados

NOMBRE: modificaEstados
PROPÓSITO: gestiona la modificación de un estado
PARAMETERS: n_id,n_des, rst
INPUT: n_id,n_des
OUTPUT: rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: modificaEstados('ALTA', rst);

15. modificaLecturas

NOMBRE: modificaLecturas
PROPÓSITO: gestiona el alta en la tabla de Lecturas
PARAMETERS: n_contador, n_fechaLectura,n_tipo,n_anterior,n_actual,n_consumo, rst
INPUT: n_contador, n_fechaLectura,n_tipo,n_anterior,n_actual,n_consumo
OUTPUT: rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: modificaLecturas(1,to_char(sysdate,'DD/MM/YYYY'),1,1,2,1, rst)

16. modificaLinea

NOMBRE: modificaLinea
PROPÓSITO: gestiona la modificación de una linea
PARAMETERS: n_id,n_estado, n_fechaEstado, n_cMax, n_obs, rst
INPUT: n_id,n_estado, n_fechaEstado, n_cMax, n_obs
OUTPUT: rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: modificaLinea(1,1,'10/05/2012',10000,null,rst);

17. modificaLineas

NOMBRE: modificaLineas
PROPÓSITO: gestiona la modificación de las lineas conectadas
PARAMETERS: n_id,n_estado, n_fechaEstado,n_linea, n_c1,n_c2, n_obs, rst
INPUT: n_id,n_estado, n_fechaEstado, n_linea, n_c1, n_c2, n_obs
OUTPUT: rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: modificaLineas(1,1,to_char(sysdate,'DD/MM/YYYY'),1,1,2,NULL,rst);

18. modificaLocalidad

NOMBRE: modificaLocalidad
PROPÓSITO: gestiona la modificación de un localidad en la tabla localidad
PARAMETERS: n_id, n_des,n_provincia, rst
INPUT: n_id, n_des, n_localidad
OUTPUT: rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: modificaLocalidad(50,'Mataro',10, rst);

19. modificaModelos

NOMBRE: modificaModelos
PROPÓSITO: gestiona la modificación de un modelo en la tabla modelos
PARAMETERS: n_id,n_des, n_anoFabrica, n_empresa, n_estado, n_fechaEstado, n_obs,rst;
INPUT: n_id,n_des, n_anoFabrica, n_empresa, n_estado, n_fechaEstado, n_obs
OUTPUT: rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: modificaModelos(1,'Modelo_1','2012',1,1,to_char(sysdate,'DD/MM/YYYY'),'obs_modelo_1',rst);

20. modificaPais

NOMBRE: modificaPais
PROPÓSITO: gestiona la modificación de un pais en la tabla pais
PARAMETERS: n_id, n_des,,n_cod, rst
INPUT: n_id, n_des, n_cod
OUTPUT: rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: modificaPais(1,'España','ES',rst);

21. modificaProvincia

NOMBRE: modificaProvincia
PROPÓSITO: gestiona la modificación de una provincia en la tabla provincia
PARAMETERS: n_id, n_des, n_pais rst
INPUT: n_id, n_des, n_pais
OUTPUT: rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: modificaProvincia(2,'BARCELONA',1,rst);

22. modificaTipoCentral

NOMBRE: modificaTipoCentral
PROPÓSITO: gestiona la modificación de un tipo de Central
PARAMETERS: n_id,n_des,n_estado,n_fechaEstado,n_emision,n_cantidad, n_obs, rst
INPUT: n_id,n_des,n_estado,n_fechaEstado,n_emision,n_cantidad,n_obs
OUTPUT: rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: modificaTipoCentral('NUCLEAR',1,'01/05/2012',1,20,NULL,rst);

23. modificaTipoCliente

NOMBRE: modificaTipoCliente
PROPÓSITO: gestiona la modificación de un tipo de cliente
PARAMETERS: n_id,n_des, rst
INPUT: n_id,n_des
OUTPUT: rst
RETORNA: 'OK' si la operación ha sido exitosa, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: modificaTipoCliente('PARTICULAR',rst);

24. modificaTipoInspeccion

NOMBRE: modificaTipoInspeccion
PROPÓSITO: gestiona la modificación de un tipo de inspección
PARAMETERS: n_id,n_des,n_estado,n_fechaEstado, rst
INPUT: n_id,n_des,n_estado,n_fechaEstado
OUTPUT: rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: modificaTipoInspeccion(1,'ANUAL',to_char(sysdate,'DD/MM/YYYY'),rst)

25. modificaTipoLectura

NOMBRE: modificaTipoLectura
PROPÓSITO: gestiona la modificación de un tipo de lecturas
PARAMETERS: n_id, n_des, rst
INPUT: n_id, n_des
OUTPUT: rst

RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: modificaTipoLectura (1,'PRESENCIAL', rst);

26. modificaUbicaciones

NOMBRE: modificaUbicaciones
PROPÓSITO: gestiona la modificación de una Ubicacion en la tabla ubicaciones
PARAMETERS: n_id, n_via, n_dir, n_num, n_pis, n_pta, n_codigoPostal, n_localidad, rst
INPUT: n_id, n_via, n_dir, n_num, n_pis, n_pta, n_codigoPostal, n_localidad
OUTPUT: rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: modificaUbicacion(1,1,'del pez',2,'1r',2a,'08330',1, rst);

27. modificaVia

NOMBRE: modificaVia
PROPÓSITO: gestiona la modificación de una via
PARAMETERS: n_id, n_des, rst
INPUT: n_id, n_des
OUTPUT: rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: modificaVia(1,'Rambla', rst);

Anexo.- Procedimientos Bajas

- **GESTOR_BAJAS**

- 1) bajaCentral
- 2) bajaClientes
- 3) bajaContadores
- 4) bajaContratos
- 5) bajaEmpresa
- 6) bajaLinea
- 7) bajaLineas
- 8) bajaModelos
- 9) bajaTipoCentral
- 10) bajaTipoInspeccion

1. bajaCentral

Tabla 49 Baja Central

Funcionalidad	El procedimiento permite marcar como borrado los registros de la tabla de Centrales. Se cambia el estado a borrado si no tiene líneas ni contadores asociados.
Explicación	La secuencia de acciones que se siguen en el procedimiento es la siguiente: Secuencia de acciones: <ol style="list-style-type: none"> 1. comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • Si el campo idCentral no está cumplimentado, lanza una excepción. • Comprobación de líneas asociadas, si tiene lanza una excepción. • Comprobación de contadores asociados, si tiene lanza una excepción 2. Se cambia el estado de idEstado a "Borrado". 3. realiza el Update en la tabla de Centrales, con los parámetros de entrada, excepto el parámetro de identificación -idCentral- 4. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de OK. • se inserta en la tabla LOG. 5. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	idLinea IN NUMBER, idEstado IN NUMBER donde: fechaEstado corresponde a la fecha del estado que tiene actualmente. Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.

2. bajaContador

Tabla 50 Baja Contador

Funcionalidad	<p>El procedimiento permite marcar como borrado la tabla de Contadores en la Base de Datos. Se cambia el estado a borrado, se conservan las lecturas y demás históricos.</p>
Explicación	<p>La secuencia de acciones que sigue el procedimiento es:</p> <ol style="list-style-type: none">1. Comprobación de los parámetros de entrada:<ul style="list-style-type: none">• Los campos de de idContador, sino lanza excepción.• Verificamos la existencia del contador2. Se cambia el estado de idEstado a "Borrado".3. Se realiza el Update en la tabla de Contadores, con los parámetros de entrada, excepto el identificador idContador y el número de serie.4. Si la operación es correcta:<ul style="list-style-type: none">• mensaje de Ok• se inserta en la tabla LOG5. Si se produce una excepción:<ul style="list-style-type: none">• El mensaje de salida tendrá un texto con la explicación correspondiente de excepción.• Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>idContador IN NUMBER, idEstado IN NUMBER, fechaEstado IN DATE, rst OUT VARCHAR2</p> <p>donde: idContador corresponde al identificador del contador. idEstado corresponde al identificador del estado del contador. fechaEstado corresponde a la fecha actual del estado.</p> <p>Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p>

3. bajaLineas

Tabla 51 Baja Línea

Funcionalidad	El procedimiento permite marcar como borrado de la tabla de Líneas en la Base de Datos. Se cambia el estado a borrado.
Explicación	<p>La secuencia de acciones que sigue el procedimiento es:</p> <ol style="list-style-type: none"> 1. comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • El campo de idLinea, ha de estar cumplimentado, sino lanza una excepción. • No tiene que tener Centrales o contadores asociados, si no lanza una excepción. • Verificamos la existencia de la línea que vamos a modificar. • Se cambia el estado de idEstado a "Borrado". 2. Se realiza el Update en la tabla de Líneas, con los parámetros de entrada, excepto con el parámetro de identificación. 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de Ok • se inserta en la tabla LOG 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>idLinea IN NUMBER, idEstado IN NUMBER, fechaEstado IN DATE, rst OUT VARCHAR2</p> <p>idLinea corresponde al identificador de la línea. idEstado corresponde al identificador del estado de la línea. fechaEstado IN NUMBER, rst OUT VARCHAR2</p> <p>Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p>

4. bajaClientes

Tabla 52 Baja Clientes

Funcionalidad	El procedimiento permite marcar como borrado los registros de la tabla de Clientes en la Base de Datos. Se cambia el estado a borrado.
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • Los campos de idCliente ,idUbica ,idEstado ,fechaEstado ,idTipo, han de estar cumplimentados, sino lanza una excepción • Verificamos que el cliente existe para modificar. • Se cambia el estado de idEstado a “Borrado”. 2. Se realiza el Update en la tabla de Clientes, con los parámetros de entrada, excepto el parámetro de su identificación. 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de Ok • se inserta en la tabla LOG 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>idCliente IN NUMBER, idEstado IN NUMBER, fechaEstado IN DATE, rst OUT VARCHAR2</p> <p>idCliente corresponde al identificador del cliente. idEstado corresponde al identificador del estado. fechaEstado corresponde a la fecha del estado actual rst OUT VARCHAR2</p> <p>Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p>

5. bajaUbicaciones

Tabla 53 Baja Ubicación

Funcionalidad	El procedimiento permite eliminar los registros de la tabla de Ubicaciones en la Base de Datos.
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • Si no existe idUbicacion lanza una excepción. • Todos los campos de la tabla han de estar cumplimentados, sino lanza una excepción. • Comprobamos la existencia de la Ubicación que vamos a modificar. • No ha de tener asociados clientes, centrales o contadores, sino lanzar excepción. 2. Se realiza el Delete en la tabla Ubicaciones, con los parámetros de entrada, excepto con el parámetro de su identificación. 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de Ok 4. Se inserta en la tabla en la tabla Log 5. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>idUbica IN NUMBER, idVia IN NUMBER, idLocalidad IN NUMBER2, dirección IN VARCHAR2, codigoPostal IN NUMBER, numero IN NUMBER, piso IN NUMBER, puerta IN NUMBER, tipoVivienda IN NUMBER</p> <p>idUbica corresponde al identificador de Ubicaciones. idVia corresponde al identificador de Via. idLocalidad corresponde al identificador de la Localidad. dirección corresponde a la descripción de la dirección codigoPostal corresponde al código postal. numero corresponde al número de la vivienda. piso corresponde al número del piso por defecto "-". puerta corresponde al número de la puerta por defecto "-". tipoVivienda corresponde al tipo de vivienda, desplegable con los valores posibles.</p> <p>Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p>

6. GESTOR_CONSULTAS

- C1
- C2
- C3
- C4
- C5
- C6
- C7

7. Consulta C1 -a-

Tabla 54 Consulta C1 -a-

Funcionalidad	El procedimiento permite listar los registros de la tabla de Localidad, Consumos, Contratos de la base de datos y el cálculo del consumo mensual superior al 80%.
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. comprobación que existe registro en la tabla de Localidades y registro de contrato en la tabla de Contratos <ul style="list-style-type: none"> • Si no existe la Localidad lanza una excepción. • Si no existe idContrato lanza una excepción. 2. Se realiza la consulta, tablas Localidades, Contratos , Consumos 3. Si la operación es correcta: <ul style="list-style-type: none"> • Se hacen las operaciones pertinentes. • Listado ordenado – idContrato, potenciaContrato, tanto por ciento de del consumo en relación con el consumo medio. 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>idLocalidad IN NUMBER, fecha IN DATE. rst OUT VARCHAR2</p> <p>idLocalidad corresponde al identificador de la localidad. deslocalidad corresponde a la descripción de la localidad. potenciaContrato corresponde a la potencia contratada. idContador corresponde al identificador contador. tantoPorCientoConsumo corresponde al resultado de las operaciones, ordenado ascendentemente.</p> <p>Los parámetros de salida tomarán los siguientes valores: Contratos.idContrato, Contratos.potenciaContrato, tantoPorCientoConsumo</p> <p>OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p>

8. Consulta C2 –b-

Tabla 55 Consulta C2 -b-

Funcionalidad	El procedimiento permite listar los registros de la tabla de Centrales que distribuyen más energía.
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. comprobación que existe registro en la tabla de Centrales. <ul style="list-style-type: none"> • Si no existe la idCentral lanza una excepción. • Si no existe idUbica lanza una excepción. 2. Se realiza la consulta, tablas Centrales, Ubicaciones, Lineas, Consumos. 3. Si la operación es correcta: <ul style="list-style-type: none"> • Se hacen las operaciones pertinentes. • Listado ordenado – idCentral, idUbica, consumo máximo. 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>idCentral IN NUMBER, rst OUT VARCHAR2</p> <p>idCentral corresponde al identificador de la Central. idLinea corresponde al identificador de la línea. idContador corresponde a la descripción del contador. energiaEmitida corresponde a la suma del consumo de contadores asociados. maxSuministroEnergia corresponde a la máxima capacidad de la Central</p> <p>Los parámetros de salida tomarán los siguientes valores: Centrales.idCentral, energiaEmitida, Centrales. maxSuministroEnergia</p> <p>OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p>

9. Consulta C3 –c-

Tabla 56 Consulta C3 -c-

Funcionalidad	El procedimiento permite listar los registros de la tabla de Lineas más cargadas en relación a su propia capacidad máxima.
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. comprobación que existe registro en la tabla de Lineas. <ul style="list-style-type: none"> • Si no existe la idLinea lanza una excepción. • Si no existe idCentral lanza una excepción. 2. Se realiza la consulta, tablas Lineas, Centrales, Consumos, Consumos. 3. Si la operación es correcta: <ul style="list-style-type: none"> • Se hacen las operaciones pertinentes. • Listado ordenado – idCentral, idLinea, suma consumo máximo, diferencia suma consumo máximo con capacidadMaxima. 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>idLinea IN NUMBER, idCentral IN NUMBER, idContador IN NUMBER rst OUT VARCHAR2</p> <p>idCentral corresponde al identificador de la Central. Carga corresponde a la carga de línea que soporta –diferencia en la potencia máxima y la soportada. ampliaEnergia corresponde a la energía que se puede ampliar.</p> <p>Los parámetros de salida tomarán los siguientes valores: Lineas.idlinea, carga, ampliaEnergia</p> <p>OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p>

10.Consulta C4 –d-

Tabla 57 Consulta C4 -d-

Funcionalidad	El procedimiento permite listar los registros de la tabla de Clientes con contadores conectados en alta disponibilidad.
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. comprobación que existe registro en la tabla de Contadores, Líneas y Centrales. <ul style="list-style-type: none"> • Si no existe idCliente lanza una excepción. • Si no existe la idLinea lanza una excepción. • Si no existe idCentral lanza una excepción. • Si no existe idContador lanza una excepción. 2. Se realiza la consulta. 3. Si la operación es correcta: <ul style="list-style-type: none"> • Se hacen las operaciones pertinentes. • Se comprueba que el contador tenga el módulo correspondiente instalado –altaDisponibilidad =1 – 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>idCliente IN NUMBER idLineas IN NUMBER, idCentral IN NUMBER, idContador IN NUMBER rst OUT VARCHAR2</p> <p>idCliente corresponde al identificador del cliente. idCentral corresponde al identificador de la Central. idLinea corresponde al identificador de la Linea. Carga de línea corresponde a la carga que soporta. Energía que se puede ampliar.</p> <p>Los parámetros de salida tomarán los siguientes valores:</p> <p>Cliente.NIF, Contratos.idContrato,Modelos.idModelo OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p>

11.Consulta C5 –e-

Tabla 58 Consulta C5 -e-

Funcionalidad	El procedimiento permite listar los registros de la tabla de Clientes con contadores conectados en alta disponibilidad.
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. comprobación que existe registro en la tabla de Centrales, Líneas y Consumos. <ul style="list-style-type: none"> • Si no existe idCentral lanza una excepción. • Si no existe la idLinea lanza una excepción. • Si no existe idContador lanza una excepción. 2. Se realiza la consulta. 3. Si la operación es correcta: <ul style="list-style-type: none"> • Se hacen las operaciones pertinentes. 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>idCentral IN NUMBER idLinea IN NUMBER, idContador IN NUMBER fecha_ini IN DATE fecha_fin IN DATE rst OUT VARCHAR2</p> <p>idCliente corresponde al identificador del cliente. idLinea corresponde al identificador de la Linea. idCentral corresponde al identificador de la Central. fecha_ini corresponde al periodo inicial de análisis fecha_fin corresponde al periodo final de análisis.</p> <p>Los parámetros de salida tomarán los siguientes valores:</p> <p>Fecha_ini, fecha_fin Central.idCentral sumaConsumProducido</p> <p>OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p>

12.Consulta C6 –f-

Tabla 59 Consulta C6 -f-

Funcionalidad	El procedimiento permite listar los registros de la tabla de Consumos y Lecturas efectuadas de forma presencial y telemática
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. comprobación que existe registro en la tabla de Centrales, Líneas y Consumos. <ul style="list-style-type: none"> • Si no existe idContador lanza una excepción. • Si no existe idLectura lanza una excepción. • Se realiza la consulta. 2. Si la operación es correcta: <ul style="list-style-type: none"> • Se hacen las operaciones pertinentes. 3. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>idContador IN NUMBER idLectura IN NUMBER fecha_ini IN DATE fecha_fin IN DATE rst OUT VARCHAR2</p> <p>idContador corresponde al identificador del contador. idLectura corresponde al identificador de la Lectura. fecha_ini corresponde al periodo inicial de análisis fecha_fin corresponde al periodo final de análisis. per_lecturaPresencial tanto por ciento de las lecturas presenciales. per_lecturaTelematica tanto por ciento de las lecturas telemáticas.</p> <p>Los parámetros de salida tomarán los siguientes valores:</p> <p>Fecha_ini, fecha_fin count_Lecturas.idLectura per_lecturaPresencial, per_lecturaTelematica</p> <p>OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p>

13.Consulta C7 –g-

Tabla 60 Consulta C7 -g-

Funcionalidad	El procedimiento permite listar los registros de la tabla de Contadores y Modelos Con una antigüedad determinada.
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. comprobación que existe registro en la tabla de Modelos, Contadores. <ul style="list-style-type: none"> • Si no existe idContador lanza una excepción. • Si no existe idLectura lanza una excepción. 2. Se realiza la consulta. 3. Si la operación es correcta: <ul style="list-style-type: none"> • Se hacen las operaciones pertinentes. 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>idContador IN NUMBER idModelo IN NUMBER anyoFabricacion IN NUMBER rst OUT VARCHAR2</p> <p>idContador corresponde al identificador del contador. idModelo corresponde al identificador del Modelo. anyoFabrcaion corresponde al año de fabricación del modelo antigüedad corresponde a la diferencia entre al año de fabricación y el año en curso.</p> <p>Los parámetros de salida tomarán los siguientes valores:</p> <p>Contadores.idContador, Modelos.idModelo, Modelos.anyoFabricacion, antigüedad</p> <p>OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p>

Anexo.- GESTOR ESTADÍSTICAS

Este gestor tiene asociado todos los procedimientos referentes a las estadísticas que se realizan sobre las tablas. Hemos creado procedimientos para altas, baja y modificaciones, además de funciones.

El proceso de actualización se activa simultáneamente con el proceso natural de incorporar el alta de la lectura del contador. Actualiza las tablas de estadísticas correspondientes, para tener información de las actualizaciones que se van efectuando, muestra mensajes por pantalla de las inserciones en la Base de Datos en su tabla correspondiente. . Su diseño ha sido pensado para que tenga una actualización inmediata.

Los procedimientos que forman este gestor son:

- E1 Consumo contadores de una central
- E2 Energía consumida por línea y año.
- E3 Línea más cargada
- E4 Líneas con consumo de más 50% en un año concreto.
- E5 Centrales de Producción con menos del 30% de producción de energía.
- E6 Los 10 contadores que históricamente han tenido menos consumo.
- E7 Consumo medio de todos los clientes.

1. Estadística 1 E1

Tabla 61 Estadística E1

Funcionalidad	El procedimiento permite actualizar los registros de la tabla E1_centralConsumo Central de Producción y el consumo de los contadores dependientes de esa Central.
Explicación	<p>La secuencia de acciones que se siguen en el procedimiento es la siguiente:</p> <ol style="list-style-type: none">1. comprobación de los parámetros de entrada.<ul style="list-style-type: none">• idCentral, idContador no han de estar vacíos• si los parámetros no son correctos lanzar una excepción.• Se suma la potencia contratada por cada contador.2. Se realiza la consulta3. Si la operación es correcta:<ul style="list-style-type: none">• mostramos por pantalla el resultado.• se insertan los datos en la tabla E1_centralConsumo.• Mensaje de OK• Se inserta en la tabla Log4. Si se produce una excepción:<ul style="list-style-type: none">• El mensaje de salida tendrá un texto con la explicación correspondiente de excepción.• Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>idCentral IN NUMBER, idContador IN NUMBER, rst OUT VARCHAR2</p> <p>donde: idCentral corresponde al código identificador de la Central. idContador corresponde al código del contador de central a la que se conecta. potenciaContratada corresponde a la potencia contratada.</p> <p>Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p>

2. Estadística 2 E2

Tabla 62 Estadística E2

Funcionalidad	El procedimiento permite actualizar los registros de la tabla E2_centralConsumo Central de Producción y el consumo de los contadores dependientes de esa Central.
Explicación	La secuencia de acciones que se siguen en el procedimiento es la siguiente: <ol style="list-style-type: none">1. comprobación de los parámetros de entrada.<ul style="list-style-type: none">• idCentral, idContador no han de estar vacíos• si los parámetros no son correctos lanzar una excepción.• Se suma la potencia contratada por cada contador.2. Se realiza la consulta3. Si la operación es correcta:<ul style="list-style-type: none">• mostramos por pantalla el resultado.• se insertan los datos en la tabla E1_centralConsumo.• Mensaje de OK• Se inserta en la tabla Log4. Si se produce una excepción:<ul style="list-style-type: none">• El mensaje de salida tendrá un texto con la explicación correspondiente de excepción.• Se inserta en la tabla LOG
Parámetros de entrada y salida	idCentral IN NUMBER, idContador IN NUMBER, rst OUT VARCHAR2 donde: idCentral corresponde al código identificador de la Central. idContador corresponde al código del contador de central a la que se conecta. potenciaContratada corresponde a la potencia contratada. Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.

3. Estadística 3 E3

Tabla 63 Estadística E3

Funcionalidad	El procedimiento permite actualizar los registros de la tabla E1_centralConsumo Central de Producción y el consumo de los contadores dependientes de esa Central.
Explicación	La secuencia de acciones que se siguen en el procedimiento es la siguiente: <ol style="list-style-type: none">1. comprobación de los parámetros de entrada.<ul style="list-style-type: none">• idCentral, idContador no han de estar vacíos• si los parámetros no son correctos lanzar una excepción.• Se suma la potencia contratada por cada contador.2. Se realiza la consulta3. Si la operación es correcta:<ul style="list-style-type: none">• mostramos por pantalla el resultado.• se insertan los datos en la tabla E1_centralConsumo.• Mensaje de OK• Se inserta en la tabla Log4. Si se produce una excepción:<ul style="list-style-type: none">• El mensaje de salida tendrá un texto con la explicación correspondiente de excepción.• Se inserta en la tabla LOG
Parámetros de entrada y salida	idCentral IN NUMBER, idContador IN NUMBER, rst OUT VARCHAR2 donde: idCentral corresponde al código identificador de la Central. idContador corresponde al código del contador de central a la que se conecta. potenciaContratada corresponde a la potencia contratada. Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.

4. Estadística 4 E4

Tabla 64 Estadística E4

Funcionalidad	El procedimiento permite actualizar los registros de la tabla E1_centralConsumo Central de Producción y el consumo de los contadores dependientes de esa Central.
Explicación	La secuencia de acciones que se siguen en el procedimiento es la siguiente: <ol style="list-style-type: none">1. comprobación de los parámetros de entrada.<ul style="list-style-type: none">• idCentral, idContador no han de estar vacíos• si los parámetros no son correctos lanzar una excepción.• Se suma la potencia contratada por cada contador.2. Se realiza la consulta3. Si la operación es correcta:<ul style="list-style-type: none">• mostramos por pantalla el resultado.• se insertan los datos en la tabla E1_centralConsumo.• Mensaje de OK• Se inserta en la tabla Log4. Si se produce una excepción:<ul style="list-style-type: none">• El mensaje de salida tendrá un texto con la explicación correspondiente de excepción.• Se inserta en la tabla LOG
Parámetros de entrada y salida	idCentral IN NUMBER, idContador IN NUMBER, rst OUT VARCHAR2 donde: idCentral corresponde al código identificador de la Central. idContador corresponde al código del contador de central a la que se conecta. potenciaContratada corresponde a la potencia contratada. Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.

5. Estadística 5 E5

Tabla 65 Estadística E5

Funcionalidad	El procedimiento permite actualizar los registros de la tabla E1_centralConsumo Central de Producción y el consumo de los contadores dependientes de esa Central.
Explicación	La secuencia de acciones que se siguen en el procedimiento es la siguiente: <ol style="list-style-type: none">1. comprobación de los parámetros de entrada.<ul style="list-style-type: none">• idCentral, idContador no han de estar vacíos• si los parámetros no son correctos lanzar una excepción.• Se suma la potencia contratada por cada contador.2. Se realiza la consulta3. Si la operación es correcta:<ul style="list-style-type: none">• mostramos por pantalla el resultado.• se insertan los datos en la tabla E1_centralConsumo.• Mensaje de OK• Se inserta en la tabla Log4. Si se produce una excepción:<ul style="list-style-type: none">• El mensaje de salida tendrá un texto con la explicación correspondiente de excepción.• Se inserta en la tabla LOG
Parámetros de entrada y salida	idCentral IN NUMBER, idContador IN NUMBER, rst OUT VARCHAR2 donde: idCentral corresponde al código identificador de la Central. idContador corresponde al código del contador de central a la que se conecta. potenciaContratada corresponde a la potencia contratada. Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.

6. Estadística 6 E6

Tabla 66 Estadística E6

Funcionalidad	El procedimiento permite actualizar los registros de la tabla E1_centralConsumo Central de Producción y el consumo de los contadores dependientes de esa Central.
Explicación	La secuencia de acciones que se siguen en el procedimiento es la siguiente: <ol style="list-style-type: none">1. comprobación de los parámetros de entrada.<ul style="list-style-type: none">• idCentral, idContador no han de estar vacíos• si los parámetros no son correctos lanzar una excepción.• Se suma la potencia contratada por cada contador.2. Se realiza la consulta3. Si la operación es correcta:<ul style="list-style-type: none">• mostramos por pantalla el resultado.• se insertan los datos en la tabla E1_centralConsumo.• Mensaje de OK• Se inserta en la tabla Log4. Si se produce una excepción:<ul style="list-style-type: none">• El mensaje de salida tendrá un texto con la explicación correspondiente de excepción.• Se inserta en la tabla LOG
Parámetros de entrada y salida	idCentral IN NUMBER, idContador IN NUMBER, rst OUT VARCHAR2 donde: idCentral corresponde al código identificador de la Central. idContador corresponde al código del contador de central a la que se conecta. potenciaContratada corresponde a la potencia contratada. Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.

7. Estadística 7 E7

Tabla 67 Estadística E7

Funcionalidad	<p>El procedimiento permite actualizar los registros de la tabla E1_centralConsumo Central de Producción y el consumo de los contadores dependientes de esa Central.</p> <ol style="list-style-type: none"> 1. Consumo medio de todos los clientes
Explicación	<p>La secuencia de acciones que se siguen en el procedimiento es la siguiente:</p> <ol style="list-style-type: none"> 1. comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • idCentral, idContador no han de estar vacíos • si los parámetros no son correctos lanzar una excepción. • Se suma la potencia contratada por cada contador. 2. Se realiza la consulta 3. Si la operación es correcta: <ul style="list-style-type: none"> • mostramos por pantalla el resultado. • se insertan los datos en la tabla E1_centralConsumo. • Mensaje de OK • Se inserta en la tabla Log 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>idCentral IN NUMBER, idContador IN NUMBER, rst OUT VARCHAR2</p> <p>donde: idCentral corresponde al código identificador de la Central. idContador corresponde al código del contador de central a la que se conecta. potenciaContratada corresponde a la potencia contratada.</p> <p>Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p>

Anexo.- Código proceso estadísticas

Tabla 68 Código proceso alta estadísticas

CREATE OR REPLACE
PROCEDURE altaEstadísticas(n_ano IN NUMBER, rst OUT VARCHAR2) IS
/******
NOMBRE: altaEstadísticas
PROPÓSITO: gestiona la actualización de las estadísticas
PARAMETERS: n_ano ,rst
INPUT: n_ano
OUTPUT: rst
RETORNA: 'OK' si la operación ha tenido éxito, 'ERROR: Tipo de Error' en caso contrario
EJEMPLO DE USO: altaEstadísticas(2011,rst);
*****/
-- variables para E_2 y E6
n_contador NUMBER;
n_consumo NUMBER;
n_ano_ins NUMBER; -- para E_2
e7_test INTEGER;
central_test INTEGER;
lineas_test INTEGER;
i INTEGER;
iMax INTEGER;
iMax INTEGER;
porcentaje NUMBER;
tipo_test INTEGER;
err_msg VARCHAR2(100);
dup_central EXCEPTION;
dup_lineas EXCEPTION;
dup_null EXCEPTION;
-- para actualizar E_2
CURSOR c2 IS
select distinct linea.id, consumos.ANYO, avg(consumos.CONTADOR)
from linea, lineas, consumos
where
linea.ID = lineas.LINEA
AND lineas.C2 = consumos.CONTADOR
AND consumos.ANYO = n_ano
group by linea.id, consumos.ANYO
order by linea.id;
-- cursor para extraer los datos de Estadística E6
CURSOR c1 IS
select consumos.CONTADOR,sum(consumos.consumo)
from consumos
group by consumos.contador
having sum(consumos.consumo) >0
order by sum(consumos.consumo) desc;

--proceso

BEGIN
--inicializar variables
i :=1;

```
-- actualizar estadísticas E_1
-----

-- obtenemos el numero de centrales de producción activas
select count (*)
INTO central_test
FROM centrales, ccentral
WHERE
    ccentral.ID = centrales.CLASE
AND centrales.CLASE = 2;
IF (central_test = 0)
THEN
RAISE dup_central;
END IF;

--actualizar registros
WHILE (i < central_test) LOOP
altaE1(i,getConsumoCentral(i),to_char(sysdate,'DD/MM/YYYY'),rst); -- actualizar el registro
-- DBMS_OUTPUT.PUT_LINE("");
-- DBMS_OUTPUT.PUT_LINE('central: ' || i || ' consumo:' ||getConsumoCentral(i)|| fecha: ' ||
to_char(sysdate,'DD/MM/YYYY') );
i := i + 1;
END LOOP;
-- Fin de las operaciones
rst := 'OK: Actualizo E_1';
DBMS_OUTPUT.PUT_LINE('-----');
DBMS_OUTPUT.PUT_LINE(rst);
DBMS_OUTPUT.PUT_LINE('-----');
altaLog('Proc. altaEstadisticas_E1', to_char(sysdate,'DD/MM/YYYY HH24:MI:SS'),
'alta Estadisticas'
, 'rst', rst, rst);
COMMIT;

-----

-- actualizar estadística E_2
-----

--inicializar variables
i :=0;
n_contador :=0; -- variable reutilizada para linea
n_consumo :=0; -- variable reutilizada para guardar el consumo medio de la linea
n_ano_ins :=n_ano;

OPEN c2;
    FETCH c2 INTO n_contador ,n_ano_ins ,n_consumo ;
    WHILE c2%FOUND LOOP

--DBMS_OUTPUT.PUT_LINE("");
--DBMS_OUTPUT.PUT_LINE(' linea: ' || n_contador || ' n_ano: ' || n_ano || ' consumo_medio: ' || n_consumo );
    FETCH c2 INTO n_contador ,n_ano_ins ,n_consumo ;
    altaE2(n_contador ,n_ano_ins ,n_consumo,rst); -- actualizar el registro
    end loop;
    CLOSE c2;
-- Fin de las operaciones

rst := 'OK: Actualizo E_2';
DBMS_OUTPUT.PUT_LINE('-----');
DBMS_OUTPUT.PUT_LINE(rst);
DBMS_OUTPUT.PUT_LINE('-----');

altaLog('Proc. altaEstadisticas_E2', to_char(sysdate,'DD/MM/YYYY HH24:MI:SS'),
'alta Estadisticas'
|| ' n_contador'
|| n_contador
|| ' n_ano: '
|| n_ano
|| ' n_consumo: '
|| n_consumo
, 'rst', rst, rst);
COMMIT;

-----
```

```
-- actualizar estadística E_3
-----
i := 1;
IMax := 0;
iMax := 0;
-- obtener numero de lineas conectadas
select count (distinct linea.ID)
INTO lineas_test
FROM lineas, linea
WHERE
    lineas.LINEA = linea.ID;
IF (lineas_test = 0)
THEN
RAISE dup_lineas;
END IF;
WHILE (i < lineas_test) LOOP
IF (getConsumoLinea(i) > IMax)
    THEN IMax := getConsumoLinea(i); -- buscar el consumo máximo
        iMax := i; -- id de la linea máxima
    END IF;
    i := i + 1; --avanza
END LOOP;
altaE3(i,getConsumoLinea(i),to_char(sysdate,'DD/MM/YYYY'),rst); -- actualizar el registro
-- DBMS_OUTPUT.PUT_LINE("");
-- DBMS_OUTPUT.PUT_LINE('Consumo Maximo : ||i ||, ' || IMax );
-- DBMS_OUTPUT.PUT_LINE('Linea consumo Maximo : ' ||i ||, ' || getConsumoLinea(i) );
-- Fin de las operaciones

rst := 'OK: Actualizo E_3';
DBMS_OUTPUT.PUT_LINE('-----');
DBMS_OUTPUT.PUT_LINE(rst);
DBMS_OUTPUT.PUT_LINE('-----');

altaLog('Proc. altaEstadisticas_E3', to_char(sysdate,'DD/MM/YYYY HH24:MI:SS'),
'alta Estadisticas'
, 'rst', rst, rst);
COMMIT;

-----
-- actualizar estadística E_4
-----

--lineas_test -> lineas conectadas
i := 2;
IMax := 0; --contador si linea supera el 50%
porcentaje := 0; -- porcentaje final de lineas

WHILE (i < lineas_test) LOOP
IF (getConsumoLineaAnyo(i,n_ano) > getLineaCMax(i)/2)
    THEN IMax := IMax +1; -- incrementamos el contador de lina que supera el 50%
        -- DBMS_OUTPUT.PUT_LINE('lineas_test : ' || lineas_test );
        -- DBMS_OUTPUT.PUT_LINE('IMax : ' || IMax );
        -- DBMS_OUTPUT.PUT_LINE('i : ' || i );
    END IF;
    i := i + 1; --avanza
END LOOP;

porcentaje := IMax*100/lineas_test;

altaE4(n_ano, iMax,rst); -- actualizar el registro
-- DBMS_OUTPUT.PUT_LINE("");
-- DBMS_OUTPUT.PUT_LINE('lineas_test : ' || lineas_test );
-- DBMS_OUTPUT.PUT_LINE('n_ano : ' ||i ||, ' || n_ano );
-- DBMS_OUTPUT.PUT_LINE('Porcentaje de lineas que superan el 50% : ' ||porcentaje ||'%' );
-- Fin de las operaciones

rst := 'OK: Actualizo E_4';
DBMS_OUTPUT.PUT_LINE('-----');
DBMS_OUTPUT.PUT_LINE(rst);
```

DBMS_OUTPUT.PUT_LINE('-----');
altaLog('Proc. altaEstadisticas_E4', to_char(sysdate,'DD/MM/YYYY HH24:MI:SS'),
'alta Estadisticas'
, 'rst', rst, rst);
COMMIT;

-- actualizar estadística E_5

i := 1;
iMax := 0;
WHILE (i < central_test) LOOP
IF (getConsumoCentralAnyo(i ,n_ano) < getCentralEmax(i) * 0.3) -- produccion < del 30% de eMax de la central
THEN
iMax := iMax+1; -- acumulo el numero de centrales
END IF;
i := i + 1;
END LOOP;
-- DBMS_OUTPUT.PUT_LINE("");
-- DBMS_OUTPUT.PUT_LINE('n_ano : ' n_ano);
-- DBMS_OUTPUT.PUT_LINE('iMax : ' iMax);
altaE5(n_ano,iMax,rst); -- actualizar el registro

-- Fin de las operaciones
rst := 'OK: Actualizo E_5';
DBMS_OUTPUT.PUT_LINE('-----');
DBMS_OUTPUT.PUT_LINE(rst);
DBMS_OUTPUT.PUT_LINE('-----');
altaLog('Proc. altaEstadisticas_E5', to_char(sysdate,'DD/MM/YYYY HH24:MI:SS'),
'alta Estadisticas'
, 'rst', rst, rst);
COMMIT;

-- actualizar estadística E_6

--inicializar contador
i :=0;
--verificamos que los campos no nulos no sean nulos
OPEN c1;
FETCH c1 INTO n_contador,n_consumo ;
WHILE (c1%FOUND and (c1%ROWCOUNT <= 10)) LOOP
DBMS_OUTPUT.PUT_LINE('-----');
DBMS_OUTPUT.PUT_LINE(' n_contador: ' n_contador ' n_consumo: ' n_consumo);
FETCH c1 INTO n_contador,n_consumo ;
altaE6(n_contador, n_consumo,rst); -- actualizar el registro
end loop;
CLOSE c1;
-- Fin de las operaciones

rst := 'OK: Actualizo E_6';
DBMS_OUTPUT.PUT_LINE('-----');
DBMS_OUTPUT.PUT_LINE(rst);
DBMS_OUTPUT.PUT_LINE('-----');
altaLog('Proc. altaEstadisticas_E6', to_char(sysdate,'DD/MM/YYYY HH24:MI:SS'),
'alta Estadisticas'
' n_contador'
n_contador
' n_consumo: '
n_consumo
, 'rst', rst, rst);
COMMIT;

-- actualizar estadística E_7

```

i :=1;
--obtener el numero de clientes
select count(distinct clientes.id)
into e7_test
from contratos, clientes, contadores, consumos
where clientes.ID = contratos.CLIENTE
and contadores.ID = contratos.CONTADOR
and contadores.ID = consumos.CONTADOR;
WHILE (i < e7_test) LOOP
altaE7(getIdCliente(i), getConsumoMedioCliente(i),rst); -- actualizar el registro
i := i + 1;
-- DBMS_OUTPUT.PUT_LINE("");
-- DBMS_OUTPUT.PUT_LINE('Cliente y consumo medio : '||getIdCliente(i) ||', '|| getConsumoMedioCliente(i) );
END LOOP;

-- Fin de las operaciones
rst := 'OK: Actualizo E_7';
DBMS_OUTPUT.PUT_LINE('-----');
DBMS_OUTPUT.PUT_LINE(rst);
DBMS_OUTPUT.PUT_LINE('-----');

altaLog('Proc. altaEstadisticas_E7', to_char(sysdate,'DD/MM/YYYY HH24:MI:SS'),
'alta Estadisticas'
, 'rst', rst, rst);
COMMIT;
-----

EXCEPTION
WHEN dup_central
THEN
rst := 'ERROR: No hay centrales';
DBMS_OUTPUT.PUT_LINE("");
DBMS_OUTPUT.PUT_LINE(rst);
altaLog('Proc. altaEstadistica E_1', to_char(sysdate,'DD/MM/YYYY HH24:MI:SS'),
'alta Estadisticas'
, 'rst', rst, rst);
ROLLBACK;
WHEN dup_lineas
THEN
rst := 'ERROR: No hay lineas conectadas';
DBMS_OUTPUT.PUT_LINE("");
DBMS_OUTPUT.PUT_LINE(rst);
altaLog('Proc. altaEstadistica E_3', to_char(sysdate,'DD/MM/YYYY HH24:MI:SS'),
'alta Estadisticas'
, 'rst', rst, rst);
ROLLBACK;

WHEN dup_null
THEN
rst := 'ERROR: hay campos NO nulos, sin contenido';
DBMS_OUTPUT.PUT_LINE("");
DBMS_OUTPUT.PUT_LINE(rst);
altaLog('Proc. altaEstadisticas', to_char(sysdate,'DD/MM/YYYY HH24:MI:SS'),
'alta Estadisticas'
, 'rst', rst, rst);
ROLLBACK;
WHEN OTHERS
THEN
err_msg := SUBSTR (SQLERRM, 1, 100);
rst := 'ERROR:'
|| err_msg;
altaLog('Proc. altaEstadisticas', to_char(sysdate,'DD/MM/YYYY HH24:MI:SS'),
'alta Estadisticas'
, 'rst', rst, rst);
ROLLBACK;
END altaEstadisticas;

--- lanzar proceso
declare
rst varchar2(250);

```

begin
altaEstadisticas(2011,rst);
END;

PROCEDIMIENTOS DE ESTADÍSTICAS:

altaE1 (1,300,'01/01/2011',rst)

Dada una central de producción, el consumo de los contadores que dependen de la central.

ejemplo de uso	altaE1(1,300,rst);
campos no nulos	- los parámetros de entrada no deben ser nulos : n_central, n_consumoContadores
clave(s) externa(s)	- n_central referenciado a : centrales.id
clave(s) alternativa(s)	-
log	Si
módulo estadístico	

altaE2 (1,'2012',90,50,rst);

Dada una línea de comunicación y un año en concreto, el valor medio de la energía consumida si el contador hace servir dos líneas, computa en las dos

ejemplo de uso	altaE2(1,'2012',90,50,rst);
campos no nulos	- los parámetros de entrada no deben ser nulos: n_linea,n_ano, n_consumoContadores,n_consumoEnergiaMedio, rst
clave(s) externa(s)	- n_linea referenciado a : linea.id
clave(s) alternativa(s)	-
log	Si
módulo estadístico	

altaE3 (1,200,rst);

Línea que ha estat més carregada a nivell d'energia consumida.

ejemplo de uso	altaE3(1,200,rst);
campos no nulos	- los parámetros de entrada no deben ser nulos: n_linea, n_consumoContadores -
clave(s) externa(s)	- n_linea referenciado a : linea.id
clave(s) alternativa(s)	-
log	Si
módulo estadístico	

altaE4 ('2012',90,rst);

Dado un año en concreto: porcentaje de líneas que superan 50% de energía consumida.

ejemplo de uso	altaE4(1, '12', '2009', 1, rst);
campos no nulos	- los parámetros de entrada no deben ser nulos: n_ano, n_perLineaMas50-
clave(s) externa(s)	-
clave(s) alternativa(s)	-
log	Si
módulo estadístico	

altaE5 ('2012',90,rst)

Dado un año en concreto: el numero de centrales de produccion que generan menos del 30% de producción.

ejemplo de uso	altaE5('2012',90,rst)
campos no nulos	- los parámetros de entrada no deben ser nulos: n_ano, cenProduccionMenos30-
clave(s) externa(s)	-
clave(s) alternativa(s)	-
log	Si
módulo estadístico	

altaE6 altaE6('2012',90,rst);

--Dado un año en concreto: el numero de centrales de produccion que generan menos del 30% de producción.

ejemplo de uso altaE6('2012',90,rst);
campos no nulos - los parámetros de entrada no deben ser nulos: n_contador,n_consumo-
clave(s) externa(s) - n_contador referenciado a : contadores.id
clave(s) alternativa(s) -
log Si
módulo estadístico

altaE7 (2,90,rst);
--Consumo medio de todos los clientes.
ejemplo de uso altaE7(2,90,rst);
campos no nulos - los parámetros de entrada no deben ser nulos : n_contador,n_consumo
clave(s) externa(s) -n_contador referenciado a contadores.id
clave(s) alternativa(s) -
log Si
módulo estadístico

Anexo.- FUNCIONES

Guia funcion & procedimiento.

Tabla 69 Función - procedimiento

función	procecimiento
getCentralEmax	E5
getConsumoCentral	E1
getConsumoCentralAnyo	E5
getConsumoCliente	
getConsumoContador	
getConsumoContadorFecha	C1
getConsumoLinea	E3, E4
getConsumoLocalidadFecha	C1
getConsumoMedioCliente	E7
getConsumoMedioLocalidadFecha	C1
getLineaCmax	E4
getLineaConsumo	C3
getNomCliente	
getUbica	
getIdCliente	E7
getConsumoLineaAnyo	E4

Tabla 70 Funciones

NOMBRE: getCentralEmax
PROPÓSITO: para conocer la energia máxima que puede producir la central
PARAMETERS: n_id
INPUT: n_id
OUTPUT: ret
RETORNA: ret (variable con el consumo)
NOMBRE: getConsumoCentral
PROPÓSITO: para conocer el consumo la central -contadores conectados-, activa
PARAMETERS: n_id
INPUT: n_id
OUTPUT: ret
RETORNA: ret (variable con el consumo)
NOMBRE: getConsumoCentralAnyo
PROPÓSITO: para conocer el consumo en un año determinado de una la central de producción
PARAMETERS: n_id, n_ayno
INPUT: n_id, n_ayo
OUTPUT: ret
RETORNA: ret (variable con el consumo)
EJEMPLO DE USO: getConsumoCentralAnyo (2,2011)
CREATE OR REPLACE FUNCTION getConsumoCliente(n_id NUMBER)
RETURN NUMBER
NOMBRE: getConsumoMedioCliente
PROPÓSITO: para conocer el consumo de un cliente en alta, contadores activos y conectados

PARAMETERS: n_id
INPUT: n_id
OUTPUT: ret
RETORNA: ret (variable con el consumo medio del cliente)
EJEMPLO DE USO: getConsumoMedioCliente (2)
NOMBRE: getConsumoContador
PROPÓSITO: para conocer el consumo de un contador, activo y conectado
PARAMETERS: n_id
INPUT: n_id
OUTPUT: -
RETORNA: ret (variable con el consumo total del contador)
EJEMPLO DE USO: getConsumoContador (2)
NOMBRE: getConsumoContadorFecha
PROPÓSITO: para conocer el consumo del contador en un año y mes determinado
PARAMETERS: n_id, n_ano, n_mes
INPUT: n_id, n_ano, n_mes
OUTPUT: ret
RETORNA: ret (variable con el consumo)
EJEMPLO DE USO: getConsumoContadorFecha (2,2011,1)
NOMBRE: getConsumoCentral
PROPÓSITO: para conocer el consumo la linea -contadores conectados-, activa
PARAMETERS: n_id
INPUT: n_id
OUTPUT: ret
RETORNA: ret (variable con el consumo)
EJEMPLO DE USO: getConsumoLinea (2)
NOMBRE: getConsumoLocalidadFecha
PROPÓSITO: para conocer el consumo en un año determinado de una la central de producción
PARAMETERS: n_id, n_ano, n_mes
INPUT: n_id, n_ano, n_mes
OUTPUT: ret
RETORNA: ret (variable con el consumo)
EJEMPLO DE USO: getConsumoLocalidadFecha (2,2011,1)
NOMBRE: getConsumoMedioCliente
PROPÓSITO: para conocer el consumo medio de un cliente en alta, contadores activos y conectados
PARAMETERS: n_id
INPUT: n_id
OUTPUT: ret
RETORNA: ret (variable con el consumo medio del cliente)
EJEMPLO DE USO: getConsumoMedioCliente (2)
NOMBRE: getConsumoMedioLocalidadFecha
PROPÓSITO: para conocer el consumo en un año determinado de una la central de producción
PARAMETERS: n_id, n_ano, n_mes
INPUT: n_id, n_ano, n_mes
OUTPUT: ret
RETORNA: ret (variable con el consumo)
EJEMPLO DE USO: getConsumoMedioLocalidadFecha (2,2011,1)
NOMBRE: getLineaCmax
PROPÓSITO: para conocer la energia máxima que tiene la linea
PARAMETERS: n_id
INPUT: n_id
OUTPUT: ret
RETORNA: ret (variable con la capaciad)
EJEMPLO DE USO: getLineaCmax (2)
NOMBRE: getConsumoCentral

PROPÓSITO: para conocer el consumo la linea -contadores conectados-, activa
PARAMETERS: n_id
INPUT: n_id
OUTPUT: ret
RETORNA: ret (variable con el consumo)
EJEMPLO DE USO: getLineaConsumo (2)
NOMBRE: getNomCliente
PROPÓSITO: conocer los apellidos y nombre del cliente
PARAMETERS: n_id
INPUT: n_id
OUTPUT: ret
RETORNA: ret (contiene los apellidos y nombre del cliente)
EJEMPLO DE USO: getNomCliente (2)
NOMBRE: getUbica
PROPÓSITO: conocer los datos de la ubicación
PARAMETERS: n_id
INPUT: n_id
OUTPUT: ret
RETORNA: ret (contiene los datos de la ubicación: dirección, num, piso, puerta, localidad, código postal, provincia y pais)
EJEMPLO DE USO: getUbica (2)
NOMBRE: getIdCliente
PROPÓSITO: conocer el codigo del cliente
PARAMETERS: n_id
INPUT: n_id
OUTPUT: ret
RETORNA: ret (el código del cliente)
EJEMPLO DE USO: getIdCliente (2)
NOMBRE: getConsumoLineaAnyo
PROPÓSITO: para conocer el consumo de una linea en una año concreto
PARAMETERS: n_id, n_anyo
INPUT: n_id, n_anyo
OUTPUT: ret
RETORNA: ret (variable con el consumo)
EJEMPLO DE USO: getConsumoLineaAnyo (2,2011);

Anexo.- Código – pruebas de salida de utilidades

Tabla 71 Prueba inicializarLecturas

drop table succeeded.
CREATE TABLE succeeded.
drop SEQUENCE succeeded.
CREATE succeeded.
TRIGGER Compiled.
drop table succeeded.
CREATE TABLE succeeded.
drop sequence succeeded.
CREATE succeeded.
TRIGGER Compiled.
drop table consumos succeeded.
CREATE TABLE succeeded.
DROP SEQUENCE succeeded.
CREATE succeeded.
TRIGGER Compiled.
anonymous block completed
n_cliente: 1, n_contador : 1, n_estado : 1, n_fechaEstado: 01/01/2011, n_pContrato: 4000
Ok, Alta contrato realizada con exito
n_cliente: 2, n_contador : 2, n_estado : 1, n_fechaEstado: 01/01/2011, n_pContrato: 4000
Ok, Alta contrato realizada con exito
n_cliente: 3, n_contador : 3, n_estado : 1, n_fechaEstado: 01/01/2011, n_pContrato: 4000
Ok, Alta contrato realizada con exito
n_cliente: 4, n_contador : 4, n_estado : 1, n_fechaEstado: 01/01/2011, n_pContrato: 4000
Ok, Alta contrato realizada con exito
n_cliente: 5, n_contador : 5, n_estado : 1, n_fechaEstado: 01/01/2011, n_pContrato: 4000
Ok, Alta contrato realizada con exito
n_cliente: 6, n_contador : 6, n_estado : 1, n_fechaEstado: 01/01/2011, n_pContrato: 4000
Ok, Alta contrato realizada con exito

n_cliente: 7, n_contador : 7, n_estado : 1, n_fechaEstado: 01/01/2011, n_pContrato: 4000
Ok, Alta contrato realizada con exito
n_cliente: 8, n_contador : 8, n_estado : 1, n_fechaEstado: 01/01/2011, n_pContrato: 4000
Ok, Alta contrato realizada con exito
n_cliente: 9, n_contador : 9, n_estado : 1, n_fechaEstado: 01/01/2011, n_pContrato: 4000
Ok, Alta contrato realizada con exito
n_cliente: 10, n_contador : 10, n_estado : 1, n_fechaEstado: 01/01/2011, n_pContrato: 4000
Ok, Alta contrato realizada con exito
n_cliente: 11, n_contador : 11, n_estado : 1, n_fechaEstado: 01/02/2012, n_pContrato: 4000
Ok, Alta contrato realizada con exito
n_cliente: 12, n_contador : 12, n_estado : 1, n_fechaEstado: 01/03/2012, n_pContrato: 4000
Ok, Alta contrato realizada con exito
n_cliente: 13, n_contador : 13, n_estado : 1, n_fechaEstado: 01/04/2012, n_pContrato: 4000
Ok, Alta contrato realizada con exito
n_cliente: 14, n_contador : 14, n_estado : 1, n_fechaEstado: 01/05/2012, n_pContrato: 4000
Ok, Alta contrato realizada con exito
n_cliente: 15, n_contador : 15, n_estado : 1, n_fechaEstado: 01/06/2012, n_pContrato: 4000
Ok, Alta contrato realizada con exito
n_cliente: 16, n_contador : 16, n_estado : 1, n_fechaEstado: 01/07/2012, n_pContrato: 4000
Ok, Alta contrato realizada con exito
n_cliente: 17, n_contador : 17, n_estado : 1, n_fechaEstado: 01/08/2012, n_pContrato: 4000
Ok, Alta contrato realizada con exito
n_cliente: 18, n_contador : 18, n_estado : 1, n_fechaEstado: 01/09/2012, n_pContrato: 4000
Ok, Alta contrato realizada con exito
n_cliente: 19, n_contador : 19, n_estado : 1, n_fechaEstado: 01/10/2012, n_pContrato: 4000
Ok, Alta contrato realizada con exito
n_cliente: 20, n_contador : 20, n_estado : 1, n_fechaEstado: 01/11/2012, n_pContrato: 4000
Ok, Alta contrato realizada con exito
anonymous block completed
n_actual : 100
n_lectura : 100
n_anterior : 0
n_consumo : 100
n_actual : 120
n_lectura : 120
n_anterior : 0
n_consumo : 120
n_actual : 130

n_lectura : 130
n_anterior : 0
n_consumo : 130
n_actual : 140
n_lectura : 140
n_anterior : 0
n_consumo : 140
n_actual : 150
n_lectura : 150
n_anterior : 0
n_consumo : 150
n_actual : 160
n_lectura : 160
n_anterior : 0
n_consumo : 160
n_actual : 170
n_lectura : 170
n_anterior : 0
n_consumo : 170
n_actual : 180
n_lectura : 180
n_anterior : 0
n_consumo : 180
n_actual : 190
n_lectura : 190
n_anterior : 0
n_consumo : 190
n_actual : 200
n_lectura : 200
n_anterior : 0
n_consumo : 200
n_actual : 220
n_lectura : 220
n_anterior : 0
n_consumo : 220
n_actual : 300
n_lectura : 300
n_anterior : 0
n_consumo : 300
n_actual : 10
n_lectura : 10
n_anterior : 0
n_consumo : 10
n_actual : 20
n_lectura : 20
n_anterior : 0
n_consumo : 20

n_actual : 30
n_lectura : 30
n_anterior : 0
n_consumo : 30
n_actual : 40
n_lectura : 40
n_anterior : 0
n_consumo : 40
n_actual : 50
n_lectura : 50
n_anterior : 0
n_consumo : 50
n_actual : 60
n_lectura : 60
n_anterior : 0
n_consumo : 60
n_actual : 70
n_lectura : 70
n_anterior : 0
n_consumo : 70
n_actual : 80
n_lectura : 80
n_anterior : 0
n_consumo : 80
n_actual : 90
n_lectura : 90
n_anterior : 0
n_consumo : 90
n_actual : 100
n_lectura : 100
n_anterior : 0
n_consumo : 100
n_actual : 110
n_lectura : 110
n_anterior : 0
n_consumo : 110
n_actual : 120
n_lectura : 120
n_anterior : 0
n_consumo : 120
n_actual : 30
n_lectura : 30
n_anterior : 0
n_consumo : 30
n_actual : 31
n_lectura : 31
n_anterior : 0
n_consumo : 31

n_actual : 32
n_lectura : 32
n_anterior : 0
n_consumo : 32
n_actual : 33
n_lectura : 33
n_anterior : 0
n_consumo : 33
n_actual : 34
n_lectura : 34
n_anterior : 0
n_consumo : 34
n_actual : 35
n_lectura : 35
n_anterior : 0
n_consumo : 35
n_actual : 36
n_lectura : 36
n_anterior : 0
n_consumo : 36
n_actual : 37
n_lectura : 37
n_anterior : 0
n_consumo : 37
n_actual : 38
n_lectura : 38
n_anterior : 0
n_consumo : 38
n_actual : 39
n_lectura : 39
n_anterior : 0
n_consumo : 39
n_actual : 40
n_lectura : 40
n_anterior : 0
n_consumo : 40
n_actual : 41
n_lectura : 41
n_anterior : 0
n_consumo : 41
n_actual : 40
n_lectura : 40
n_anterior : 0
n_consumo : 40
n_actual : 41
n_lectura : 41
n_anterior : 0
n_consumo : 41

n_actual : 42
n_lectura : 42
n_anterior : 0
n_consumo : 42
n_actual : 43
n_lectura : 43
n_anterior : 0
n_consumo : 43
n_actual : 44
n_lectura : 44
n_anterior : 0
n_consumo : 44
n_actual : 45
n_lectura : 45
n_anterior : 0
n_consumo : 45
n_actual : 46
n_lectura : 46
n_anterior : 0
n_consumo : 46
n_actual : 47
n_lectura : 47
n_anterior : 0
n_consumo : 47
n_actual : 48
n_lectura : 48
n_anterior : 0
n_consumo : 48
n_actual : 49
n_lectura : 49
n_anterior : 0
n_consumo : 49
n_actual : 50
n_lectura : 50
n_anterior : 0
n_consumo : 50
n_actual : 51
n_lectura : 51
n_anterior : 0
n_consumo : 51
n_actual : 50
n_lectura : 50
n_anterior : 0
n_consumo : 50
n_actual : 51
n_lectura : 51
n_anterior : 0

n_consumo : 51
n_actual : 52
n_lectura : 52
n_anterior : 0
n_consumo : 52
n_actual : 53
n_lectura : 53
n_anterior : 0
n_consumo : 53
n_actual : 54
n_lectura : 54
n_anterior : 0
n_consumo : 54
n_actual : 55
n_lectura : 55
n_anterior : 0
n_consumo : 55
n_actual : 56
n_lectura : 56
n_anterior : 0
n_consumo : 56
n_actual : 57
n_lectura : 57
n_anterior : 0
n_consumo : 57
n_actual : 58
n_lectura : 58
n_anterior : 0
n_consumo : 58
n_actual : 59
n_lectura : 59
n_anterior : 0
n_consumo : 59
n_actual : 60
n_lectura : 60
n_anterior : 0
n_consumo : 60
n_actual : 61
n_lectura : 61
n_anterior : 0
n_consumo : 61
n_actual : 70
n_lectura : 70
n_anterior : 0
n_consumo : 70
n_actual : 71
n_lectura : 71

n_anterior : 0
n_consumo : 71
n_actual : 72
n_lectura : 72
n_anterior : 0
n_consumo : 72
n_actual : 73
n_lectura : 73
n_anterior : 0
n_consumo : 73
n_actual : 74
n_lectura : 74
n_anterior : 0
n_consumo : 74
n_actual : 75
n_lectura : 75
n_anterior : 0
n_consumo : 75
n_actual : 76
n_lectura : 76
n_anterior : 0
n_consumo : 76
n_actual : 77
n_lectura : 77
n_anterior : 0
n_consumo : 77
n_actual : 78
n_lectura : 78
n_anterior : 0
n_consumo : 78
n_actual : 79
n_lectura : 79
n_anterior : 0
n_consumo : 79
n_actual : 80
n_lectura : 80
n_anterior : 0
n_consumo : 80
n_actual : 81
n_lectura : 81
n_anterior : 0
n_consumo : 81
n_actual : 170
n_lectura : 170
n_anterior : 0
n_consumo : 170
n_actual : 171

n_lectura : 171
n_anterior : 0
n_consumo : 171
n_actual : 172
n_lectura : 172
n_anterior : 0
n_consumo : 172
n_actual : 173
n_lectura : 173
n_anterior : 0
n_consumo : 173
n_actual : 174
n_lectura : 174
n_anterior : 0
n_consumo : 174
n_actual : 175
n_lectura : 175
n_anterior : 0
n_consumo : 175
n_actual : 176
n_lectura : 176
n_anterior : 0
n_consumo : 176
n_actual : 177
n_lectura : 177
n_anterior : 0
n_consumo : 177
n_actual : 178
n_lectura : 178
n_anterior : 0
n_consumo : 178
n_actual : 179
n_lectura : 179
n_anterior : 0
n_consumo : 179
n_actual : 180
n_lectura : 180
n_anterior : 0
n_consumo : 180
n_actual : 181
n_lectura : 181
n_anterior : 0
n_consumo : 181
n_actual : 180
n_lectura : 180
n_anterior : 0
n_consumo : 180

n_actual : 181
n_lectura : 181
n_anterior : 0
n_consumo : 181
n_actual : 182
n_lectura : 182
n_anterior : 0
n_consumo : 182
n_actual : 183
n_lectura : 183
n_anterior : 0
n_consumo : 183
n_actual : 184
n_lectura : 184
n_anterior : 0
n_consumo : 184
n_actual : 185
n_lectura : 185
n_anterior : 0
n_consumo : 185
n_actual : 186
n_lectura : 186
n_anterior : 0
n_consumo : 186
n_actual : 187
n_lectura : 187
n_anterior : 0
n_consumo : 187
n_actual : 188
n_lectura : 188
n_anterior : 0
n_consumo : 188
n_actual : 189
n_lectura : 189
n_anterior : 0
n_consumo : 189
n_actual : 190
n_lectura : 190
n_anterior : 0
n_consumo : 190
n_actual : 191
n_lectura : 191
n_anterior : 0
n_consumo : 191
n_actual : 290
n_lectura : 290
n_anterior : 0
n_consumo : 290

n_actual : 291
n_lectura : 291
n_anterior : 0
n_consumo : 291
n_actual : 292
n_lectura : 292
n_anterior : 0
n_consumo : 292
n_actual : 293
n_lectura : 293
n_anterior : 0
n_consumo : 293
n_actual : 294
n_lectura : 294
n_anterior : 0
n_consumo : 294
n_actual : 295
n_lectura : 295
n_anterior : 0
n_consumo : 295
n_actual : 296
n_lectura : 296
n_anterior : 0
n_consumo : 296
n_actual : 297
n_lectura : 297
n_anterior : 0
n_consumo : 297
n_actual : 298
n_lectura : 298
n_anterior : 0
n_consumo : 298
n_actual : 299
n_lectura : 299
n_anterior : 0
n_consumo : 299
n_actual : 300
n_lectura : 300
n_anterior : 0
n_consumo : 300
n_actual : 301
n_lectura : 301
n_anterior : 0
n_consumo : 301
n_actual : 500
n_lectura : 500
n_anterior : 0
n_consumo : 500

n_actual : 501
n_lectura : 501
n_anterior : 0
n_consumo : 501
n_actual : 502
n_lectura : 502
n_anterior : 0
n_consumo : 502
n_actual : 503
n_lectura : 503
n_anterior : 0
n_consumo : 503
n_actual : 504
n_lectura : 504
n_anterior : 0
n_consumo : 504
n_actual : 505
n_lectura : 505
n_anterior : 0
n_consumo : 505
n_actual : 506
n_lectura : 506
n_anterior : 0
n_consumo : 506
n_actual : 507
n_lectura : 507
n_anterior : 0
n_consumo : 507
n_actual : 508
n_lectura : 508
n_anterior : 0
n_consumo : 508
n_actual : 509
n_lectura : 509
n_anterior : 0
n_consumo : 509
n_actual : 510
n_lectura : 510
n_anterior : 0
n_consumo : 510
n_actual : 511
n_lectura : 511
n_anterior : 0
n_consumo : 511

Anexo procederes anteriores a la modificación

1. modCentral

Tabla 72 Modificación Central

Funcionalidad	El procedimiento permite modificar los registros de la tabla de Centrales.
Explicación	<p>La secuencia de acciones que se siguen en el procedimiento es la siguiente: Secuencia de acciones:</p> <ol style="list-style-type: none"> 1. comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • Los campos de idCentral, idTipoCentral, idEstado, idConexion1, idConexion2 y el resto de campos ha de estar cumplimentados, sino lanza una excepción. • Se puede cambiar el valor del campo de, para especificar un nuevo estado (no operativa, en revisión, avería etc.) • La fecha de estado tiene por defecto el valor del día. • El valor del campo de idInspeccion para poner el resultado de la nueva inspección, si no es correcto, lanza excepción, el campo de fecha con la fecha de la inspección, conservaremos también la fecha de la última inspección por requisitos del enunciado. • Modificar el valor de la capacidad máxima, en el caso de Centrales de Distribución la suma total no ha de sobrepasar a la capacidad máxima de la Central de Producción a la que estén conectados. • Modificar los valores valor1 y valor2 para ir cambiando los parámetros de emisión de cada tipo de central –idTipoCentral-. 2. Se realiza el Update en la tabla de Centrales, con los parámetros de entrada, excepto el parámetro de identificación –idCentral- 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de OK. • se inserta en la tabla LOG. 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>idLinea IN NUMBER, idConexion1 IN NUMBER, idConexion2 IN NUMBER, capacidadMaxima IN NUMBER, idEstado IN NUMBER, fechaEstado IN DATE, rst OUT VARCHAR2</p> <p>donde: idLinea corresponde al código identificador de la línea. idConexion1 corresponde al código de la central a la que se conecta. idConexion2 corresponde al código de la central o bien el código del contador. capacidadMaxima corresponde a la máxima capacidad que podemos tener conectado en esa línea. idEstado corresponde al estado de uso de esta línea, si está operativa o n, valores de la tabla de Estados. fechaEstado corresponde a la fecha del estado que tiene actualmente.</p> <p>Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p>

2. modContador

Tabla 73 Modificación Contador

Funcionalidad	El procedimiento permite modificar la tabla de Contadores en la Base de Datos-
Explicación	<p>La secuencia de acciones que sigue el procedimiento es:</p> <ol style="list-style-type: none"> 1. Comprobación de los parámetros de entrada: <ul style="list-style-type: none"> • Los campos de de idContador, idModelo ,idUbica ,idEstado,fechaEstado ,altaDisponibilidad ,telemático han de estar cumplimentados, sino lanza excepción. • Verificamos la existencia del contador que vamos a modificar. 2. Se realiza el Update en la tabla de Contadores, con los parámetros de entrada, excepto el identificador idContador y el número de serie. 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de Ok • se inserta en la tabla LOG 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>idContador IN NUMBER, idModelo IN NUMBER, idUbica IN NUMBER, idEstado IN NUMBER, fechaEstado IN DATE, altaDisponibilidad IN NUMBER, telemático IN NUMBER, rst OUT VARCHAR2</p> <p>donde: idContador corresponde al identificador del contador. idModelo corresponde al identificador del modelo. idUbica corresponde al identificador de la ubicación. idEstado corresponde al identificador del estado del contador. fechaEstado corresponde a la fecha actual del estado. altaDisponibilidad corresponde al valor de si tiene instalado el módulo de conexión dual o no. telemático corresponde al valor de si tiene instalado el módulo telemático o no.</p> <p>Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p>

3. modLineas

Tabla 74 Modificación Línea

Funcionalidad	El procedimiento permite modificar los registros de la tabla de Líneas en la Base de Datos.
Explicación	<p>La secuencia de acciones que sigue el procedimiento es:</p> <ol style="list-style-type: none"> 1. comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • Si los campos de idLinea, idConexion1, idConexion2, capacidadMaxima, idEstado y fechaEstado, están cumplimentados, sino lanza una excepción. • Verificamos la existencia de la línea que vamos a modificar. 2. Se realiza el Update en la tabla de Líneas, con los parámetros de entrada, excepto con el parámetro de identificación. 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de Ok • se inserta en la tabla LOG 4. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>idLinea IN NUMBER, idConexion1 IN NUMBER, idConexion2 IN NUMBER, capacidadMaxima IN NUMBER, idEstado IN NUMBER, fechaEstado IN DATE, rst OUT VARCHAR2</p> <p>idLinea corresponde al identificador de la línea. idConexion1 corresponde al identificador de la conexión 1. idConexion2 corresponde al identificador de la conexión capacidadMaxima corresponde al valor de la capacidad que soporta la línea. idEstado corresponde al identificador del estado de la línea. fechaEstado IN NUMBER, rst OUT VARCHAR2</p> <p>Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p>

4. modClientes

Tabla 75 Modificación Clientes

Funcionalidad	El procedimiento permite modificar los registros de la tabla de Clientes en la Base de Datos.
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 2. comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • Los campos de idCliente ,idUbica ,idEstado ,fechaEstado ,idTipo, han de estar cumplimentados, sino lanza una excepción • Verificamos que el cliente existe para modificar. 3. Se realiza el Update en la tabla de Clientes, con los parámetros de entrada, excepto el parámetro de su identificación. 4. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de Ok • se inserta en la tabla LOG 5. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>idCliente IN NUMBER, idUbica IN NUMBER, idEstado IN NUMBER, fechaEstado IN DATE, idTipo IN NUMBER, obsCliente IN VARCHAR2, NIF IN VARCHAR2, idEmpresa IN NUMBER, idPersona IN NUMBER, rst OUT VARCHAR2</p> <p>idCliente corresponde al identificador del cliente. idUbica corresponde al identificador de la ubicación. idEstado corresponde al identificador del estado. fechaEstado corresponde a la fecha del estado actual idTipo corresponde al identificador del tipo de cliente. obsCliente corresponde a las observaciones de cliente. NIF corresponde al identificador del NIF del cliente particular. idEmpresa corresponde al identificador de la Empresa. idPersona corresponde al identificador de la persona rst OUT VARCHAR2</p> <p>Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p>

5. modUbicaciones

Tabla 76 Modificación Ubicación

Funcionalidad	El procedimiento permite modificar los registros de la tabla de Ubicaciones en la Base de Datos.
Explicación	<p>La secuencia de acciones que sigue en el procedimiento es:</p> <ol style="list-style-type: none"> 1. comprobación de los parámetros de entrada. <ul style="list-style-type: none"> • Todos los campos de la tabla han de estar cumplimentados, sino lanza una excepción. • Comprobamos la existencia de la Ubicación que vamos a modificar. 2. Se realiza el Update en la tabla Ubicaciones, con los parámetros de entrada, excepto con el parámetro de su identificación. 3. Si la operación es correcta: <ul style="list-style-type: none"> • mensaje de Ok 4. Se inserta en la tabla en la tabla Log 5. Si se produce una excepción: <ul style="list-style-type: none"> • El mensaje de salida tendrá un texto con la explicación correspondiente de excepción. • Se inserta en la tabla LOG
Parámetros de entrada y salida	<p>idUbica IN NUMBER, idVia IN NUMBER, idLocalidad IN NUMBER2, dirección IN VARCHAR2, codigoPostal IN NUMBER, numero IN NUMBER, piso IN NUMBER, puerta IN NUMBER, tipoVivienda IN NUMBER</p> <p>idUbica corresponde al identificador de Ubicaciones. idVia corresponde al identificador deVia. idLocalidad corresponde al identificador de la Localidad. dirección corresponde a la descripción de la dirección codigoPostal corresponde al código postal. numero corresponde al número de la vivienda. piso corresponde al número del piso por defecto "-". puerta corresponde al número de la puerta por defecto "-". tipoVivienda corresponde al tipo de vivienda, desplegable con los valores posibles.</p> <p>Los parámetros de salida tomarán los siguientes valores: OK, si se ha producido correctamente ERROR + descripción del error, en caso de que se lance alguna excepción.</p>